Quasi-Dense Similarity Learning for Multiple Object Tracking

Jiangmiao Pang[1] Linlu Qiu[2] Xia Li[3] Haofeng Chen[4] Qi Li[1] Trevor Darrell[5] Fisher Yu[3]

[1]Zhejiang University [2]Georgia Institute of Technology [3]ETH Zürich
[4]Stanford University [5]UC Berkeley

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 on the images. In this paper, we present Quasi-Dense Similarity Learning, which densely samples hundreds of region proposals on a pair of images for contrastive learning. We can directly combine this similarity learning with existing detection methods to build Quasi-Dense Tracking (QDTrack) without turning to displacement regression or motion priors. We also find that the resulting distinctive feature space admits a simple nearest neighbor search at the inference time. Despite its simplicity, QDTrack outperforms all existing methods on MOT, BDD100K, Waymo, and TAO tracking benchmarks. It achieves 68.7 MOTA at 20.3 FPS on MOT17 without using external training data. Compared to methods with similar detectors, it boosts almost 10 points of MOTA and significantly decreases the number of ID switches on BDD100K and Waymo datasets. Our code and trained models are available at https://github.com/SysCV/qdtrack.

1. 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 [2, 4, 44, 45, 54] mainly follow the tracking-by-detection paradigm [36]. That is, they detect objects on each frame and then associate them according to the estimated instance similarity. Recent works [2, 4, 5, 54] show that if the detected objects are accurate, the spatial proximity between objects in consecutive frames, measured by Interaction of Unions (IoUs) or center distances, is a strong prior to associate the objects. However, this location heuristic only works well in simple scenarios. If the objects are occluded or the scenes are crowded, this location heuristic can easily lead to mistakes. To remedy this problem, some methods introduce motion estimation [7, 30] or displacement regression [10, 35, 54] to ensure accurate distance estimation.

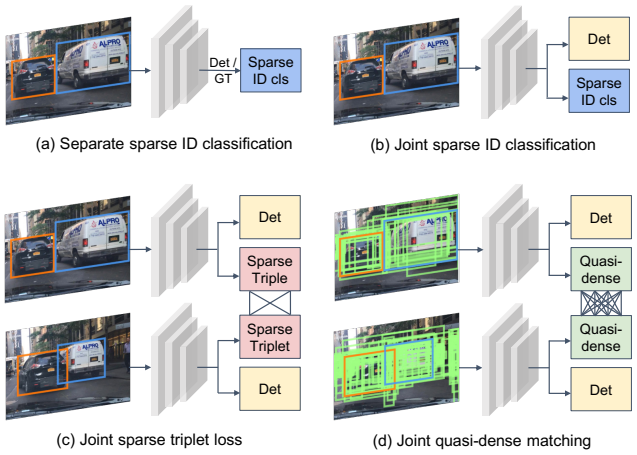


Figure 1: (a) Traditional ReID model that decouples with detector and learns with sparse ID loss; (b) joint learning ReID model with sparse ID loss; (c) joint learning ReID model with sparse triplet loss; (d) our quasi-dense similarity learning.

However, object appearance similarity usually takes a secondary role [26, 45] to strengthen object association or re-identify vanished objects. The search region is constrained to be local neighborhoods to avoid distractions because the appearance features can not effectively distinguish different objects. On the contrary, humans can easily associate the identical objects only through appearance. We conjecture this is because the image and object information is not fully utilized for learning object similarity. As shown in Figure 1, previous methods regard instance similarity learning as a post hoc stage after object detection or only use sparse ground truth bounding boxes as training samples [45]. These processes ignore the majority of the regions proposed on the images. Because objects in an image are rarely iden-

tical to each other, if the object representation is properly learned, a nearest neighbor search in the embedding space should associate and distinguish instances without bells and whistles.

We observe that besides the ground truths and detected bounding boxes, which sparsely distribute on the images, many possible object regions can provide valuable training supervision. They are either close to the ground truth bounding boxes to provide more positive training examples or in the background as negative examples. In this paper, we propose quasi-dense similarity learning, which densely matches hundreds of regions of interest on a pair of images for contrastive learning. The quasi-dense samples can cover most of the informative regions on the images, providing both more box examples and hard negatives.

Because one sample has more than one positive samples on the reference image, we extend the contrastive learning [12, 39, 47] to multiple positive forms that makes the quasi-dense learning feasible. Each sample is thus trained to distinguish all proposals on the other image simultaneously. This contrast provides stronger supervision than using only the handful ground truth labels and enhances the instance similarity learning.

The inference process, which maintains the matching candidates and measures the instance similarity, also plays an important role in the tracking performance. Besides similarity, MOT also needs to consider false positives, id switches, new appeared objects, and terminated tracks. To tackle the missing targets with our similarity metric, we include backdrops, the unmatched objects in the last frame, for matching and use *bi-directional softmax* to enforce the bi-directional consistency. The objects that do not have matching targets will lack the consistency thus has low similarity scores to any objects. To track the multiple targets, we also conduct duplicate removal to filter the matching candidates.

Quasi-dense similarity learning can be easily used with most existing detectors since generating region of interests is widely used in object detection algorithms. In this paper, we apply our method to Faster R-CNN [37] along with a lightweight embedding extractor and residual networks [15] and build *Quasi-Dense Tracking* (QDTrack) models. We conduct extensive experiments on MOT [28], BDD100K [51], Waymo [41], and TAO [8] tracking benchmarks. Despite its simplicity, QDTrack outperforms all existing methods without bells and whistles. It achieves 68.7 MOTA on MOT17 at 20.3 FPS without using external training data. Moreover, it boosts almost 10 points of MOTA and significantly decreases the number of ID switches on BDD100K and Waymo datasets, establishing solid records on these brand-new large-scale benchmarks. QDTrack allows end-to-end training, thereby simplifying the training and testing procedures of multi-object tracking frameworks. The simplicity and effectiveness shall benefit further research.

2. Related work

Recent developments in multiple object tracking [23] follow the tracking-by-detection paradigm [36]. These approaches present different methods to estimate the instance similarity between detected objects and previous tracks, then associate objects as a bipartite matching problem [31].

Location and motion in MOT The spatial proximity has been proven effective to associate objects in consecutive frames [4, 5]. However, they cannot do well in complicated scenarios such as crowd scenes. Some methods use motion priors, such as Kalman Filter [4, 52], optical flow [48], and displacement regression [10, 16], to ensure accurate distance estimations. In contrast to the old paradigm that detects objects and predicts displacements separately, Detect & Track [10] is the first work that jointly optimizes object detection and tracking modules. It predicts the displacements of the objects in consecutive frames and associates the objects with the Viterbi algorithm. Tracktor [2] directly adopts a detector as a tracker. CenterTrack [54] and Chained-Tracker [35] predict the object displacements with pair-wise inputs to associate the objects. Although these methods show promising results, they [2, 45] still need an extra re-identification model as complementary to re-identify vanished objects, making the entire framework complicated.

Appearance similarity in MOT To exploit instance appearance similarity to strengthen tracking and re-identify vanished objects, some methods directly use an independent model [2, 20, 22, 29, 38, 40, 45, 49] or add an extra embedding head to the detector for end-to-end training [26, 44, 50, 53]. However, they still learn the appearance similarity following the practice in image similarity learning, then measure the instance similarity by cosine distance. That is, they train the model either as a n-classes classification problem [45] where n equals to the number of identities in the whole training set or using triplet loss [18]. The classification problem is hard to extend to large-scale datasets, while the triplet loss only compares each training sample with two other identities. These rudimentary training samples and objectives leave instance similarity learning not fully explored in MOT. Meanwhile, they still heavily rely on motion models and displacement predictions to track objects, and the appearance similarity only takes the secondary role.

In contrast to these methods, QDTrack learns the instance similarity from dense-connected contrastive pairs and associates objects from the feature space with a simple nearest neighbor search. QDTrack has higher performance but with a simpler framework. The promising results prove the power of quasi-dense similarity learning in multiple object tracking.

Contrastive learning Contrastive learning and its variants [1, 6, 13, 17, 32, 42, 43, 47] have shown promising performance in self-supervised representation learning. However, it does not draw much attention when learning the instance similarity in multiple object tracking. In this paper, we

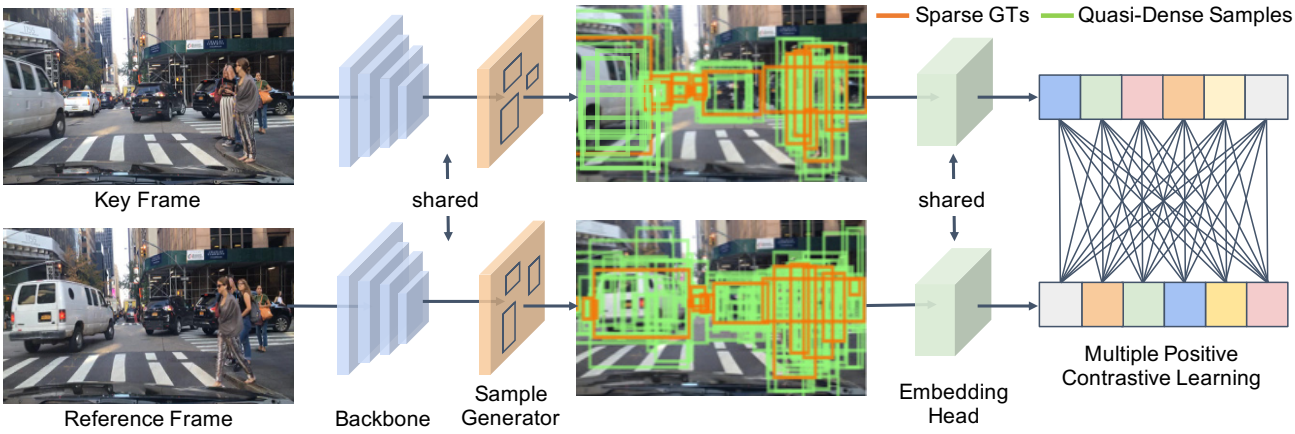


Figure 2: The training pipeline of our method. We apply dense matching between quasi-dense samples on the pair of images and optimize the network with multiple positive contrastive learning.

supervise dense matched quasi-dense samples with multiple positive contrastive learning by the inspiration of [42]. In contrast to these image-level contrastive methods, our method allows multiple positive training, while these methods can only handle the case when there is only one positive target. The promising results of our method shall draw the attention to contrastive learning in the multiple object tracking community.

3. Methodology

We propose *quasi-dense similarity learning* to learn the 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 box candidates at all pixels, and *quasi-dense* means only considering the potential object candidates at informative regions. Accordingly, *sparse matching* means the method only considers ground truth labels as matching candidates when learning object association. The main ingredients of using quasi-dense matching for multiple object tracking are object detection, instance similarity learning, and object association.

3.1. Object detection

Our method can be easily coupled with most existing detectors with end-to-end training. In this paper, we take Faster R-CNN [37] with Feature Pyramid Network (FPN) [24] as an example, while we can also apply other detectors with minor modifications. Faster R-CNN is a two-stage detector that uses Region Proposal Network (RPN) to generate Region of Interests (RoIs). It then localizes and classifies the regions to obtain semantic labels and locations. Based on Faster R-CNN, FPN exploits lateral connections to build the top-down feature pyramid and tackles the scale-variance problem. The entire network is optimized with a multi-task loss function

$$\mathcal{L}_{\text{det}} = \mathcal{L}_{\text{rpn}} + \lambda_1 \mathcal{L}_{\text{cls}} + \lambda_2 \mathcal{L}_{\text{reg}}, \tag{1}$$

where the RPN loss $\mathcal{L}_{\text{rpn}}$, classification loss $\mathcal{L}_{\text{cls}}$, regression loss $\mathcal{L}_{\text{reg}}$ remain the same as the original paper [37]. The loss weights λ_1 and λ_2 are set to 1.0 by default.

3.2. Quasi-dense similarity learning

We use the region proposals generated by RPN to learn the instance similarity with quasi-dense matching. As shown in Figure 2, given a key image I_1 for training, we randomly select a reference image I_2 from its temporal neighborhood. The neighbor distance is constrained by an interval k, where $k \in [-3, 3]$ in our experiments. We use RPN to generate RoIs from the two images and RoI Align [14] to obtain their feature maps from different levels in FPN according to their scales [24]. We add an extra lightweight embedding head, in parallel with the original bounding box head, to extract features for each RoI. An RoI is defined as positive to an object if they have an IoU higher than α_1, or negative if they have an IoU lower than α_2. α_1 and α_2 are 0.7 and 0.3 in our experiments. The matching of RoIs on two frames is positive if the two regions are associated with the same 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 [32, 47] with cross-entropy to optimize the feature embeddings

$$\mathcal{L}_{\text{embed}} = -\log \frac{\exp(\mathbf{v} \cdot \mathbf{k}^+)}{\exp(\mathbf{v} \cdot \mathbf{k}^+) + \sum_{\mathbf{k}^-} \exp(\mathbf{v} \cdot \mathbf{k}^-)}, \tag{2}$$

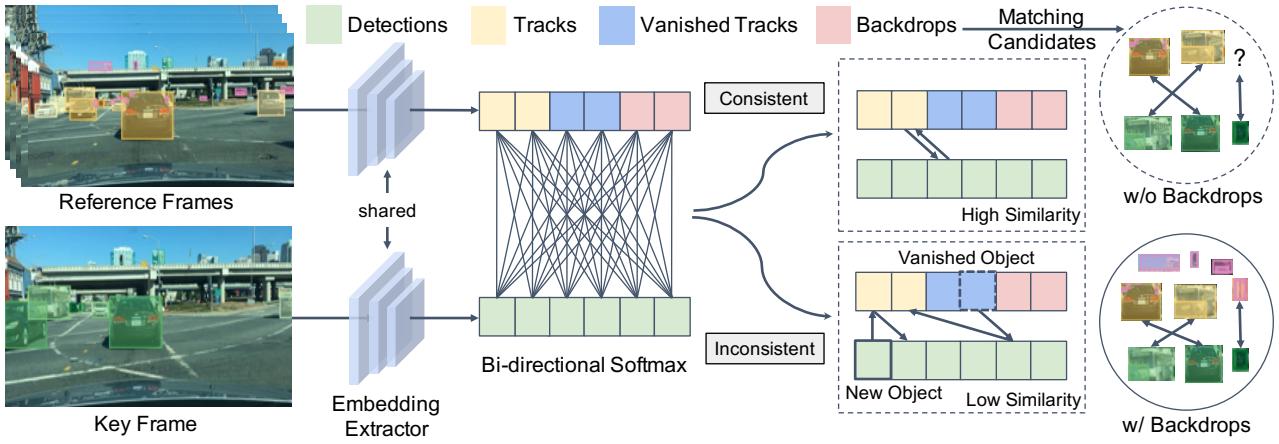


Figure 3: The testing pipeline of our method. We maintain the matching candidates and use bi-softmax to measure the instance similarity so that we can associate objects with a simple nearest neighbour search in the feature space.

where $\mathbf{v}$, $\mathbf{k}^+$, $\mathbf{k}^-$ 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 simplicity.

We apply dense matching between RoIs on the pairs of images, namely, each sample on I_1 is matched to all samples on I_2, in contrast to only using sparse sample crops, mostly ground truth boxes, to learn instance similarity in previous works [3, 18]. Each training sample on the key frame has more than one positive targets on the reference frame, so Eq. (2) can be extended as

$$\mathcal{L}_{\text{embed}} = -\sum_{\mathbf{k}^+} \log \frac{\exp(\mathbf{v}\cdot\mathbf{k}^+)}{\exp(\mathbf{v}\cdot\mathbf{k}^+) + \sum_{\mathbf{k}^-}\exp(\mathbf{v}\cdot\mathbf{k}^-)}. \quad (3)$$

However, this equation does not treat positive and negative targets fairly. Namely, each negative one is considered multiple times while only once for positive counterparts. Alternatively, we can first reformulate Eq. (2) as

$$\mathcal{L}_{\text{embed}} = \log[1 + \sum_{\mathbf{k}^-} \exp(\mathbf{v}\cdot\mathbf{k}^- - \mathbf{v}\cdot\mathbf{k}^+)]. \quad (4)$$

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

$$\mathcal{L}_{\text{embed}} = \log[1 + \sum_{\mathbf{k}^+}\sum_{\mathbf{k}^-} \exp(\mathbf{v}\cdot\mathbf{k}^- - \mathbf{v}\cdot\mathbf{k}^+)]. \quad (5)$$

We further adopt L2 loss as an auxiliary loss

$$\mathcal{L}_{\text{aux}} = (\frac{\mathbf{v}\cdot\mathbf{k}}{||\mathbf{v}||\cdot||\mathbf{k}||} - c)^2, \quad (6)$$

where c is 1 if the match of two samples is positive and 0 otherwise. Note the auxiliary loss aims to constrain the logit magnitude and cosine similarity instead of improving the performance.

The entire network is joint optimized under

$$\mathcal{L} = \mathcal{L}_{\text{det}} + \gamma_1\mathcal{L}_{\text{embed}} + \gamma_2\mathcal{L}_{\text{aux}}, \quad (7)$$

where γ_1 and γ_2 are set to 0.25 and 1.0 by default in this paper. We sample all positive pairs and three times more negative pairs to calculate the auxiliary loss.

3.3. Object association

Tracking objects across frames purely based on object feature embeddings is not trivial. For example, if an object has no target or more than one target during matching, the nearest search will be ambiguous. In other words, an object should have only one target in the matching candidates. However, the actual tracking process is complex. The false positives, id switches, newly appeared objects, and terminated tracks all increase the matching uncertainty. We observe that our inference strategy, including ways of maintaining the matching candidates and measuring the instance similarity, can mitigate these problems.

Bi-directional softmax Our main inference strategy is bi-directional matching in the embedding space. Figure 3 shows our testing pipeline. 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 similarity $\mathbf{f}$ between the objects and matching candidates is obtained by bi-directional softmax (bi-softmax):

$$\mathbf{f}(i,j) = [\frac{\exp(\mathbf{n}_i\cdot\mathbf{m}_j)}{\sum_{k=0}^{M-1}\exp(\mathbf{n}_i\cdot\mathbf{m}_k)} + \frac{\exp(\mathbf{n}_i\cdot\mathbf{m}_j)}{\sum_{k=0}^{N-1}\exp(\mathbf{n}_k\cdot\mathbf{m}_j)}]/2. \quad (8)$$

The high score under bi-softmax will satisfy a bi-directional consistency. Namely, the two matched objects should be

Table 1: Results on MOT16 and MOT17 test set with private detectors. Note that we do not use extra data for training. ↑ means higher is better, ↓ means lower is better. * means external data besides COCO and ImageNet is used.

Dataset	Method	MOTA ↑	IDF1 ↑	MOTP ↑	MT ↑	ML ↓	FP ↓	FN ↓	IDs ↓
MOT16	TAP [55]	64.8	**73.5**	78.7	292	164	12980	50635	**571**
	CNNMTT [27]	65.2	62.2	78.4	246	162	6578	55896	946
	POI* [52]	66.1	65.1	**79.5**	258	158	**5061**	55914	3093
	TubeTK_POI* [33]	66.9	62.2	78.5	296	**122**	11544	47502	1236
	CTrackerV1 [35]	67.6	57.2	78.4	250	175	8934	48305	1897
	Ours	**69.8**	67.1	79.0	**316**	150	9861	**44050**	1097
MOT17	Tracktor++v2 [2]	56.3	55.1	78.8	498	831	**8866**	235449	1987
	Lif_T* [19]	60.5	65.6	78.3	637	791	14966	206619	**1189**
	TubeTK* [33]	63.0	58.6	78.3	735	**468**	27060	177483	4137
	CTrackerV1 [35]	66.6	57.4	78.2	759	570	22284	160491	5529
	CenterTrack* [54]	67.8	64.7	78.4	816	579	18498	160332	3039
	Ours	**68.7**	**66.3**	**79.0**	**957**	516	26589	**146643**	3378

each other's nearest neighbor in the embedding space. The instance similarity **f** can directly associate objects with a simple nearest neighbor search.

No target cases Objects without a target in the feature space should not be matched to any candidates. Newly appeared objects, vanished tracks, and some false positives fall into this category. The bi-softmax can tackle this problem directly, as it is hard for these objects to obtain bi-directional consistency, leading to low matching scores. If a newly detected object has high detection confidence, it can start a new track. Moreover, previous methods often directly drop the objects that do not match any tracks. We argue that despite most of them are false positives, they are still useful regions that the following objects are likely to match. We name these unmatched objects *backdrops* and keep them during matching. Experiments show that backdrops can reduce the number of false positives.

Multi-targets cases Most state-of-the-art detectors only do intra-class duplicate removal by None Maximum Suppression (NMS). Consequently, some objects at the same locations might have different categories. In most cases, only one of these objects is true positive while the others not. This process can boost the object recall and contribute to a high mean Average Precision (mAP) [9, 25]. However, it will create duplicate feature embeddings. To handle this issue, we do inter-class duplicate removal by NMS. The IoU threshold for NMS is 0.7 for objects with high detection confidence (larger than 0.5) and 0.3 for objects with low detection confidence (lower than 0.5).

4. Experiments

We conduct experiments not only on the MOT [28] benchmark but also on the other brand-new large-scale benchmarks including BDD100K [51], Waymo [41], and TAO [8]. We hope our efforts can facilitate future multiple object tracking research to benefit from these large-scale datasets. We also show the generalization ability of our method on BDD100K segmentation tracking benchmark. More results, such as oracle analyses and failure case analyses are presented in the supplementary material.

4.1. Datasets

MOT Challenge We perform experiments on two MOT benchmarks: MOT16 and MOT17 [28]. The dataset contains 7 videos (5,316 images) for training and 7 videos (5,919 images) for testing. Only pedestrians are evaluated in this benchmark. The video frame rate is 14 - 30 FPS.

BDD100K We use BDD100K [51] detection training set and tracking training set for training, and tracking validation/testing set for testing. It annotates 8 categories for evaluation. The detection set has 70,000 images. The tracking set has 1,400 videos (278k images) for training, 200 videos (40k images) for validation, and 400 videos (80k images) for testing. The images in the tracking set are annotated per 5 FPS with a 30 FPS video frame rate.

Waymo Waymo open dataset [41] 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 (790k images) for training, 1,010 videos (200k images) for validation, and 750 videos (148k images) for testing. It annotates 3 classes for evaluation. The videos are annotated in 10 FPS.

TAO TAO dataset [8] annotates 482 classes in total, which are the subset of LVIS dataset [11]. It has 400 videos, 216 classes in the training set, 988 videos, 302 classes in the validation set, and 1419 videos, 369 classes in the test set. The classes in train, validation, and test sets may not overlap. The videos are annotated in 1 FPS. The objects in TAO are in a long-tailed distribution that half of the objects are person

Table 2: Results on BDD100K tracking validation and test set. Our method outperforms all methods on this benchmark.

Method	Split	mMOTA ↑	mIDF1 ↑	MOTA ↑	IDF1 ↑	FN ↓	FP ↓	ID Sw. ↓	MT ↑	ML ↓	mAP ↑
Yu *et al*. [51]	val	25.9	44.5	56.9	66.8	122406	52372	8315	8396	3795	28.1
Ours	val	**36.6**	**50.8**	**63.5**	**71.5**	**108614**	**46621**	**6262**	**9481**	**3034**	**32.6**
Yu *et al*. [51]	test	26.3	44.7	58.3	68.2	213220	100230	14674	16299	6017	27.9
DeepBlueAI	test	31.6	38.7	56.9	56.0	292063	**35401**	25186	10296	12266	-
madamada	test	33.6	43.0	59.8	55.7	209339	76612	42901	16774	**5004**	-
Ours	test	**35.5**	**52.3**	**64.3**	**72.3**	**201041**	80054	**10790**	**17353**	5167	**31.8**

Table 3: Results on Waymo tracking validation set using py-motmetrics library (top) [1] and test set using official evaluation. * indicates methods using undisclosed detectors.

Method	Split	Category	MOTA ↑	IDF1 ↑	FN ↓	FP ↓	ID Sw. ↓	MT ↑	ML ↓	mAP ↑
IoU baseline [26]	val	Vehicle	38.25	-	-	-	-	-	-	45.78
Tracktor++ [2, 26]	val	Vehicle	42.62	-	-	-	-	-	-	42.41
RetinaTrack [26]	val	Vehicle	44.92	-	-	-	-	-	-	45.70
Ours	val	Vehicle	55.6	66.2	514548	214998	24309	17595	5559	49.5
Ours	val	All	44.0	56.8	674064	264886	30712	21410	7510	40.1

Method	Split	Category	MOTA/L1 ↑	FP/L1 ↓	MisM/L1 ↓	Miss/L1 ↓	MOTA/L2 ↑	FP/L2 ↓	MisM/L2 ↓	Miss/L2 ↓
Tracktor [21, 41]	test	Vehicle	34.80	10.61	14.88	39.71	28.29	8.63	12.10	50.98
CascadeRCNN-SORTv2*	test	All	50.22	7.79	2.71	39.28	44.15	**6.94**	2.44	46.46
HorizonMOT*	test	All	51.01	7.52	2.44	**39.03**	**45.13**	7.13	2.25	**45.49**
Ours (ResNet-50)	test	All	49.40	**7.41**	1.46	41.74	43.88	7.10	**1.31**	48.21
Ours (ResNet-101 + DCN)	test	All	**51.18**	7.64	**1.45**	39.73	45.09	7.20	**1.31**	46.41

and 1 / 6 of the objects are car.

4.2. Implementation details

We use ResNet-50 [15] as the backbone by default in this paper. 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 [34] to sample RoIs. We use *4conv-1fc* head with group normalization [46] to extract feature embeddings. The channel number of embedding features is set to 256 by default. We train our models with a total batch size of 16 and an initial learning rate of 0.02 for 12 epochs. We decrease the learning rate by 0.1 after 8 and 11 epochs.

Here, we first talk about the common practices if not specified mentioned afterwards. We use the original scale of the images for training and inference. We do not use any other data augmentation methods except random horizontal flipping. We use a model pre-trained on ImageNet for training. When conducting online joint object detection and tracking, we initialize a new track if its detection confidence is higher than 0.8. The backdrops are only kept for one frame. The objects can be associated only when they are classified as the same category.

For fair comparison with recent works, we follow the practice [44] on MOT17 that randomly resizes and crops the longer side of the images to 1088 and does not change the aspect ratio at the training and inference time. Other data augmentation includes random horizontal flipping and color jittering, which is the common practice in [35, 44, 54]. We do not use extra data for training except a pre-trained model from COCO. Note that COCO is not considered as additional training data by the official rules and widely used in most methods.

On TAO, we randomly select a scale between 640 to 800 to resize the shorter side of images during training. At inference time, the shorter side of the images are resized to 800. We use a LVIS [11] pre-trained model, consistent with the implementation of [8]. However, we observe severe over-fitting problem when training on the training videos of TAO, which hurts the detection performance. So we freeze the detection model and only fine-tune the embedding head to extract instance representations.

More details such as more hyper-parameters and momentum updating are presented in the supplementary material.

4.3. Main results

Our method outperforms all existing methods on aforementioned benchmarks without bells and whistles. The performance are evaluated with the official metrics.

MOT The results with private detectors on MOT16 and MOT17 benchmarks are shown in Table 1. Our model

[1] https://github.com/cheind/py-motmetrics

Table 4: Ablation studies on quasi-dense matching and the inference strategy 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".

Quasi-Dense		Metric	Matching candidates		MOTA ↑	IDF1 ↑	mMOTA ↑	mIDF1 ↑	MOTA(P) ↑	IDF1(P) ↑
one-positive	multi-positive		D. R.	Backdrops						
-	-	*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**
					+3.1	**+8.5**	**+2.6**	**+2.9**	**+9.1**	**+10.5**

Table 5: Ablations studies on location and motion cues on the BDD100K tracking validation set.

Appearance	IoU	Motion	Regression	mMOTA ↑	mIDF1 ↑
-	✓	-	-	26.3	36.0
-	✓	✓	-	27.7	38.5
-	✓	-	✓	28.6	39.3
✓	-	-	-	**36.6**	**50.8**
✓	✓	-	-	36.3	49.8
✓	✓	✓	-	36.4	49.9
✓	✓	-	✓	36.4	50.1

achieves the best MOTA of 68.7% and IDF1 of 66.3% on the MOT17. We outperform the state-of-the-art tracker CenterTrack [54] by 0.9 points on MOTA and 1.6 points on IDF1 respectively. Our method does not achieve a relatively low ID Sw. because we have a higher recall. The number of ID Sw. will likely increase when we have more tracks. This is also why the results with public detectors, which are shown in the supplementary material, have lower IDs, because their recall are lower (FN is higher).

BDD100K The main results on BDD100K tracking validation and testing sets are in Table 2. The mMOTA and mIDF1, which represent object coverage and identity consistency respectively, are 36.6% and 50.8% on the validation set, and 35.5% and 52.3% on the testing set. On the two sets, our method outperforms the baseline benchmark method by 10.7 points and 9.2 points in terms of mMOTA, and 6.3 points and 7.6 points in terms of mIDF1 respectively. We also outperform the champion of BDD100K 2020 MOT Challenge (madamada) by a large margin but with a simpler detector. The significant advancements demonstrate that our method enables more stable object tracking.

Waymo Table 3 shows our main results on Waymo open dataset. We report the results on the validation set following the setup of RetinaTrack [26], 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 a MOTA of 44.0% and a IDF1 of 56.8% on the validation set. We also obtain a MOTA/L1 of 49.40% and a MOTA/L2 of 43.88% on the test set. The performance of vehicle on the validation set is 10.7, 13.0, and 17.4 points higher than RetinaTrack [26], Tracktor++ [2, 26], and IoU baseline [26], respectively. Our model with ResNet-101 and deformable convolution (DCN) has the state-of-the-art performance on the test benchmark which is on par with the champion of Waymo 2020 2D Tracking Challenge (HorizonMOT) but only with a simple single model.

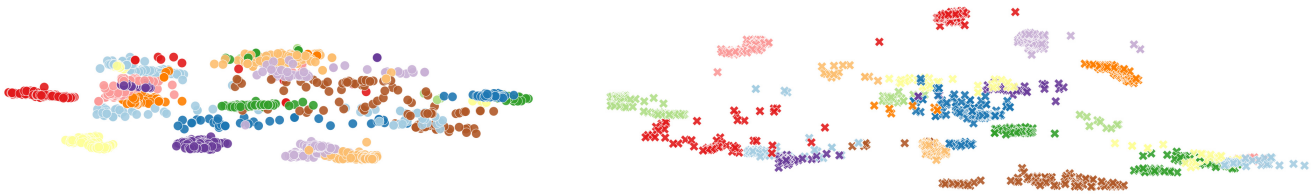
TAO We obtain 16.1 points and 12.4 points of AP50 on the validation and test set, respectively. The results are 2.9 points and 2.2 points higher than TAO's solid baseline, which are 13.2 points and 10.2 points respectively. Although we only boost the overall performance by 2 - 3 points, we observe that we outperform the baseline by a large margin on frequent classes, that is, 38.6 points vs. 18.5 points on person. This improvement is buried by the average across the entire hundreds of classes. It shows that the crucial part on TAO is still how to improve the tracking on tail classes, which should be a meaningful direction for further research. Other details are presented in the supplementary material.

4.4. Ablation studies

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

Importance of quasi-dense matching The results are presented in the top sub-table of Table 4. 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



(a) Sparse (b) Quasi-Dense

Figure 4: Visualizations of instance embeddings with (a) sparse matching and (b) quasi-dense matching using t-SNE.

Eq. (2), 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 process. The multiple-positive contrastive learning following Equation (5) 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 part of Table 4, 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 the location and motion priors to understand whether they are still helpful when we have good feature embeddings for similarity measure. These experiments follow the procedures in Tracktor [2] and use the same detector for fair comparisons. As shown in Table 5, without appearance features, the tracking performance is consistently improved with the introduction of additional information. However, these cues barely enhance the performance of our approach. Our method yields the best results when only using appearance embeddings. The results indicate that our instance feature embeddings are sufficient for multiple object tracking with the effective quasi-dense matching, which greatly simplify the testing pipeline.

Inference speed To understand the runtime efficiency, we profile our method on NVIDIA Tesla V100. Because it only adds a lightweight embedding head to Faster R-CNN, our method only bring marginal inference cost overhead. With an input size of 1296×720 and a ResNet-50 backbone on BDD100K, the inference FPS is 16.4. With an input size of 1088×608 and a ResNet-50 backbone on MOT17, the inference FPS is 20.3.

Table 6: Results on the BDD100K segmentation tracking test set. I: ImageNet. C: COCO. S: Cityscapes. B: BDD100K. "frozen" means adopting the pretrained model from the BDD100K tracking set and only finetune the mask head.

Method	Pretrained	mMOTSA ↑	mMOTSP ↑	mIDF1 ↑	ID sw. ↓
SORT [4]	I, C, S	12.8	67.3	28.8	3525
Ours	I, C, S	24.0	66.3	42.5	1581
Ours (frozen)	I, B	30.8	65.5	50.6	884

4.5. Embedding visualizations

We use t-SNE to visualize the embeddings trained with sparse matching and our quasi-dense matching and show them in Figure 4. The instances are selected from a video in BDD100K tracking validation set. The same instance is shown with the same color. We observe that it is easier to separate objects in the feature space of quasi-dense matching. More visualizations are presented in the supplementary material.

4.6. Segmentation tracking

We show the generalization ability of our method by extending it to instance segmentation tracking. BDD100K provides a subset for the segmentation tracking task. There are 154 videos in the training set, 32 videos in the validation set, and 37 videos in the test set. Table 6 shows the results on BDD100K segmentation tracking task. The results on the validation set are presented in the supplementary material.

5. Conclusion

We present QDTrack, a tracking method based on quasi-dense matching for instance similarity learning. In contrast to previous methods that use sparse ground-truth matching as similarity supervision, we learn instance similarity from hundreds of region proposals on pairs of images, and train the feature embeddings with multiple positive contrastive learning. In the resulting feature space, a simple nearest neighbor search can distinguish instances without bells and whistles. Our method can be easily coupled with most of the existing detectors and trained end-to-end for multiple object tracking and segmentation tracking.

References

[1] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In *Advances in Neural Information Processing Systems*, 2019.

[2] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixé. Tracking without bells and whistles. *arXiv preprint arXiv:1903.05625*, 2019.

[3] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *European Conference on Computer Vision*, 2016.

[4] Alex Bewley, ZongYuan Ge, Lionel Ott, Fabio Tozeto Ramos, and Ben Upcroft. Simple online and realtime tracking. In *International Conference on Image Processing*, 2016.

[5] Erik Bochinski, Volker Eiselein, and Thomas Sikora. High-speed tracking-by-detection without using image information. In *IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2017.

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

[7] Wongun Choi and Silvio Savarese. Multiple target tracking in world coordinate with single, minimally calibrated camera. In *European Conference on Computer Vision*, 2010.

[8] Achal Dave, Tarasha Khurana, Pavel Tokmakov, Cordelia Schmid, and Deva Ramanan. Tao: A large-scale benchmark for tracking any object. In *European Conference on Computer Vision*, 2020.

[9] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 2010.

[10] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Detect to track and track to detect. In *IEEE International Conference on Computer Vision*, 2017.

[11] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[12] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[13] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[14] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *IEEE International Conference on Computer Vision*, 2017.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[16] David Held, Sebastian Thrun, and Silvio Savarese. Learning to track at 100 FPS with deep regression networks. In *European Conference on Computer Vision*, 2016.

[17] Olivier J Hénaff, Aravind Srinivas, Jeffrey De Fauw, Ali Razavi, Carl Doersch, SM Eslami, and Aaron van den Oord. Data-efficient image recognition with contrastive predictive coding. *arXiv preprint arXiv:1905.09272*, 2019.

[18] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017.

[19] Andrea Hornakova, Roberto Henschel, Bodo Rosenhahn, and Paul Swoboda. Lifted disjoint paths with application in multiple object tracking. In *International Conference on Machine Learning*, 2020.

[20] Chanho Kim, Fuxin Li, Arridhana Ciptadi, and James M. Rehg. Multiple hypothesis tracking revisited. In *IEEE International Conference on Computer Vision*, 2015.

[21] Chanho Kim, Fuxin Li, and James M. Rehg. Multi-object tracking with neural gating using bilinear LSTM. In *European Conference on Computer Vision*, 2018.

[22] Laura Leal-Taixé, Cristian Canton-Ferrer, and Konrad Schindler. Learning by tracking: Siamese CNN for robust target association. In *IEEE Conference on Computer Vision and Pattern Recognition Workshop*, 2016.

[23] Laura Leal-Taixé, Anton Milan, Konrad Schindler, Daniel Cremers, Ian Reid, and Stefan Roth. Tracking the trackers: an analysis of the state of the art in multiple object tracking. *arXiv preprint arXiv:1704.02781*, 2017.

[24] Tsung-Yi Lin, Piotr Dollár, Ross B Girshick, Kaiming He, Bharath Hariharan, and Serge J Belongie. Feature pyramid networks for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[25] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, 2014.

[26] Zhichao Lu, Vivek Rathod, Ronny Votel, and Jonathan Huang. Retinatrack: Online single stage joint detection and tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[27] Nima Mahmoudi, Seyed Mohammad Ahadi, and Mohammad Rahmati. Multi-target tracking using cnn-based features: Cnnmtt. *Multimedia Tools and Applications*, 2019.

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

[29] Anton Milan, Seyed Hamid Rezatofighi, Anthony R. Dick, Ian D. Reid, and Konrad Schindler. Online multi-target tracking using recurrent neural networks. In *The AAAI Conference on Artificial Intelligence*, 2017.

[30] Anton Milan, Stefan Roth, and Konrad Schindler. Continuous energy minimization for multitarget tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013.

[31] James Munkres. Algorithms for the assignment and transportation problems. *Society for Industrial and Applied Mathematics*, 1957.

[32] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[33] Bo Pang, Yizhuo Li, Yifan Zhang, Muchen Li, and Cewu Lu. Tubetk: Adopting tubes to track multi-object in a one-step training model. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[34] Jiangmiao Pang, Kai Chen, Jianping Shi, Huajun Feng, Wanli Ouyang, and Dahua Lin. Libra r-cnn: Towards balanced learning for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[35] Jinlong Peng, Changan Wang, Fangbin Wan, Yang Wu, Yabiao Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Yanwei Fu. Chained-tracker: Chaining paired attentive regression results for end-to-end joint multiple-object detection and tracking. In *European Conference on Computer Vision*, 2020.

[36] Deva Ramanan and David A Forsyth. Finding and tracking people from the bottom up. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2003.

[37] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, 2015.

[38] Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Tracking the untrackable: Learning to track multiple cues with long-term dependencies. In *IEEE International Conference on Computer Vision*, 2017.

[39] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in Neural Information Processing Systems*, 2016.

[40] Jeany Son, Mooyeol Baek, Minsu Cho, and Bohyung Han. Multi-object tracking with quadruplet convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[41] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset, 2019.

[42] Yifan Sun, Changmao Cheng, Yuhan Zhang, Chi Zhang, Liang Zheng, Zhongdao Wang, and Yichen Wei. Circle loss: A unified perspective of pair similarity optimization. *arXiv preprint arXiv:2002.10857*, 2020.

[43] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019.

[44] Zhongdao Wang, Liang Zheng, Yixuan Liu, and Shengjin Wang. Towards real-time multi-object tracking. *arXiv preprint arXiv:1909.12605*, 2019.

[45] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *International Conference on Image Processing*, 2017.

[46] Yuxin Wu and Kaiming He. Group normalization. In *European Conference on Computer Vision*, 2018.

[47] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[48] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *European Conference on Computer Vision*, 2018.

[49] Bo Yang and Ram Nevatia. An online learned CRF model for multi-target tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[50] Linjie Yang, Yuchen Fan, and Ning Xu. Video instance segmentation. In *IEEE International Conference on Computer Vision*, 2019.

[51] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[52] Fengwei Yu, Wenbo Li, Quanquan Li, Yu Liu, Xiaohua Shi, and Junjie Yan. POI: multiple object tracking with high performance detection and appearance feature. In *European Conference on Computer Vision Workshop*, 2016.

[53] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. Fairmot: On the fairness of detection and re-identification in multiple object tracking. *arXiv preprint arXiv:2004.01888*, 2020.

[54] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. In *European Conference on Computer Vision*, 2020.

[55] Zongwei Zhou, Junliang Xing, Mengdan Zhang, and Weiming Hu. Online multi-target tracking with tensor-based high-order graph matching. In *International Conference on Pattern Recognition*, 2018.