QDTrack: Quasi-Dense Similarity Learning for Appearance-Only Multiple Object Tracking

Tobias Fischer, Thomas E. Huang, Jiangmiao Pang, Member, IEEE, Linlu Qiu, Haofeng Chen, Trevor Darrell, and Fisher Yu, Member, IEEE

Abstract—Similarity learning has been recognized as a crucial step for object tracking. However, existing multiple object tracking methods only use sparse ground truth matching as the training objective, while ignoring the majority of the informative regions in images. In this paper, we present Quasi-Dense Similarity Learning, which densely samples hundreds of object regions on a pair of images for contrastive learning. We combine this similarity learning with multiple existing object detectors to build Quasi-Dense Tracking (QDTrack), which does not require displacement regression or motion priors. We find that the resulting distinctive feature space admits a simple nearest neighbor search at inference time for object association. In addition, we show that our similarity learning scheme is not limited to video data, but can learn effective instance similarity even from static input, enabling a competitive tracking performance without training on videos or using tracking supervision. We conduct extensive experiments on a wide variety of popular MOT benchmarks. We find that, despite its simplicity, QDTrack rivals the performance of state-of-the-art tracking methods on all benchmarks and sets a new state-of-the-art on the large-scale BDD100K MOT benchmark, while introducing negligible computational overhead to the detector.

Index Terms—Multiple object tracking, quasi-dense similarity learning.

I. INTRODUCTION

MULTIPLE Object Tracking (MOT) is a fundamental and challenging problem in computer vision, widely used in safety monitoring, autonomous driving, video analytics, and other applications. Contemporary MOT methods [1], [2], [3], [4], [5] mainly follow the tracking-by-detection paradigm [6]. That is, they detect objects on each frame and then associate them according to the estimated similarity between each instance.

...
Besides similarity, the inference pipeline, which measures the instance similarity and maintains a track history, also plays an important role in the tracking performance, since it needs to consider false positives, missed detections, newly appeared objects, and terminated tracks. To better deal with these cases, we introduce the bi-directional softmax similarity metric that enforces bi-directional consistency. In particular, objects that do not have matching targets in the other frame will lack a bi-directional matching and thus have low similarity scores to all other objects. Furthermore, we include unmatched objects in the previous frame, which we call backdrops, for matching to better filter false positives that could otherwise act as distractors in following frames. We compose object detectors, quasi-dense similarity learning, and our inference pipeline to build Quasi-Dense Tracking (QDTrack) models. Since the publication of our initial work [17], QDTrack has been widely adopted for other tracking problems, such as segmentation tracking [18], long-tailed multi-object tracking [19], [20], and 3D object tracking [21], [22].

In addition to the findings of our initial work [17], we show that quasi-dense instance similarity learning is not limited to video data, but can learn effective instance representations from static images alone. In particular, we show that we can achieve competitive tracking performance without any annotations for association and/or video input. We systematically analyze the effectiveness of different augmentation techniques for appearance similarity learning. Moreover, in this journal extension we conduct extensive experiments on a wide variety of tracking benchmarks, namely MOT [23], DanceTrack [24], BDD100K [25], Waymo [26], and TAO [27]. In addition, we show the flexibility of our method by combining it with different base models and object detectors. Despite its simplicity, QDTrack rivals the performance of state-of-the-art methods without bells and whistles, and sets a new state-of-the-art on the large-scale BDD100K tracking benchmark. QDTrack allows for joint, end-to-end training of detection and instance similarity, thereby simplifying the training and inference pipelines of MOT frameworks. In addition, our embedding extractor only adds negligible overhead to the inference time of the detector. We hope the simplicity and strengths of QDTrack motivates further research on similarity learning for multiple object tracking.

II. RELATED WORK

In MOT, the current leading paradigm is tracking-by-detection [6]. Tracking-by-detection methods detect objects in each individual frame, and subsequently associate the detections over time. They differ in their data association mechanisms and cues that are used in the association process. A variety of approaches have been developed to solve the data association problem, e.g., network flow formulations [28], quadratic pseudo boolean optimization [29], conditional random fields [30], or multi-hypothesis tracking [31]. Many works have focused on finding the best cues to exploit for data association, such as 2D motion [2], [7], [10], [32], [33], [34], 3D motion [21], [35], [36], [37], [38], or visual appearance similarity [1], [3], [31], [39], [40], [41], [42], [43]. In this work, we focus on learning visual appearance similarity and follow the tracking-by-detection paradigm.

Location and motion in MOT: Spatial proximity has been proven effective to associate objects in consecutive frames [2], [7]. Some methods use 2D motion, such as predictions of a Kalman Filter [2], [33], [44], [45], optical flow [32], and displacement regression [10], [34], to estimate similarity for object association. However, these methods are brittle when it comes to varying video frame rate and complex camera motion, since the 2D motion of the objects depends highly on these factors. Thus, other methods instead rely on 3D motion cues to associate objects over time, since in 3D camera and object motion can be decomposed. A common paradigm [35], [36] is to track objects with 3D bounding boxes and motion estimates derived from e.g. scene flow. In contrast, [37], [38], [46] explored to track and reconstruct objects in 3D by estimating the object’s rigid-body transformation between two frames. Although these methods show promising results, many [3], [44] still rely on an extra appearance similarity model as a complementary component to re-identify vanished objects, complicating the entire framework. Our approach is orthogonal to this line of work, as we rely solely on appearance-based instance similarity and a simple nearest-neighbor search to associate objects.

Appearance similarity in MOT: In order to strengthen tracking and re-identify vanished objects, some methods exploit appearance similarity extracted from an independent model [1], [3], [31], [39], [40], [41], [42], [43] or add an extra embedding head to the detector for end-to-end training [5], [12], [47], [48]. However, they still learn appearance similarity following the practice in image similarity learning, and measure the instance similarity by cosine distance. An appearance similarity model is trained either as a $n$-class classification problem [3], where $n$ is equal to the number of identities in the whole training set, or using triplet loss [49]. The classification problem is hard to extend to large-scale datasets, while the triplet loss only compares each training sample with two other samples, leading to sub-optimal instance similarity learning. As a consequence, these methods still rely heavily on motion models and displacement predictions to track objects, and appearance similarity only takes a secondary role. In contrast, QDTrack learns instance similarity from densely-connected contrastive pairs and associates objects with a simple nearest neighbor search in feature space, which allows for a simpler tracking framework without compromising accuracy.

Joint detection and tracking: Instead of treating object detection and association as separate modules, Detect & Track [10] is the first work that jointly optimizes object detection and tracking modules. It predicts the displacements of each object in consecutive frames and associates them with the Viterbi algorithm. Tracktor [1] directly adopts a detector as a tracker. CenterTrack [4] and Chained-Tracker [11] predict the object displacements with pair-wise inputs to associate the objects. Other methods focus on learning visual appearance and detection jointly [5], [47], [48], [50], adding an extra embedding head to the detection network. However, these methods do not fully exploit image information for similarity learning. Recent work [51], [52], [53] focuses on leveraging Transformer networks to integrate
tracking and detection into a single, query-based architecture. These methods track by propagating queries across timesteps, processing them with a Transformer that outputs the tracking result. In this work, we focus on learning appearance similarity from quasi-dense samples jointly with detection.

**Self-supervised representation learning:** The field of self-supervised representation learning has seen significant progress in recent years, fueled by a number of methods relying on contrastive learning [13], [16], [54], [55], [56], [57], [58], [59] that have shown promising performance. The main paradigm of these methods is to learn a representation that is similar for two versions of the same image, where one is distorted with random image augmentations, while enforcing that this representation is dissimilar to other pairs in the current training batch. While this has proven to be very effective, it has not yet drawn much attention when learning the instance similarity in MOT. In this paper, we supervise densely matched quasi-dense samples with multiple positive contrastive learning inspired by [59]. In contrast to image-level contrastive learning, our method allows for multiple positive training, while the methods mentioned above can only handle the case when there is only a single positive target. The promising results of our method show the importance of representation learning for the MOT problem.

**Learning to track from static images:** Learning to track objects from static images where no association annotations are available has recently been proposed by multiple methods [4], [48]. CenterTrack [4] proposes to use data augmentation to simulate video input from a given static image to obtain 2D offsets to learn object motion. FairMOT [48] treats every object in a given detection dataset as a unique class and learns to distinguish between those to learn tracking from static images. In contrast to learning simulated motion or treating every object over a whole dataset as unique, we show that our similarity learning scheme can effectively learn to track objects from static images with comparable accuracy to video input without further modification. We draw inspiration from the success of recent self-supervised representation learning methods and apply our similarity learning scheme between two augmented instances of the same input image to improve appearance-based object association.

III. METHOD

We propose quasi-dense similarity learning: to learn a feature embedding space that can associate identical objects and distinguish different objects for online multiple object tracking. We define dense matching to be matching between bounding box candidates at all pixel locations and sparse matching to be matching between ground truth box labels as matching candidates. In contrast, quasi-dense matching considers potential object candidates specifically at potential object regions. The main ingredients of Quasi-Dense Tracking (QDTrack) are object detection, instance similarity learning, and object association.

A. Object Detection

Our method can be easily coupled with both two-stage and one-stage detectors with end-to-end training. Object detectors contain two components, a feature extractor and a bounding box prediction head. The feature extractor is typically composed of a base model to extract image features and a Feature Pyramid Network (FPN) [60] to obtain multi-scale features. The bounding box prediction head produces dense bounding box candidates, from which we sample quasi-dense samples by filtering with Non-Maximal Suppression (NMS). The resulting samples indicate likely object regions that include multiple overlapped bounding boxes for each object.

B. Quasi-Dense Similarity Learning

We use regions that likely contain objects to learn the instance similarity with quasi-dense matching. The full training pipeline is shown in Fig. 1. Given a key image $I_k$ for training, we randomly select a reference image $I_r$ from its temporal neighborhood. The neighbor distance is constrained by an interval $k$. We use the object regions from both images and RoI Align [61] to obtain their corresponding feature maps from the image features. We add an extra lightweight embedding head in parallel with the original bounding box head to extract features for each region. A region is defined as a positive sample to a ground truth object if it has an IoU higher than $\alpha_1$ or negative if lower than $\alpha_2$. A matching of regions on two frames is positive if the two regions are associated with the same ground truth object and negative otherwise.

Assume there are $V$ samples on the key frame as training samples and $K$ samples on the reference frame as contrastive targets. For each training sample, we can use the non-parametric softmax [13], [16] with cross-entropy to optimize the feature embeddings

$$L_{\text{embed}} = -\log \frac{\exp(v \cdot k^+)}{\exp(v \cdot k^+) + \sum_{k^-} \exp(v \cdot k^-)},$$

where $v$, $k^+$, $k^-$ are feature embeddings of the training sample, its positive target, and negative targets in $K$. The overall embedding loss is averaged across all training samples, but we only illustrate one training sample for brevity.

We apply dense matching between object regions on the pairs of images. Specifically, each sample in $I_k$ is matched to all samples in $I_r$, in contrast to only using sparse sample crops (mostly ground truth boxes) to learn instance similarity in previous works [49], [62]. The goal of the contrastive matching between regions in $I_k$ and $I_r$ is to make the feature embeddings robust to perturbations commonly encountered in video data, such as change in object scale, motion blur, and viewpoint or lighting changes ($k^+$), while discriminating appearance differences associated with object identity, such as different texture ($k^-$).

Each training sample in the key frame has more than one positive targets in the reference frame, so (1) can be extended as

$$L_{\text{embed}} = - \sum_{k^+} \log \frac{\exp(v \cdot k^+)}{\exp(v \cdot k^+) + \sum_{k^-} \exp(v \cdot k^-)}.$$

However, this equation does not treat positive and negative targets fairly. Namely, each negative is considered multiple times while each positive is considered only once. Alternatively, we
Fig. 1. Training pipeline. After we extract feature embeddings for all quasi-dense samples on a pair of key and reference images, we apply dense matching between them and optimize the learned representation with multiple positive contrastive learning. The resulting embedding space effectively discriminates different instances.

Algorithm 1: Inference Pipeline of QDTrack for Associating Objects Across a Video Sequence.

Input: frame index $t$, detections $b_i$, scores $s_i$, detection embeddings $n_i$ for $i = 1 \ldots N$, and track embeddings $m_j$ for $j = 1 \ldots M$

1: DuplicateRemoval($b_i$)
2: for $i = 1 \ldots N$, $j = 1 \ldots M$ # compute matching scores
3: $f(i,j) = \text{bisoftmax}(n_i, m_j)$
4: end for
5: for $i = 1 \ldots N$ # track management
6: $c = \max(f(i))$ # match confidence
7: $j_{\text{match}} = \argmax(f(i))$ # matched track ID
8: if $c > \beta_{\text{match}}$ and $s_i > \beta_{\text{obj}}$ and $\text{isNotBackdrop}(j_{\text{match}})$ # object match found
9: updateTrack($j_{\text{match}}, b_i, n_i, t$) # update track
10: else if $s_i > \beta_{\text{new}}$
11: createTrack($b_i, n_i, t$) # create new track
12: else
13: addBackdrop($b_i, n_i, t$) # add new backdrop
14: end if
15: end for

We further adopt L2 loss as an auxiliary loss

$$L_{\text{aux}} = \left( \frac{\mathbf{v} \cdot \mathbf{k} - c}{\|\mathbf{v}\|\|\mathbf{k}\|} \right)^2,$$ (5)

where $c$ is 1 if the match of two samples is positive and 0 otherwise. Note the auxiliary loss aims to constrain the magnitude of the logits and cosine similarity instead of improving the performance. We sample all positive pairs and three times more negative pairs to calculate the auxiliary loss and use hard negative mining.

The entire network is jointly optimized under

$$L = L_{\text{det}} + \gamma_1 L_{\text{embed}} + \gamma_2 L_{\text{aux}},$$ (6)

where $\gamma_1$ and $\gamma_2$ are set to 0.25 and 1.0 by default in this paper.

C. Object Association and Track Management

Tracking objects across frames purely based on object feature embeddings introduces many challenges. False positives, ID switches, newly appeared objects, and terminated tracks all increase the matching difficulty. We here introduce our inference pipeline that utilizes instance similarity for object association and a track management scheme to address these problems. The entire pipeline is shown in Fig. 2 and described in Algorithm 1.

Duplicate removal: Most object detectors only use intra-class NMS to remove duplicate detections within each class, which results in some detections that are in the same location but with different categories. For object tracking, this is undesirable as it will create duplicate object embeddings. We instead use inter-class NMS to avoid this issue.

Bi-directional softmax: Our main inference strategy is bi-directional matching in the feature embedding space. Assume there are $N$ detected objects in frame $t$ with feature embeddings $\mathbf{n}$ and $M$ matching candidates with feature embeddings $\mathbf{m}$ from the past $x$ frames. The instance similarity $f$ between objects and their matching candidates is obtained by a bi-directional softmax

$$L_{\text{embed}} = \log \left[ 1 + \sum_{\mathbf{k}^+} \exp(\mathbf{v} \cdot \mathbf{k}^- - \mathbf{v} \cdot \mathbf{k}^+) \right].$$ (3)

Then in the multi-positive scenario, it can be extended by accumulating the positive term as

$$L_{\text{embed}} = \log \left[ 1 + \sum_{\mathbf{k}^+} \sum_{\mathbf{k}^-} \exp(\mathbf{v} \cdot \mathbf{k}^- - \mathbf{v} \cdot \mathbf{k}^+) \right].$$ (4)
#### IV. Experiments

We conduct experiments on a variety of MOT benchmarks including MOT17 [23] and MOT20 [63], DanceTrack [24], BDD100K [25], Waymo [26], and TAO [27], and compare our method extensively to the state-of-the-art. In addition, we show that we can effectively perform tracking even without tracking supervision or video data. We demonstrate the flexibility of our method by combining it with different detection methods and feature-extraction base models and conduct extensive ablation studies on all aspects of our method. Finally, we also present a straightforward extension of our method to segmentation tracking and give insights on the limitations of our method. More detailed oracle and failure case analyses are presented in the appendix, available online.

#### A. Datasets

**MOT Challenge:** We perform experiments on two of the MOT Challenge benchmarks, namely MOT17 [23] and MOT20 [63]. The MOT Challenge videos contain high-density public spaces such as street scenes and malls with many pedestrians, creating challenging tracking conditions with heavy occlusions. Only pedestrians are evaluated in this benchmark. Since these datasets do not provide official validation sets, we split each training video into two halves: the first half for training and the second half for validation following [4], [5], [48].

The MOT17 dataset contains 7 videos (5,316 images) for training and 7 videos (5,919 images) for testing. The video frame rate is 25–30 frames per second (FPS). The MOT20 dataset includes heavily crowded scenes and contains 4 videos (8,931...
images) for training and 4 videos (4,479 images) for testing. The video frame rate is 25 FPS.

DanceTrack: The DanceTrack [24] benchmark is a large-scale dataset for multi-human tracking consisting mostly of group dancing videos. The dataset is unique in that by relying mostly on group dancing videos, the objects to track often have similar appearance, diverse motion, and extreme articulation. It features 40 videos for training, 25 videos for validation and 35 videos for testing, with a total of 105,855 frames captured at 20 FPS.

BDD100K: The large-scale, diverse driving dataset BDD100K [25] contains 100,000 video sequences of dashcam driving footage. It contains several subsets with different types of annotations. We use the detection and tracking sets for training and the tracking set for evaluation. The tracking set annotates 8 categories for evaluation. It contains 1,400 videos (278 k images) for training, 200 videos (40 k images) for validation, and 400 videos (80 k images) for testing. The detection set has 70,000 images for training. The images in the tracking set are annotated at 5 FPS.

Waymo: Waymo open dataset [26] contains images from 5 cameras associated with 5 different directions: front, front left, front right, side left, and side right. There are 3,990 videos (790 k images) for training, 1,010 videos (200 k images) for validation, and 750 videos (148 k images) for testing. It annotates 3 classes for evaluation. The videos are annotated at 10 FPS.

TAO: TAO dataset [27] annotates 482 classes in total, which are a subset of the classes annotated in the LVIS dataset [79]. It has 400 videos, 216 classes in the training set, 988 videos, 320 classes in the validation set, and 1,419 videos, 369 classes in the test set. The classes in train, validation, and test sets may not overlap. The videos are annotated at 1 FPS. The annotated classes in TAO follow a long-tailed distribution, e.g., half of the annotated instances are of class person and a sixth of the objects are of class car, while there are many classes with only few annotated instances.

B. Metrics

We use several well-established tracking metrics for evaluation.

MOTA: The Multiple Object Tracking Accuracy (MOTA) [80] metric computes tracking accuracy in tandem with detection accuracy. It is defined as

\[
\text{MOTA} = 1 - \frac{\sum_t (m_t + f_t + e_t)}{\sum_t g_t},
\]

where \( t \) is the timestep, \( m_t \) is the number of misses, \( f_t \) is the number of false positives, \( e_t \) is the number of mismatches, and \( g_t \) is the number of objects. MOTA weights detection performance more heavily than association performance. For tracking with multiple classes, we compute MOTA for each class independently then take an average over the number of classes (mMOTA).

IDF1: The Identification F1 Score (IDF1) [81] matches ground truth and predictions on the trajectory level and computes a corresponding F1-score. It is defined as

\[
\text{IDF1} = \frac{|\text{IDTP}|}{|\text{IDTP}| + 0.5|\text{IDFN}| + 0.5|\text{IDFP}|},
\]

where IDTP, IDFN, and IDFP are the true positive, false negative, and false positive trajectories. IDF1 focuses on measuring association performance. Similar to MOTA, we compute an average over multiple classes for multi-class tracking (mIDF1).

HOTA: Higher Order Tracking Accuracy (HOTA) [82] aims to fairly combine the evaluation of detection and association. Therefore, HOTA is composed of two accuracy scores, detection accuracy \( \text{DetA} \) and association accuracy \( \text{AssA} \). \( \text{DetA} \) is defined as

\[
\text{DetA} = \frac{|\text{TP}|}{|\text{TP}| + |\text{FN}| + |\text{FP}|},
\]

where TP, FN, and FP are the true positive, false negative, and false positive detections. Additionally, detection recall \( \text{DetRe} \) and detection precision \( \text{DetPr} \) are used. \( \text{AssA} \) is defined as

\[
\text{AssA} = \frac{1}{|\text{TP}|} \sum_{a \in \text{TP}} \frac{|\text{TPA}(a)|}{|\text{TPA}(a)| + |\text{FNA}(a)| + |\text{FPNa}(a)|},
\]

where TPA, FNA, and FPA are the true positive, false negative, and false positive associations. Similarly, association recall \( \text{AssRe} \) and association precision \( \text{AssPr} \) are used. HOTA is computed as a geometric mean of \( \text{DetA} \) and \( \text{AssA} \).

C. Implementation Details

Two-stage object detectors use a Region Proposal Network (RPN) to first generate a set of proposal bounding boxes, i.e., Region of Interests (RoIs). We use the RoIs from the RPN for similarity learning. One-stage object detectors do not have a proposal stage and instead perform detection directly on the entire dense grid of bounding box locations. As our similarity learning protocol requires object regions, we generate them by simply using the dense detection outputs before post-processing. We follow the same box filtering procedure as the RPN [76], where we keep the most confident 1000 boxes then apply Non-maximum Suppression (NMS) with an IoU threshold of 0.7. We investigate Faster R-CNN [76] for two-stage detectors and RetinaNet [83] (anchor-based) and YOLOX [73] (anchor-free) for one-stage detectors in this work.

We select 128 RoIs from the key frame as training samples, and 256 RoIs from the reference frame with a positive-negative ratio of 1.0 as contrastive targets. We use IoU-balanced sampling [84] to sample negative RoIs, which better balances the sampling of hard negatives according to their IoU. We use 4conv-1fc head with group normalization [85] to extract feature embeddings. The channel number of embedding features is set to 256 by default. We keep backdrops only from the previous frame. For association, we associate objects only when they are classified as the same category.

On MOT17 and MOT20, we follow the recent practice of [44], [45], [71] and train QDTrack with the popular YOLOX [73] detector on the union of CrowdHuman [64] and the respective MOT benchmark. On DanceTrack and BDD100K, we again follow [44] and use the same detector, but we only train on the
TABLE I
Comparison to State-of-the-Art on MOT Challenge Benchmarks

| Method       | Detector | Base model | Datasets | MOTA ↑ | IDFI ↑ | HOTA ↑ | AssA ↑ | DetA ↑ | AssRe ↑ | DetRe ↑ | DistRe ↑ |
|--------------|----------|------------|----------|--------|-------|--------|--------|--------|--------|--------|---------|
| MOT17        |          |            |          |        |       |        |        |        |        |        |         |
| CenterTrack  | CenterNet | DLA-34     | MOT, CH  | 73.7   | 64.7  | 59.2   | 51.0   | 55.8   | 56.6   | 73.0   | 57.5   | 78.9    |
| EstMOT [49]  | CenterNet | DLA-34     | MOT, CH  | 73.7   | 64.7  | 59.2   | 51.0   | 55.8   | 56.6   | 73.0   | 57.5   | 78.9    |
| ReMOT [69]   |          |            |          |        |       |        |        |        |        |        |         |
| OC-SORT [46] | YOLO-X   | Motifizrd CSP | MOT, CH | 73.7   | 64.7  | 59.2   | 51.0   | 55.8   | 56.6   | 73.0   | 57.5   | 78.9    |
| MAA [70]     | CrowdDes | R50 FPN    | MOT, CH  | 73.7   | 64.7  | 59.2   | 51.0   | 55.8   | 56.6   | 73.0   | 57.5   | 78.9    |
| StrongSORT [72] | YOLO-X | Motifized CSP | MOT, CH | 73.7   | 64.7  | 59.2   | 51.0   | 55.8   | 56.6   | 73.0   | 57.5   | 78.9    |
| ByteTrack [45] | YOLO-X   | Motifized CSP | MOT, CH | 73.7   | 64.7  | 59.2   | 51.0   | 55.8   | 56.6   | 73.0   | 57.5   | 78.9    |
| QDTrack (Ours) |          |            |          |        |       |        |        |        |        |        |         |

We benchmark our method against existing works on both MOT17 and MOT20 test sets with private detections. Datasets include crowdhuman (CH) [64], citperson (CP) [65], and cth [66]. ↑ indicates using COCO pre-trained weights. ↑ means higher is better.

D. Comparison to State-of-the-Art

We compare our method to existing literature across five challenging multi-object tracking benchmarks.

MOT: The official benchmark results with private detectors on MOT17 and MOT20 benchmarks are shown in Table I. Our method achieves competitive performance on both benchmarks, despite only utilizing appearance cues for association. Notably, QDTrack obtains a high score of 63.5 HOTA on MOT17 and 60.0 HOTA on MOT20. Since the MOT benchmarks are captured at a relatively high frame rate and include only limited camera motion, 2D motion based association [44], [45], [71] works very well in this scenario. However, this only holds true for the high frame rate scenario. In Fig. 3 we show that when reducing the video frame rate on the MOT17 validation split, the performance of ByteTrack [44] drops quickly and even completely fails at a frame rate of 1 FPS, while our tracker still achieves 58.6 MOTA at this frame rate. Furthermore, we show that our tracker also compares favorably to other appearance-based trackers in this regime, namely FairMOT [48], which drops to 44.3 MOTA maintaining only 64.1% of its original performance, while we maintain 77.4%. This demonstrates that QDTrack can effectively handle a range of video frame rates, which is crucial in real-world applications.

DanceTrack: The results on the benchmark are shown in Table II. Surprisingly, while DanceTrack was specifically designed to provide a platform to develop MOT algorithms that rely less on visual appearance and more on motion analysis, we find that our appearance based tracker performs very well...
on this dataset, reaching a HOTA score only marginally behind the state-of-the-art method OC-SORT [45] (−0.9 HOTA). We achieve this score without any bells and whistles, naively applying the same configuration as in our MOT17 experiments to train on the DanceTrack dataset, following [44]. This reinforces our argument that one can in fact build a robust tracking algorithm by relying on our quasi-dense instance similarity.

**BDD100 K:** The main results on the BDD100 K tracking validation and testing sets are in Table III. On the validation set, QDTrack with YOLOX-X achieves 42.1 mMOTA and 54.3 mIDF1, which are the second-best results behind ByteTrack [45]. Still, QDTrack achieves much better results in IDF1 (73.3 versus 70.4). On the test set, QDTrack with YOLOX-X achieves a high score of 42.4 mMOTA, 55.6 mIDF1, and 73.9 IDF1, outperforming all other methods by a significant margin. In particular, QDTrack outperforms ByteTrack by 2.3 mMOTA and 2.6 IDF1. QDTrack with Faster R-CNN also achieves a competitive score of 38.7 mMOTA, 54.1 mIDF1, and 74.0 IDF1, outperforming other methods using the same detector. TETer [19] is an extension of QDTrack that employs a new association strategy designed for improving long-tailed object tracking. These results demonstrate that QDTrack can perform well even on a more challenging large-scale benchmark with a simple framework.

**Waymo:** Table IV shows our main results on Waymo open dataset. We report the results on the validation set following the setup of RetinaTrack [12], which only conduct experiments on the vehicle class. We also report the overall performance for future comparison. We report the results on the test set via official rules. Our method outperforms all baselines on both validation set and test set. We obtain 44.0 MOTA and 56.8 IDF1 on the validation set and 49.4 MOTA/L1 and 43.9 MOTA/L2 on the test set. The performance of vehicle on the validation set is 10.7, 13.0, and 17.4 points higher than RetinaTrack [12], Tracktor++ [1, 12], and IoU baseline [12], respectively. Our model with ResNet-101 and deformable convolution (DCN) [87] has state-of-the-art performance on the test benchmark, which is on par with the champion of Waymo 2020 2D Tracking Challenge (HorizonMOT) despite only using a simple single model.

**TAO:** The results for TAO are shown in Table V. We obtain 16.1 AP50 on the validation set and 12.4 AP50 on the test set. The results are 2.9 points and 2.2 points higher than TAO’s baseline. Although we only boost the overall performance by 2 to 3 points, we outperform the baseline by a large margin on frequent classes, i.e., 38.6 points versus 18.5 points on person. This improvement is not well represented in the standard evaluation metrics of TAO, since it averages per-class scores across hundreds of classes.

We report results and compare with existing works on the BDD100 k tracking validation and test set. † indicates using COCO pre-trained weights.

| Split | Method | Detector | Base model | mMOTA ↑ | mIDF1 ↑ | MOTA ↑ | IDF1 ↑ | FN ↓ | FP ↓ | ID Sw. ↓ | MT ↑ | ML ↓ |
|-------|--------|----------|------------|----------|----------|--------|--------|------|------|----------|------|------|
| val   | Yu et al [25] | FRCCN | DLA-34 | 25.9 | 44.5 | 56.9 | 66.8 | 122406 | 52372 | 8315 | 8396 | 3795 |
|       | DeepSORT [3] | FRCCN | R50-FPN | 35.2 | 49.3 | - | - | - | - | - | - | - |
|       | TETer [19] | FRCCN | R50-FPN | 39.1 | 53.3 | - | - | - | - | - | - | - |
|       | ByteTrack [45] † | YOLOX-X | Modified CSP | 45.5 | 54.8 | 69.1 | 70.4 | 92805 | 34998 | 9140 | 9626 | 3005 |
| QDTrack (Ours) | FRCCN | R50-FPN | 37.7 | 52.9 | 65.7 | 72.7 | 104861 | 41355 | 5640 | 9649 | 2874 |
|       | QDTrack (Ours) † | YOLOX-X | Modified CSP | 42.1 | 54.3 | 68.2 | 73.3 | 83395 | 48798 | 8478 | 10925 | 2272 |

We show results of our method compared with existing methods on the Waymo open tracking validation set using pp-netometrics library (top) and test set using official evaluation (bottom). We use faster R-CNN [76] as our detector. * indicates methods using undisclosed detectors.

| Split | Method | Category | MOTA ↑ | IDF1 ↑ | FN ↓ | FP ↓ | ID Sw. ↓ | MT ↑ | ML ↓ | mAP ↑ |
|-------|--------|----------|--------|--------|------|------|----------|------|------|-------|
| val   | IoU baseline [12] | Vehicle | 38.3 | - | - | - | - | - | - | 45.8 |
|       | Tracktor++ [1, 12] | Vehicle | 42.6 | - | - | - | - | - | - | 42.4 |
|       | RetinaTrack [12] | Vehicle | 44.9 | - | - | - | - | - | - | 45.7 |
| QDTrack (Ours) | Vehicle | 55.6 | 66.2 | 514544 | 214998 | 24309 | 17595 | 5559 | 49.5 |
|       | All | 44.0 | 56.8 | 674064 | 264886 | 30712 | 21410 | 7510 | 40.1 |

We show results of our method compared with existing methods on the Waymo open tracking validation set using pp-netometrics library (top) and test set using official evaluation (bottom). We use faster R-CNN [76] as our detector. * indicates methods using undisclosed detectors.
GTR [75] and AOA [78] are recent methods proposed to tackle long-tail multi-object tracking. Although they outperform our method, GTR is an offline method and AOA utilizes separate ReID networks trained on additional data.

### E. Learning to Track From Static Images

Since our quasi-dense instance similarity learning is agnostic to how the image pair is generated during training, we investigate how we can leverage static images where no association annotations are available. Inspired by recent literature in self-supervised representation learning [57], [58], we experiment with different data augmentations on static images to learn discriminative instance representations from static input. In particular, for a given training sample in a detection dataset, we generate two distorted images via data augmentation techniques. We find that random horizontal flip (HF), multi-scale resize and crop (MS), color jittering (Color), and MixUp / Mosaic augmentations are the most suitable for our use-case. If the augmentation parameters are not shared across the key and reference view, we denote it with ‘NC’ (non-consistent). We only use MixUp / Mosaic with consistent parameters in order to compose the same images between key and reference views and thus be able to match objects across them. To train our models, we utilize the detection (image) and tracking (video) splits of BDD100 K. Note that the tracking split contains much more data, thus influencing the detection performance.

The results of our experiments are shown in Table VI. We use Faster-RCNN [76] as the detector with ResNet-50 [88] and FPN [60] as the base model and evaluate the tracking performance on the BDD100 K tracking benchmark [25]. We observe that when we only apply consistent HF, the tracking performance is far behind the version trained with full tracking supervision. By adding in non-consistent augmentations and MixUp/Mosaic, we can narrow this gap and achieve comparable accuracy to the fully supervised model. In particular, we exceed the mMOTA of the fully supervised baseline trained without augmentations besides HF when training on the same amount of training data by a significant margin. This clearly shows that not only detection, but also association benefits greatly from the data augmentation, and that with proper data augmentation, our similarity learning scheme can track objects effectively while trained on static images alone. If we use the same amount of training data, we indeed rival the performance of the best supervised model, shown by the small gap in nMOTA (−0.1 points).

In addition, we observe that the data augmentation scheme can also benefit the supervised models, reaching a much higher score than in our initial work [17] without changing the network architecture (+1.2 points in mMOTA, +2.1 points in mIDF1). The increase in mIDF1 highlights the benefit of data augmentation to the robustness of instance similarity learning.

### F. Ablation Studies

We conduct ablation studies on the validation set of BDD100K [25], where we investigate the importance of the major model components for training and inference procedures.

**Different object detectors, feature extractors, and training schedules:** We combine our method with different object detectors and feature extractors to verify the flexibility of our instance similarity learning scheme. In Table VII, we show the tracking performance of our method with ResNet-50, ResNet-101 [88], as well as the modified CSPNet [89] on the tracking validation set of BDD100 K. We combine those feature extractors with a Faster-RCNN [76] detector and observe that ResNet-101 achieves the best performance with 66.2 MOTA, 73.1 IDF1, and 35.3 AP.

In addition, we apply our method on two more base object detection models, namely the anchor-based, single-stage RetinaNet [83] and the anchor-free, single-stage YOLOX [73]. Both methods produce reasonable results, and the YOLOX model achieves the best overall scores with 68.2 MOTA, 73.3 IDF1, and 38.9 AP. It shows that our method can work independent of feature extractor or base detection model. Finally, we also experiment with different training schedules. We investigate the effect of longer training, increasing the epochs from 12 (1x schedule) to 24 (2x schedule) and 25. Note that we use the extensive data augmentation techniques presented in Section IV-E in this ablation study to counteract overfitting when training with longer schedules. We find that increasing the number of epochs does not help the smaller ResNet models, but is beneficial for training very large models like YOLOX-X.

**Importance of quasi-dense matching:** The results are presented in the top sub-table of Table VIII. We use a Faster R-CNN detector with ResNet-50 base model. MOTA and IDF1 are calculated over all instances without considering categories as overall evaluations. We use cosine distance to calculate the similarity scores during the inference procedure. Compared to learning with sparse ground truths, quasi-dense tracking improves the overall IDF1 by 4.8 points (63.0% to 67.8%). The significant improvement on IDF1 indicates quasi-dense tracking greatly improves the feature embeddings and enables more accurate associations.

We then analyze the improvements in detail. In the table, we can observe that when we match each training sample to more negative samples and train the feature space with (1), the IDF1 is significantly improved by 3.4 points. This improvement contributes 70% to the total improved 4.8 points IDF1. This experiment shows that more contrastive targets, even most of them are negative samples, can improve the feature learning

| Split | Method | AP50 | AP75 | AP | AP50(S) | AP50(M) | AP50(L) |
|-------|--------|------|------|---|--------|--------|--------|
| val   | SORT-TAO [27] | 13.2 | -    | -  | -      | -      | -      |
|       | QDTrack (Ours) | 16.1 | 5.0  | 7.0| 2.4    | 4.6    | 9.6    |
|       | GTR [76] † | 22.3 | -    | -  | -      | -      | -      |
|       | AOA [79] ‡ | 23.8 | -    | -  | -      | -      | -      |
| test  | SORT-TAO [27] | 10.2 | 4.4  | 4.9| 7.7    | 8.2    | 15.2   |
|       | QDTrack (Ours) | 12.4 | 4.5  | 5.2| 3.7    | 8.3    | 18.8   |
|       | GTR [76] † | 20.1 | -    | -  | -      | -      | -      |
|       | AOA [79] ‡ | 27.5 | -    | -  | -      | -      | -      |

We evaluate and compare our method on the TAO challenge benchmark. We use faster R-CNN [76] as our detection method. † indicates offline methods, ‡ indicates methods using additional data.
process. The multiple-positive contrastive learning following (4) further improves the IDF1 by 1 point (66.8% to 67.8%).

**Importance of bi-softmax:** We investigate how different inference strategies influence the performance. As shown in the bottom of Table VIII, replacing cosine similarity by bi-softmax improves overall IDF1 by 2.2 points and the IDF1 of pedestrian by 4.5 points. This experiment also shows that the one-to-one constraint further strengthens the estimated similarity.

**Importance of matching candidates:** Duplicate removal and backdrops improve IDF1 by 1.5 points. Overall, our training and inference strategies improve the IDF1 by 8.5 points (63.0% to 71.5%). The total number of ID switches is decreased by 30%. Especially, the MOTA and IDF1 of pedestrian are improved by 9.1 points and 10.5 points respectively, which further demonstrate the power of quasi-dense contrastive learning.

**Combinations with motion and location:** Finally, we try to add location and motion priors to understand whether they are still helpful when we have good feature embeddings for measuring similarity. These experiments follow the procedures in Tracktor [1] and use the same Faster R-CNN detector for fair comparisons. We use the BDD100 K and MOT17 datasets to show the influence of motion cues under varying video frame-rates and camera motion patterns. As shown in Table IX, without appearance features, we observe that motion cues yield good performance on the high frame-rate MOT17 dataset, but exhibit significantly degraded performance on the more challenging BDD100 K dataset. Further, when combined with our appearance features, these cues barely enhance the performance of our approach. Our method yields the best results when only using appearance embeddings on both datasets. The results indicate that our instance feature embeddings are sufficient for multiple object tracking with the effective quasi-dense matching, which greatly simplifies the inference pipeline.

**Inference speed:** To understand the runtime efficiency, we profile our method on a single NVIDIA RTX 3090 graphics card. Because it only adds a lightweight embedding head to the detector, our method only causes marginal overhead in inference speed. With an input size of $1280 \times 736$ and a Faster R-CNN detector with ResNet-50 base model on BDD100 K, the inference time is 61 ms, equating to 16.3 FPS. With the stronger YOLOX-X detector, the inference time is 74 ms, equating to 13.5 FPS. However, the embedding extractor consumes only 3 ms in both cases, representing only 5% of the total runtime. Note that we measure runtime at 32-bit floating point precision.

**G. Embedding Visualizations**

We use t-SNE to visualize the embeddings trained with sparse matching and our quasi-dense matching and show them in Fig. 4. The instances are selected from a video in BDD100 K tracking validation set.

**H. Segmentation Tracking**

Owing to the simplicity of our method, we can extend it to instance segmentation tracking in a straightforward manner. To do so, we simply add a Mask R-CNN [61] mask prediction head to the existing network architecture and use a pre-trained QDTrack model trained on the BDD100 K tracking set to fine-tune the mask head on MOTS data. In particular, BDD100 K provides a subset for the segmentation tracking task. There are 154 videos in the training set, 32 videos in the validation set,
TABLE VIII
ABLATION STUDY ON QUASI-DENSE MATCHING AND INFERENCE STRATEGY

| Quasi-Dense matching & | Metric | Matching candidates | Backdrops | MOTA ↑ | IDFI ↑ | mMOTA ↑ | mIDFI ↑ | MOTA(P) ↑ | IDF1(P) ↑ |
|------------------------|--------|---------------------|------------|--------|--------|---------|---------|-----------|-----------|
| one-positive            | cosine | -                   | -          | 60.4   | 63.0   | 34.0    | 47.9    | 37.6      | 49.7      |
| -                      | cosine | -                   | -          | 61.5   | 66.8   | 35.5    | 50.0    | 40.5      | 52.7      |
| -                      | cosine | -                   | -          | 62.5   | 67.8   | 36.2    | 50.0    | 44.0      | 54.3      |
| -                      | bi-softmax | -                   | -          | 62.9   | 70.0   | 35.4    | 48.5    | 45.5      | 58.8      |
| -                      | bi-softmax | ✓                 | ✓          | 63.2   | 70.1   | 36.4    | 50.4    | 45.5      | 58.3      |
| ✓                      | bi-softmax | ✓                 | ✓          | 63.5   | 71.5   | 36.6    | 50.8    | 46.7      | 60.2      |

We investigate the contribution of various components on the BDD100K tracking validation set. All models are comparable on detection performance. D. R. Means duplicate removal. (P) means results of the class “pedestrian”.

Fig. 4. Instance embedding space visualization. We visualize the instance embedding space learned via (a) sparse matching and (b) quasi-dense matching using t-SNE. We show ground truth embedding identities as color and plot embedding vectors sampled from a sequence in the BDD100K tracking validation set.

Fig. 5. Illustration of failure cases. We illustrate the two most common failure cases of our method (best viewed digitally). In the top, we can see that the bus (light green and violet) switches identity due to extreme occlusion by pedestrians. In the bottom, we observe that the pickup truck (green and purple) switches identity when the class prediction changes between ‘truck’ and ‘car’. Note that we still re-identify the pickup truck once the class predictions match.

TABLE IX
ABLATION STUDY ON LOCATION AND MOTION CUES

| Appearance | Detection | Regression | BDD100K | MOT17 |
|------------|-----------|------------|---------|-------|
| -          | ✓         | -          | 26.3    | 72.8  |
| -          | ✓         | ✓          | 27.7    | 73.2  |
| -          | ✓         | ✓          | 28.4    | 73.2  |
| ✓          | ✓         | -          | 36.6    | 73.3  |
| ✓          | ✓         | ✓          | 36.5    | 73.3  |
| ✓          | ✓         | ✓          | 36.5    | 73.3  |

We investigate if our method benefits from using a range of motion priors on the BDD100K and MOT17 validation sets. We integrate bounding box IoU, a simple linear motion model, and displacement regression into the association procedure.

and 37 videos in the test set. Table X shows the results on the BDD100K segmentation tracking task compared to other methods. QDTrack achieves 25.6 mMOTSA and 45.2 mIDF1.

PCAN [18] is an extension of QDTrack that utilizes a prototypical appearance module to further improve segmentation. We observe that QDTrack based models achieve much better performance than previous methods.

1. Limitations

While our method gains in simplicity and generality by solely relying on instance similarity learning, we also identify certain challenges that arise with this paradigm. In particular, we observe that our model struggles with rapid changes in object appearance, e.g., through partial occlusion. Also, since our model relies on discrete class labels to aid the matching process, we observe that classification errors can lead to truncated object tracks. These cases are illustrated in Fig. 5. In addition, inaccurate object localization can lead to difficulties in association when
regions within a bounding box cover background and/or other objects, thus impeding accurate instance embedding extraction. For more detailed failure case and oracle analysis, please refer to the appendix, available online.

V. CONCLUSION

We present QDTrack, a tracking method based on quasi-dense instance similarity learning. The key idea behind our method is to utilize all object regions in an image for similarity learning, in contrast to previous methods that only use sparse ground-truth regions as similarity supervision. We observe that the feature embedding space we learn from quasi-dense matches is much better suited to discriminate instances, allowing for a simple tracking framework that associates objects via nearest neighbor search in the embedding space without bells and whistles. Our method can be easily coupled with most existing object detectors and feature extractors for end-to-end training, and learns effective instance similarity even without video input or tracking annotations.

REFERENCES

[1] P. Bergmann, T. Meinhardt, and L. Leal-Taixé, “Tracking without bells and whales,” 2019, arXiv:1903.05625.
[2] A. Bewley, Z. Ge, L. Ott, F. T. Ramos, and B. Upcroft, “Simple online and realtime tracking,” in Proc. Int. Conf. Image Process., 2016, pp. 3464–3468.
[3] N. Wojke, A. Bewley, and D. Paulus, “Simple online and realtime tracking with a deep association metric,” in Proc. Int. Conf. Image Process., 2017, pp. 3645–3649.
[4] X. Zhou, V. Koltun, and P. Krähenbühl, “Tracking objects as points,” in Proc. Eur. Conf. Comput. Vis., 2020, pp. 474–490.
[5] Z. Wang, L. Zheng, Y. Liu, and S. Wang, “Towards real-time multi-object tracking,” 2019, arXiv:1909.12605.
[6] D. Ramanan and D. A. Forsyth, “Finding and tracking people from the bottom up,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2003, pp. 467–474.
[7] E. Bochinski, V. Eiselein, and T. Sikora, “High-speed tracking-by-detection without using image information,” in Proc. IEEE Int. Conf. Adv. Video Signal Based Survell., 2017, pp. 1–6.
[8] A. Milan, S. Roth, and K. Schindler, “Continuous energy minimization for multitarget tracking,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 36, no. 1, pp. 58–72, Jan. 2013.
[9] W. Choi and S. Savarese, “Multiple target tracking in world coordinate with single, minimized calibrated camera,” in Proc. Eur. Conf. Comput. Vis., 2010, pp. 553–567.
[10] C. Feichtenhofer, A. Pinz, and A. Zisserman, “Detect to track and track to detect,” in Proc. IEEE Int. Conf. Comput. Vis., 2017, pp. 3057–3065.
[11] J. Peng et al., “Chained-tracker: Chaining paired attentive regression results for end-to-end joint multiple-object detection and tracking,” in Proc. Eur. Conf. Comput. Vis., 2020, pp. 145–161.
[12] Z. Lu, V. Rathod, R. Votell, and J. Huang, “Reinatrack: Online single stage joint detection and tracking,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2020, pp. 14656–14666.
[13] A. v. d. Oord, Y. Li, and O. Vinyals, “Representation learning with contrastive predictive coding,” 2018, arXiv:1807.03749.
[42] A. Sadeghian, A. Alahi, and S. Savarese, “Tracking the untrackable: Learning to track multiple cues with long-term dependencies,” in Proc. IEEE Int. Conf. Comput. Vis., 2017, pp. 300–311.

[43] A. Milan, S. H. Rezatofighi, A. R. Dick, I. D. Reid, and K. Schindler, “Online multi-target tracking using recurrent neural networks,” in Proc. AAAI Conf. Artif. Intell., 2017, pp. 4225–4232.

[44] Y. Zhang et al., “Bytetrack: Multi-object tracking by associating every detection box,” 2021, pp. arXiv:2110.06864.

[45] J. Cao, X. Weng, R. Khridokar, J. Pang, and K. Kitani, “Observation-centric sort: Rethinking sort for robust multi-object tracking,” 2022, pp. arXiv:2202.12277.

[46] D. Held, J. Levinson, and S. Thrun, “Precision tracking with sparse 3D and dense color 2D data,” in Proc. IEEE Int. Conf. Robot. Automat., 2013, pp. 1138–1145.

[47] L. Yang, Y. Fan, and N. Xu, “Video instance segmentation,” in Proc. IEEE Int. Conf. Comput. Vis., 2019, pp. 5187–5196.

[48] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu, “FairMOT: On the fairness of detection and re-identification in multiple object tracking,” 2020, arXiv:2004.01888.

[49] A. Hermans, L. Beyer, and B. Leibe, “In defense of the triplet loss for person re-identification,” 2017, pp. arXiv:1703.07737.

[50] Y. Zhou, J. Gao, L. Xi, and W. Hu, “Anchor-free one-stage online multi-object tracking,” in Proc. Pattern Recognit. Comput. Vis., 2019, pp. 970–978.

[51] Y. Peng, Q. Liu, H. Liu, Z. C. Sun, X. Liu, H. Chen Zha, and J. Yang, Eds., Cham, Springer International Publishing, 2020, pp. 55–68.

[52] T. Meinhardt, A. Kirillov, L. Leal-Taixe, and C. Feichtenhofer, “Trackerformer: Multi-object tracking with transformers,” 2021, arXiv:2101.02702.

[53] F. Zeng, B. Dong, T. Wang, X. Zhang, and Y. Wei, “MOTR: End-to-end multiple-object tracking with transformer,” 2021, pp. arXiv:2105.03247.

[54] P. Sun et al., “Transtrack: Multi-object tracking with transformer,” 2020, arXiv:2012.15460.

[55] P. Bachman, R. D. Hjelm, and W. Buchwalter, “Learning representations by maximizing mutual information across views,” in Proc. Adv. Neural Inf. Process. Syst., 2019, pp. 15509–15519.

[56] O. J. Hénaff et al., “Data-efficient image recognition with contrastive predictive coding,” 2019, arXiv:1905.09272.

[57] Y. Tian, D. Krishnan, and P. Isola, “Contrastive multiview coding,” 2019, arXiv:1906.05849.

[58] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, “Momentum contrast for unsupervised visual representation learning,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2020, pp. 9726–9735.

[59] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A simple framework for contrastive learning of visual representations,” 2020, arXiv:2002.05709.

[60] Y. Sun et al., “Circle loss: A unified perspective of pair similarity optimization,” 2020, arXiv:2002.10857.

[61] T.-Y. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, “Feature pyramid networks for object detection,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 936–944.

[62] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN,” in Proc. IEEE Int. Conf. Comput. Vis., 2017, pp. 2980–2988.

[63] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr, “Fully-convolutional siamese networks for object tracking,” in Proc. Eur. Conf. Comput. Vis., 2016, pp. 850–865.

[64] P. Dendorfer et al., “MOT20: A benchmark for multi object tracking in crowded scenes,” 2020, arXiv:2003.09003.

[65] J. Hyun, M. Kang, D. Wee, and D.-Y. Yeung, “Detection recovery in online multi-object tracking with sparse graph tracker,” 2022, pp. arXiv:2203.00968.

[66] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, “YOLOX: Exceeding YOLO series in 2021,” 2021, arXiv:2107.08430.

[67] J. Wu, J. Cao, L. Song, Y. Wang, M. Yang, and J. Yuan, “Track to detect and segment: An online multi-object tracker,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2021, pp. 12352–12361.

[68] X. Zhou, T. Yin, V. Koltun, and P. Krähenbühl, “Global tracking transformers,” 2022, arXiv:2203.13250.

[69] S. Ren, K. He, R. Girshick, and J. Sun, “ Faster R-CNN: Towards real-time object detection with region proposal networks,” in Proc. Adv. Neural Inf. Process. Syst., 2015, pp. 91–99.

[70] C. Kim, F. Li, and J. M. Rehg, “Multi-object tracking with neural gating using bilinear LSTM,” in Proc. Eur. Conf. Comput. Vis., 2018, pp. 208–224.

[71] F. Du, B. Xu, J. Tang, Y. Zhang, F. Wang, and H. Li, “1st place solution to ECCV-TAO-2020: Detect and represent any object for tracking,” 2021, arXiv:2101.08040.

[72] A. Gupta, P. Dollar, and R. Girshick, “LVIS: A dataset for large vocabulary instance segmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 5356–5364.

[73] K. Bernardin and R. Stiefelhagen, “Evaluating multiple object tracking performance: The clear MOT metrics,” EURASIP J. Image Video Process., vol. 2008, 2008, Art. no. 246309.

[74] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, “Performance measures and a data set for multi-target, multi-camera tracking,” in Proc. Eur. Conf. Comput. Vis., Springer, 2016, pp. 17–35.

[75] J. Luiten et al., “HOTA: A higher order metric for evaluating multi-object tracking,” Int. J. Comput. Vis., vol. 129, no. 2, pp. 548–578, 2021.

[76] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in Proc. IEEE Int. Conf. Comput. Vis., 2017, pp. 2999–3007.

[77] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, and D. Lin, “Libra R-CNN: Towards balanced learning for object detection,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 821–830.

[78] Y. Wu and K. He, “Group normalization,” in Proc. Eur. Conf. Comput. Vis., 2018, pp. 3–19.

[79] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, “mixup: Beyond empirical risk minimization,” 2017, arXiv:1710.09412.

[80] J. Dai et al., “Deformable convolutional networks,” in Proc. IEEE Int. Conf. Comput. Vis., 2017, pp. 764–773.

[81] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 770–778.

[82] C.-Y. Wang, H.-Y. Mark Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and C.-Y. Wang, “CSPNet: A new backbone that can enhance learning capability of CNN,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops, 2020, pp. 390–391.

[83] A. Athar, S. Mahadevan, A. Osep, L. Leal-Taixé, and B. Leibe, “STEm-Seg: Spatio-temporal embeddings for instance segmentation in videos,” in Proc. Eur. Conf. Comput. Vis., Springer, 2020, pp. 158–177.

Tobias Fischer received the BSc and MSc degrees in computer science from RWTH Aachen University. He is currently working toward the PhD degree in computer vision supervision by Prof. Dr. Fisher Yu with ETH Zurich, Switzerland. He worked on 3D reconstruction and tracking with the Visual Computing institute. His research interests lie in the area of 3D scene understanding, with a focus on retrieving 3D information from visual perception for object recognition, tracking, and reconstruction.
Thomas E. Huang received the BSE and MSE degrees in computer science from the University of Michigan, Ann Arbor, in 2019 and 2021, respectively. He is working toward the PhD degree in computer vision supervised by Prof. Dr. Fisher Yu with ETH Zurich, Switzerland. His broad research interests are deep learning and its applications to computer vision. Recently, his research focuses on robust and sample-efficient learning, multi-object tracking, and multi-task learning.

Jiangmiao Pang (Member, IEEE) received the PhD degree from Zhejiang University and the postdoc degree from MMLab and CUHK. He is a research scientist with Shanghai AI Laboratory. He has worked with Dr. Jianping Shi, SenseTime, Prof. Fisher Yu and Prof. Trevor Darrell, UC Berkeley, and Prof. Dahua Lin, MMLab during his postgraduate study. His research is at the junction of machine learning, computer vision, and robotics. His long-term research goal is to build intelligent systems that can perceive and act like human beings.

Linlu Qiu received the BEng degree from the University of Hong Kong, and the MS degree from the Georgia Institute of Technology. She is working toward the PhD degree with the Massachusetts Institute of Technology. She worked as an AI resident with Google Research. Her recent research lies in the intersection of natural language processing and machine learning, with a focus on compositional generalization and grounded language understanding.

Haofeng Chen received the bachelor of science degree in computer science from the Georgia Institute of Technology, and the master of science degree in computer science from Stanford University. He worked with the Stanford Vision and Learning Lab on video understanding. Prior to Stanford, he worked with Berkeley DeepDrive on data annotation, 2D object detection, segmentation, and tracking.

Trevor Darrell received the BSE degree from the University of Pennsylvania, in 1988, and the SM and PhD degrees from MIT, in 1992 and 1996, respectively. He is an professor on the faculty of the CS and EE Divisions of the EECS Department at UC Berkeley. He founded and coleads Berkeley’s Berkeley Artificial Intelligence Research (BAIR) lab, the Berkeley DeepDrive (BDD) Industrial Consortium, and the recently launched BAIR Commons program in partnership with Facebook, Google, Microsoft, Amazon, and other partners. He also is faculty director of the PATH research center, UC Berkeley, and previously led the Vision group, UCaffiliated International Computer Science Institute in Berkeley. Prior to that, he was on the faculty of the MIT EECS department from 1999-2008, where he directed the Vision Interface Group. He was a member of the research staff at Interval Research Corporation from 1996-1999.

Fisher Yu (Member, IEEE) received the PhD degree from Princeton University and became a postdoctoral researcher with UC Berkeley afterwards. He is an assistant professor with ETH Zurich, Switzerland. He directs the Visual Intelligence and Systems (VIS) Group. His research is at the junction of machine learning, computer vision, and robotics.