Generalizing Multiple Object Tracking to Unseen Domains by Introducing Natural Language Representation

En Yu[1*], Songtao Liu[3*], Zhuoling Li [2], Jinrong Yang [1], Zeming Li[3], Shoudong Han[1†], Wenbing Tao[1†]

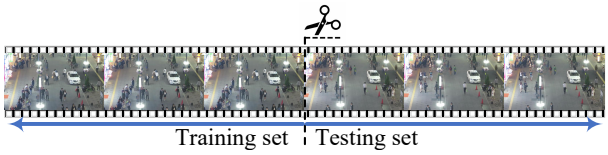
[1]Huazhong University of Science and Technology,
[2]Tsinghua University,
[3] Megvii(Face++) Inc
{yuen, yangjinrong, shoudonghan, wenbingtao}@hust.edu.cn

Abstract

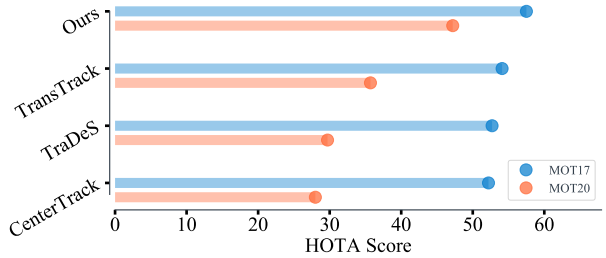
Although existing multi-object tracking (MOT) algorithms have obtained competitive performance on various benchmarks, almost all of them train and validate models on the same domain. The domain generalization problem of MOT is hardly studied. To bridge this gap, we first draw the observation that the high-level information contained in natural language is domain invariant to different tracking domains. Based on this observation, we propose to introduce natural language representation into visual MOT models for boosting the domain generalization ability. However, it is infeasible to label every tracking target with a textual description. To tackle this problem, we design two modules, namely visual context prompting (VCP) and visual-language mixing (VLM). Specifically, VCP generates visual prompts based on the input frames. VLM joints the information in the generated visual prompts and the textual prompts from a pre-defined Trackbook to obtain instance-level pseudo textual description, which is domain invariant to different tracking scenes. Through training models on MOT17 and validating them on MOT20, we observe that the pseudo textual descriptions generated by our proposed modules improve the generalization performance of query-based trackers by large margins.

Introduction

As a fundamental vision perception task, multi-object tracking (MOT) has been extensively deployed in broad applications, e.g., autonomous driving, video analysis and intelligent robots (Bewley et al. 2016; Wojke, Bewley, and Paulus 2017). There exist numerous works studying how to track targets well (Bergmann, Meinhardt, and Leal-Taixe 2019; Peng et al. 2020; Pang et al. 2020; Wang et al. 2020). The early methods usually first detect targets using either anchor-based or keypoint-based detectors, and then associate the detected targets based on extracted appearance representation (Wang, Weng, and Kitani 2020; Zhang et al. 2021) or predicted motion (Bergmann, Meinhardt, and Leal-Taixe 2019; Peng et al. 2020). Recently, some researchers apply vision transformer (Dosovitskiy et al. 2020; Carion et al. 2020) to MOT for implementing the models in a more end-to-end fashion (Meinhardt et al. 2021; Zeng et al. 2021).



(a) How the MOT17 dataset is split.



(b) Cross-domain comparision

Figure 1: (a) Since the training set and testing set of existing MOT datasets (such as MOT16 and MOT17) are usually created by splitting the same videos, they are completely from the same domains. (b) Compared with other counterparts, our method presents outstanding generalization ability when evaluated on an unseen domain.

Although previous MOT methods have achieved promising performance to some extent, almost all of them only consider training and evaluating models in the same domain. In fact, as shown in Fig. 1 (a), the training and testing sets of commonly employed MOT datasets, such as MOT17 and MOT20, are mostly produced by spliting the identical videos into two parts. Thus the existing models are only evaluated in the same domain as the training set. When we test some of them (i.e., TransTrack, TraDeS, and CenterTrack) in another unseen domain, as illustrated in Fig. 1 (b), their performance drops dramastically, leading to generalization bottleneck. And to our best knowledge, this domain generalization problem of MOT is hardly studied in literature.

In this work, we aim to tackle the poor generalization ability of exist trackers where the cross-domain training data is unattainable. Inspired by the recent works of multi-modal

[*]These authors contributed equally.
[†]Corresponding authors

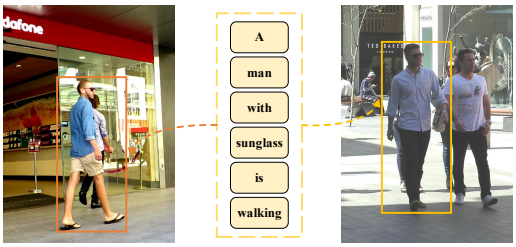


Figure 2: The generalization ability of natural language representation. As shown, since the information in a sentence is high-level, it can describe persons with similar appearances in two significantly different domains.

models, we try to seek more generalized features besides vision. To this end, we first draw the observation that natural language description is naturally domain invariant to different tracking domains. For example, as illustrated in Fig. 2, although the two men appear in two significantly different domains (e.g., illumination, background, or resolution), they can be described by the same sentence. Hence, we believe that natural language representation generalizes better. Moreover, the recent models like CLIP (Radford et al. 2021) provide unified visual and linguistic representations, allowing us to design some advanced strategies to introduce the natural language representation into MOT models.

To achieve this goal, the key challenge is how to design a model for adaptively pre-processing and effectively post-aggregating both natural language and images. CLIP-like models only bridge the features from two modals, while our model should learn to improve its generalization ability in the tracking task. By virtue of the versatile multi-modal fusion property (e.g., image, text, or point cloud) of the Transformer, we select a Transformer-based tracker, MOTR (Zeng et al. 2021), as the baseline model. In MOTR, every tracklet is represented as a track query. Hence, it is straightforward to devise modules for incorporating the text representation into the original track queries of MOTR.

However, there is still a severe obstacle hindering our design: it is too troublesome to label every tracking target in the training set with description text. To address this problem, we first prepare a Trackbook, which comprises 56 description phrases covering most tracking cases. In this way, the following problem is how to associate the information in these description phrases with detected targets.

To this end, we first convert the phrases in Trackbook to word embeddings named textual prompts through bag-of-words (Zhang, Jin, and Zhou 2010). Afterwards, we design two modules i.e., Visual Context Prompting (VCP) and Vision-Language Mixture (VLM), to associate the description text in the Trackbook with tracked targets automatically. VCP encodes the input image contexts as tokens called visual prompts, which contain domain related appearance and scene information of tracking targets. This information can be regarded as the implicit description of targets and used as the supplement to the domain-independent textual prompts. VLM associates and mixes the aforementioned textual prompts and visual prompts for generating new embeddings called pseudo textual description (PTD). The PTD contains high-level descriptions about tracking targets and generalizes well like the sentences presented in Fig. 2. Therefore, by combining the pseudo textual descriptions with the original track queries, the generalization ability of the baseline tracker is improved.

Through combining the proposed method of generating PTD with the baseline method, we obtain a new tracking model and name it as *language-guided tracker* (LTrack). To evaluate its generalization performance and compare with other advanced methods, we train all models on MOT17 and validate on MOT20 dataset. To our knowledge, this is the first work to leverage this testing protocal, which is exactly a new cross-domain MOT evaluation benchmark. The experimental results reveal that our method boosts the generalization ability of the baseline model significantly while achieving state-of-the-art (SOTA) performance on this benchmark.

Related Work

Multiple-object Tracking. Thanks to the fast development of object detection techniques (Ren et al. 2015; Tian et al. 2019; Zhou, Wang, and Krähenbühl 2019; Carion et al. 2020), existing trackers mainly follow the tracking-by-detection paradigm (Bewley et al. 2016; Wojke, Bewley, and Paulus 2017; Bergmann, Meinhardt, and Leal-Taixe 2019; Peng et al. 2020; Pang et al. 2020; Wang et al. 2020), which first localizes targets in each frame and then associates them based on their recognized identities to obtain trajectories.

According to the association strategy, early MOT methods can be further divided into motion-based trackers and appearance-based trackers. Most motion-based trackers perform the association step based on motion prediction algorithms, such as Kalman Filter (Bishop, Welch et al. 2001) and optical flow (Baker and Matthews 2004). There are also some other motion-based trackers (Feichtenhofer, Pinz, and Zisserman 2017; Bergmann, Meinhardt, and Leal-Taixe 2019; Zhou, Koltun, and Krähenbühl 2020; Sun et al. 2020; Shuai et al. 2021) that build networks to directly predict the future locations or displacements of concerned targets. In contrast to the motion-based trackers, the appearance-based trackers (Wojke, Bewley, and Paulus 2017; He et al. 2021; Wang et al. 2020; Zhang et al. 2021; Liang et al. 2020; Wu et al. 2021; Yu et al. 2022) first extract the appearance representation of targets and match targets to trajectories based on the similarity of the obtained representation.

Although the performance of the aforementioned methods is competitive, they do not realize fully end-to-end tracking due to the post-processing operations. Recently, The Transformer model originally designed for natural language processing (NLP) has been applied to computer vision. For example, DETR (Carion et al. 2020) models the 2D object detection task as a set-to-set prediction problem based on Transformer (Vaswani et al. 2017). DETR (Carion et al. 2020) is first proposed to turn the object detection into a set prediction problem. Inspired by DETR, TrackFormer (Meinhardt et al. 2021) and MOTR (Zeng et al. 2021) regard MOT as a sequence prediction problem by representing every trajectory as a track query. They are fully end-to-end and do

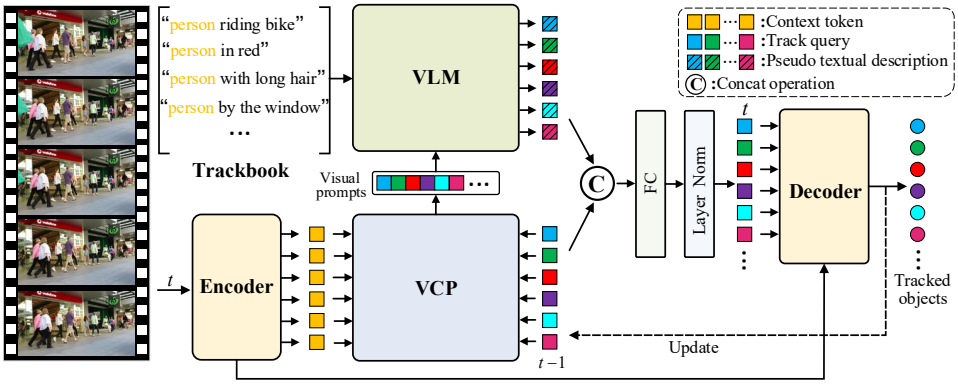


Figure 3: Overall pipeline of LTrack.

not demand post-processing. In this work, MOTR is taken as the baseline method.

Vision-language Models. Vision-language models have been widely studied in the fields of text-to-image retrieval (Wang et al. 2019), images caption (Xu et al. 2015), visual question answering (Antol et al. 2015), referring segmentation (Hu, Rohrbach, and Darrell 2016), etc. Among the many related publications, vision-language pretraining has attracted growing attention recently, and the milestone work is Contrastive Language-Image Pre-training (CLIP) (Radford et al. 2021). CLIP pre-trains models through conducting contrastive learning among 400 million image-text pairs crawled from the Internet. Its impressive generalization ability has been confirmed by being evaluated across 30 classification datasets. The pre-trained CLIP encoders are also applied to many other downstream tasks, e.g., open-vocabulary detection (Gu et al. 2021), zero-shot semantic segmentation (Rao et al. 2022) and so on. However, to the best as we know, CLIP encoders have not been applied to MOT, and we are the first to utilize the knowledge contained in the CLIP encoders to boost the generalization performance of visual MOT models.

Domain Generalization for MOT. Domain generalization is critical for practical applications of visual deep learning models. When evaluated on an unseen testing domain, a model is expected to present performance consistent with that of the training domain. Although the domain generalization problem has been extensively studied in many tasks, such as object detection (Chen et al. 2018) and person re-identification (Deng et al. 2018), it is hardly studied in MOT. In this work, we aim to bridge this gap.

Methodology

Delve into the Track Query

MOTR is an end-to-end MOT tracker that is build on the transformer. The key design of MOTR is the track query. In fact, MOTR employs two kinds of learnable embeddings, i.e., the detect queries and track queries. The detect queries are used in the same way as DETR (Carion et al. 2020), and the difference is that they are only responsible for recognizing newborn targets in each frame rather than all targets. When a new object is detected, the matched detect query obtains a corresponding track query to track this target. Meanwhile, if a previously detected object is continuously missing for N_m frames (N_m is a hyper-parameter), the track query is deleted. Hence, the track query naturally represents a trajectory, and no post-processing such as NMS is demanded.

The track queries are continuously updated according to the representation extracted from the input frame during the tracking process, Nevertheless, the representation is only learned from low-level image features and represents the statistics of the training set. Hence, they are prone to being over-fitting to a specific tracking domain. When the tracking domain changes, the tracker could fail. To tackle this problem, we hope to introduce high-level natural language representation into track queries to boost the generalization capability.

Method Overview

As illustrated in Fig. 3, LTrack comprises four components, the original MOTR (including the shown encoders and decoders). the Trackbook, and the proposed VCP and VLM modules. Given the t frame of a video as input, the encoders extract context tokens from it. The context tokens and the track queries of the $t-1$ frame are fed to VCP to produce visual prompts. Afterwards, the visual prompts and the textual prompts generated from the Trackbook are transformed as PTD tensors by VLM. The CLIP text encoder is employed in VLM to transfer the prompts to the PTD. Finally, the track queries of the $t-1$ frame and the PTD tensors are concatenated and encoded as the track queries of the t frame for further tracking.

Trackbook

As mentioned before, it is too troublesome to label every tracking target with a descriptive sentence. Alternatively, we create a dictionary called Trackbook including 56 description phrases, such as "person riding bike". These phrases can describe most tracking cases. Then, we transform these phrases into more intact sentences by adding a fixed template. For example, "person riding bike" is converted as "A photo of person riding bike". Afterwards, we convert all

these sentences as tensors named textual prompts through the classical bag-of-words method (Zhang, Jin, and Zhou 2010). The textual prompts contain the description information of various kinds of tracking targets. We hope our designed modules can learn to associate a target with the corresponding description information in the textual prompts automatically and use the information to improve the generalization performance. Notably, the Trackbook can be dynamically supplemented. And the learned PTD can be more general when applying more detailed textual descriptions in the Trackbook.

Visual Context Prompting

We aim to associate the textual information in Trackbook with the visual information of tracking targets. In addition, we hope these two kinds of information can be processed by a unified module. Thus, we hope the visual and textual information can be organized in the same format (such as tensors of the same dimension). Since textual information is often represented as prompts, the visual information should also be modeled as prompts. In this way, we design a module to learn the association between them later. Moreover, the textual information from the Trackbook is coarse-grained and the number of texts is limited. Therefore, we need to capture the fine-grained information from the image contexts to enrich the textual description.

To the aforementioned end, we design the VCP module. The VCP module is for generating visual prompts based on the context tokens generated by the encoders shown in Fig. 3. To clearly inform VCP which regions it should focus on, the track queries of the previous frame are also adopted as the input of VCP. Specifically, VCP adopts the Transformer decoder structure. It takes the context tokens produced by the final encoder (denoted as c_t) and the track queries of the previous frame q_{t-1} as input, which is formulated as follows:

$$v_t = TransDecoder(q_{t-1}, c_t), \tag{1}$$

where $v_t \in R^{M \times D}$ denotes the obtained visual prompts. M and D are the track query number and the embedding dimension, respectively. In VCP, the Query of the Transformer decoder is obtained based on the original track query q_{t-1}, and the Key and Value are derived from c_t.

Visual-Language Mixing

VLM is responsible for mixing the textual prompts from Trackbook and the visual prompts from VCP to obtain PTD. The detailed structure of VLM is illustrated in Fig. 4. As shown, VLM consists of three parts, i.e., the text encoder, the adapter, and a cross-attention module. The text encoder is exactly the publically available CLIP text encoder, which is pre-trained using about 400 million image-text pairs. In this work, the weights of this text encoder are fixed and not updated during the training process in order to protect the original CLIP knowledge.

Since the weights of the text encoder are fixed, we design an adapter network after this text encoder to blend new knowledge with the original CLIP. The adapter is a MLP

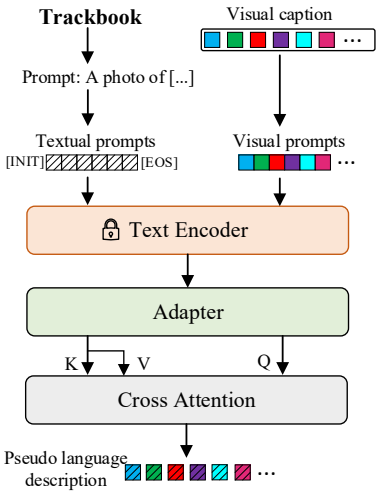


Figure 4: Illustration of the visual-language mixing (VLM) module, which consists of three components: (1) A pre-trained CLIP text encoder ($[INIT]$ and $[EOS]$ are the tokens to mark the begin and end of the text sequence). (2) An adapter module. (3) A cross-attention module.

block with a residual connection. Mathematically, it can be formulated as:

$$T_a(x) = Relu(T(x)^T W_1)W_2 + x, \tag{2}$$

where W_1 and W_2 are linear transformation weights, and $T(\cdot)$ denotes the CLIP text encoder. Notably, in the text encoder and adapter modules, the textual and visual prompts are processed independently.

After the adapter module, a cross-attention module is built to associate the textual representation and visual representation. It also adopts the Transformer decoder structure, which can be formulated as:

$$\begin{aligned} v &= [v_1, v_2, ..., v_M], \\ t &= [t_1, t_2, ..., t_K], \\ l &= CrossAttn(T_a(v), T_a(t)), \end{aligned} \tag{3}$$

where M represents the visual prompt length and K denotes the textual prompt length. In this work, we only employ the CLIP text encoder and do not use the CLIP image encoder. It is because we observe that the representation produced by the CLIP image encoder is unsuitable for tasks including localization.

The output of the cross-attention module in VLM is the learned PTD. By concatenating it with the track queries of the previous frame and further transforming the concatenated tensors (the FC and layer normalization in Fig. 3), we obtain the track queries of the t frame. These enhanced track queries are used to update the tracklets using the algorithm proposed by MOTR, leading to promising performance.

Optimization

Given a clip V_ξ of N frames as input, the results predicted by the model are denoted as $\widehat{P} = \{\hat{p}_i\}_{i=1}^N$, and the corre-

sponding ground-truths are $P = \{p_i\}_{i=1}^{N}$. The overall loss L_{clip} is computed based on $\widehat{P}$ and P. It consists of two parts, the tracking loss and detection loss. These two losses exactly share the same form. The difference is that the tracking loss is for localizing the targets that have been recognized in previous frames, and the detection loss is to tackle the newborn targets. Mathematically, L_{clip} can be formulated as follows:

$$\mathcal{L}_{clip} = \frac{\sum_{n=1}^{N}(\mathcal{L}(\widehat{P}_{tr}^{i}|_{q_t}, P_{tr}^{i}) + \mathcal{L}(\widehat{P}_{det}^{i}|_{q_d}, P_{det}^{i}))}{\sum_{n=1}^{N}(T_i)}, \quad (4)$$

where $\widehat{P}_{tr}^{i}|_{q_t}$, $\widehat{P}_{tr}^{i}$, $\widehat{P}_{det}^{i}|_{q_d}$, and $\widehat{P}_{det}^{i}|_{q_d}$ are the association predictions, association labels, detection predictions, and prediction labels, respectively. T_i denotes the total number of the targets in the $i-\mathrm{th}$ frame. $\mathcal{L}(\cdot)$ is implemented similarily to the one in DETR, which is formulated as:

$$\mathcal{L}(\widehat{P}_i|_{q_i}, P_i) = \lambda_{cls}\mathcal{L}_{cls} + \lambda_{l_1}\mathcal{L}_{l_1} + \lambda_{giou}\mathcal{L}_{giou}, \quad (5)$$

where $\mathcal{L}_{cls}$, $\mathcal{L}_{l_1}$, and $\mathcal{L}_{giou}$ are the focal loss (Lin et al. 2017) for classification, L_1 loss for regressing width and height, and the common generalized IoU loss. λ_{cls}, λ_{l_1}, λ_{giou} are three hyper-parameters.

Experiments

Benchmark

As mentioned before, almost all previous works train and validate their trackers in the same domain, and the available datasets are usually collected in a single domain. However, in this work, we want to evaluate the domain generalization ability of trackers in unseen domains. Hence, we need to design a new cross-domain benchmark. To this end, we first revisit existing MOT datasets carefully, and select MOT17 (Milan et al. 2016) and MOT20 (Dendorfer et al. 2020) to build this benchmark. In the following, we first introduce MOT17 and MOT20 separately. Afterwards, we explain how we use them to compose the cross-domain benchmark. Finally, the used metrics are described.

MOT17 and MOT20. The MOT17 dataset consists of 14 video sequences. Among them, 7 sequences are for training and the other 7 sequences are used to validate models. These sequences cover various scenarios and weathers. Compared with MOT17, MOT20 is more challenging. It is composed of 8 long video sequences captured in 3 crowded scenes. In some scenes, more than 220 pedestrians are contained in a single frame. Besides, the scenes in the MOT20 dataset are more diversified including indoor and outdoor, day and night, etc.

Cross-domain Benchmark. To verify the domain generalization ability of models in unseen domains, we use MOT17 to train models and validate them in MOT20. Since there exists a significant domain gap between MOT17 and MOT20, this evaluation setting can serve as a new cross-domain benchmark. The adopted metrics in this benchmark include the HOTA (Luiten et al. 2021) and CLEAR-MOT Metrics (Bernardin and Stiefelhagen 2008). Specifically, HOTA consists of higher order tracking accuracy (HOTA), association accuracy score (AssA) and detection accuracy score (DetA). CLEAR-MOT Metrics include ID F1 score (IDF1), multiple object tracking accuracy (MOTA) and identity switches (IDS). Among them, HOTA, AssA, IDF1 are the most important metrics for comparing tracking performance. And all the metrics are evaluated by TrackEval [1].

Implementation Details

Following MOTR, we build LTrack based on Deformable-DETR (Zhu et al. 2020), which is pre-trained on COCO (Lin et al. 2014) and employs ResNet50 (He et al. 2016) as the image encoder. During the training process, the batch size is 1 and each batch contains a multi-frame video clip. The length of each video clip is 2 at the beginning of training, and it is increased by 1 after every 50 epochs. The frames in each clip are selected from training videos with a random interval between 1 to 10. We employ Adam as the optimizer (Kingma and Ba 2014) and the initial learning rate is set to 2×10^{-4}. LTrack is trained for totally 200 epochs and the learning rate decays by 10 at the $100\mathrm{th}$ epoch. The data augmentation strategy of LTrack follows the setting in MOTR, which includes random flip and random crop. During the training process, λ_{cls}, λ_{l_1}, and λ_{giou} are set as 2, 5, and 2, respectively.

Comparison with Previous SOTA Methods

In this part, we compare the performance of LTrack with preceding SOTA methods. The comparison is conducted under two protocols, in-domain evaluation and cross-domain evaluation. For in-domain evaluation, all models are trained using the MOT17 training set and the CrowdHuman dataset (a 2D pedestrian dataset) (Shao et al. 2018), and the trained models are verified with the MOT17 validation set. Notably, incorporating CrowdHuman into the training process is a common operation in previous publications. Thus, we also adopt it. For the in-domain evaluation part of Tab. 1, the methods marked in gray are trained in the same setting as LTrack. In the cross-domain evaluation protocol, all the methods are trained using the same data setting, and they are tested in the MOT20 validation set. The evaluation results are reported in Tab. 1.

In-domain Evaluation. As illustrated in Tab. 1, LTrack obtains the metrics HOTA of 57.5%, AssA of 56.1%, and IDF1 of 69.1% on MOT17. This means LTrack outperforms all compared methods in the in-domain evaluation setting. The results indicate that LTrack can tackle in-domain tracking scenes well. Notably, LTrack also obtains promising performance on the detection related metrics (59.4% DetA and 72.1% MOTA). Meanwhile, it only produces 2100 IDS, which is the lowest among all the methods. Therefore, the tracklets generated by LTrack are more continuous.

Cross-domain Evaluation. As presented in Tab. 1, when we test the compared methods in our built cross-domain evaluation benchmark, their performance drops significantly. By contrast, the performance drop of LTrack is relatively small. For instance, in MOT20, LTrack achieves 46.8% HOTA,

[1] TrackEval: https://github.com/JonathonLuiten/TrackEval

Method	Published	Data	HOTA↑	AssA↑	DetA↑	MOTA↑	IDF1↑	IDS↓
MOT17 (In-domain)								
TubeTK* (Pang et al. 2020)	CVPR20	17	48.0	45.1	51.4	63.0	58.6	4317
CTracker* (Peng et al. 2020)	ECCV20	17	49.0	45.2	53.6	66.6	57.4	5529
QDTrack* (Pang et al. 2021)	CVPR21	17	53.9	52.7	55.6	68.7	66.3	3378
FairMOT* (Zhang et al. 2021)	IJCV21	CH+EX+17	59.3	58.0	60.9	73.7	72.3	3303
CSTrack* (Liang et al. 2020)	Arxiv20	EX+17	-	-	-	70.6	71.6	3465
CenterTrack (Zhou, Koltun, and Krähenbühl 2020)	ECCV20	CH+17	52.2	51.0	53.8	67.8	64.7	3039
TraDeS (Wu et al. 2021)	CVPR21	CH+17	52.7	50.8	55.2	69.1	63.9	3555
TransTrack (Sun et al. 2020)	Arxiv20	CH+17	54.1	47.9	**61.6**	**74.5**	63.9	3663
TransCenter (Xu et al. 2021)	Arxiv21	CH+17	54.5	49.7	-	73.2	62.2	3663
MOTR (Zeng et al. 2021)	ECCV22	CH+17	57.2	55.8	58.9	71.9	68.4	2115
LTrack	Ours	CH+17	**57.5**	**56.1**	59.4	72.1	**69.1**	**2100**
MOT20 (Cross-domain)								
CenterTrack (Zhou, Koltun, and Krähenbühl 2020)	ECCV20	CH+17	29.7	25.6	35.0	42.9	39.0	6397
FairMOT (Zhang et al. 2021)	IJCV21	CH+17	41.9	35.9	**49.7**	57.6	53.8	13621
TraDeS (Wu et al. 2021)	CVPR21	CH+17	28.0	25.5	32.7	44.9	39.3	7729
CSTrack (Liang et al. 2020)	Arxiv20	CH+17	33.9	29.8	38.8	49.6	44.9	10041
TransTrack (Sun et al. 2020)	Arxiv20	CH+17	35.8	27.3	47.3	**58.1**	44.8	13189
OMC (Liang et al. 2022)	AAAI22	CH+17	38.8	32.2	46.9	55.9	49.4	13813
MTrack (Yu, Li, and Han 2022)	CVPR22	CH+17	40.6	37.0	44.9	54.8	52.9	7639
MOTR (Zeng et al. 2021)	ECCV22	CH+17	43.7	41.7	45.9	56.0	56.8	2184
LTrack	Ours	CH+17	**46.8**	**45.4**	48.4	57.8	**61.1**	**1841**

Table 1: Performance comparison with preceding SOTAs on the in-domain and cross-domain evaluation benchmarks. ↑/↓ indicates that higher/lower score is better. Previous MOT trackers are often trained using different data volumes. For the in-domain evaluation part (MOT17) of this table, * means different training data setting is used.

45.4% AssA and $61,1\%$ IDF1, which significantly outperforms all compared methods by large margins. The results indicate that the generalization ability of LTrack to unseen domains is promising. In addition, LTrack surpasses its baseline method, MOTR, by 3.1% (46.8% vs. 43.7%) on HOTA, 3.7% (45.4% vs. 41.7%) on AssA and 4.3% (61.1% vs. 56.8%) on IDF1, which further confirms the benefit of introducing text representation into visual MOT.

Ablation Study

In this subsection, we verify the effectiveness of the proposed modules separately through ablation studies. All the experiments are conducted on the cross-domain evaluation benchmark. To accelerate the ablation study process, We use a lite version of LTrack by reducing the number of Transformer encoders from 6 to 1. The models are trained on the MOT17 training set for 200 epochs.

Analysis on VCP. In this part, we conduct an in-depth analysis of VCP. VCP adopts the transformer decoder structure to produce visual prompts. Among its inputs, the information in visual context tokens is crucial. However, we are unclear about which visual features from previous modules (the backbone and Transformer encoders) should be taken as the context tokens. To address this problem, We select the last three stages' feature maps of the backbone ResNet-50 and the output of the transformer encoder as the input to VCP separately. The results are reported in Tab. 2.

As shown in Tab. 2, the output from the Transformer encoder leads to the best performance, and using the c4 feature behaves better than employing the c3 and c5 features. This phenomenon reveals that both the resolution and semantic information of the visual features are important for generating visual prompts. Due to the deformable attention (Zhu et al. 2020), the final output of the encoder contains multi-scale semantic context knowledge. Therefore, we choose the output of the transformer encoder as the input to VCP.

Key feature	HOTA↑	AssA↑	MOTA↑	IDF1↑	IDS↓
c3	15.8	19.9	12.0	19.0	5157
c4	18.3	21.1	14.5	22.1	4849
c5	15.5	16.6	15.0	19.1	6712
Enc	**19.2**	**21.9**	**16.7**	**24.1**	**4674**

Table 2: Comparison between different visual features to VCP. c3, c4, and c5 are the feature maps from last three stages of the ResNet-50 network. Enc denotes the output of the final Transformer encoder.

Prompt	HOTA↑	AssA↑	MOTA↑	IDF1↑	IDS↓
Textual	16.1	17.9	10.4	19.2	6434
Visual	17.5	20.4	13.7	21.4	5138
Both	**19.2**	**21.9**	**16.7**	**24.1**	**4674**

Table 3: Comparisons between using different prompts.

Analysis of the visual and textual prompts. As mentioned

Adapter	HOTA↑	AssA↑	MOTA↑	IDF1↑	IDS↓
w	**19.2**	**21.9**	**16.7**	**24.1**	**4674**
w/o	16.7	16.9	14.8	19.9	6835

Table 4: Comparisions between VLM with (w) and without (w/o) the adapter.

before, textual and visual prompts are obtained by Trackbook and VCP, respectively, then they are encoded as PTD in VLM. In this part, we verify the performance of only using one of them, and the results are reported in Tab. 3. As shown, both the textual and visual prompts contribute to the final tracking performance. When we combine both the text information from textual prompts and the image information from visual prompts to produce PTD, the best result is obtained. This observation suggests that incorporating textual description information is helpful for improving MOT performance, and combining the knowledge of multiple modals is a promising research direction.

Analysis on the adapter. In this part, we analyze the importance of the adapter in the proposed VLM module. The tracking performance corresponding to the models with and without the adapter is reported in Tab. 4. We can observe that the model without the adapter behaves significantly poorer. Specifically, by removing the adapter, the metric HOTA is decreased by 2.5% (19.2% - 16.7%) and AssA is decreased by 5.0% (21.9% - 16.9%). This issue suggests that directly using the output of the CLIP text encoder to generate PTD through cross-attention is not proper. We think this is because of the significant difference between textual prompts and visual prompts, and the adapter can alleviate this difference. Therefore, the adapter module is necessary for VLM.

Visualization

To better demonstrate the superiority of LTrack, we visualize some tracking cases of our method and the baseline, MOTR. One case is illustrated in Fig. 5 and it is selected from the testing set of MOT17. As shown, a person is partly occluded and hard to track. The original track query in MOTR can not provide clear guidance to networks on how to track the target correctly, which results in the tracklet disconnection (FN) and identity switch (IDS). On the contrary, LTrack recognizes targets and generates trajectories successfully by using the track queries with textual description information. This result further confirms the value of PTD in LTrack, which guides the tracker to focus on targets effectively.

More Result and Discussion

Performance on BDD100K. We further evaluate LTrack on a autonomous driving dataset, BDD100K (Yu et al. 2020). As shown in Tab. 5, LTrack still achieves better performance than MOTR on BDD100K. However, the improvement is not that remarkable. We speculate this phenomenon is due to the lack of text descriptions about autonomous driving scenes. This issue merits further exploration in future work.

Limitation and future work. LTrack makes a first attempt for the cross-domain MOT task and achieves promising performance, but it still leaves some issues deserving further study. For example, LTrack does not behave well when many similar targets are shared by the same textual information appearing in the same frame. Thus, leveraging more fine-grained textual descriptions may be a future direction to resolve this long-standing problem in MOT. Another limitation is the related open-vocabulary MOT dataset. The paper explains that LTrack only uses 56 coarse-grained textual descriptions in Trackbook. If we have fine-grained textual captions for every target in the video, we can use textual knowledge to guide the tracker explicitly. Nevertheless, the labeling process is very expensive. Hence, to further explore the problem of how to use textual information to guide MOT models, an open-vocabulary MOT dataset is needed.

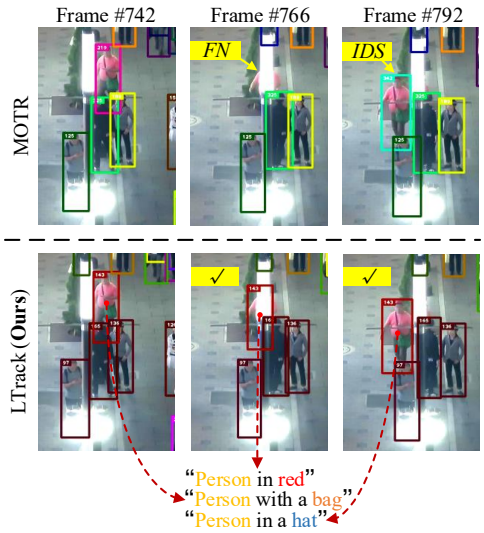


Figure 5: Qualitative comparison between the LTrack and the baseline method, MOTR. **FN**: False negative, **IDS**: Identity switch.

Method	Data	HOTA-p↑	AssA↑	IDF1-p↑
MOTR	CH+17	32.4	38.8	39.3
LTrack	CH+17	**33.7**	**39.3**	**40.6**

Table 5: Performance on BDD100K (pedestrian only).

Conclusion

In this work, we have pointed out that the domain generalization ability of MOT is hardly studied. To bridge this gap, we first highlighted that the knowledge contained in natural language is inherently more invariant to various domains. Based on this insight, we have implemented a new tracker, namely LTrack, which contains VCP and VLM module to combine the textual and visual prompts. We hope this work can shed light on how to develop MOT trackers with promising generalization ability to some extent.

Acknowledgements. This work was supported by the National Natural Science Foundation of China under Grants 62176096 and 61991412.

References

Antol, S.; Agrawal, A.; Lu, J.; Mitchell, M.; Batra, D.; Zitnick, C. L.; and Parikh, D. 2015. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, 2425–2433.

Baker, S.; and Matthews, I. 2004. Lucas-kanade 20 years on: A unifying framework. *Int J Comput Vis*, 56(3): 221–255.

Bergmann, P.; Meinhardt, T.; and Leal-Taixe, L. 2019. Tracking without bells and whistles. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 941–951.

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

Bewley, A.; Ge, Z.; Ott, L.; Ramos, F.; and Upcroft, B. 2016. Simple online and realtime tracking. In *2016 IEEE international conference on image processing*, 3464–3468. IEEE.

Bishop, G.; Welch, G.; et al. 2001. An introduction to the kalman filter. *Proc of SIGGRAPH, Course*, 8(27599-23175): 41.

Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; and Zagoruyko, S. 2020. End-to-end object detection with transformers. In *European Conference on Computer Vision*, 213–229.

Chen, Y.; Li, W.; Sakaridis, C.; Dai, D.; and Van Gool, L. 2018. Domain adaptive faster r-cnn for object detection in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3339–3348.

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

Deng, W.; Zheng, L.; Ye, Q.; Kang, G.; Yang, Y.; and Jiao, J. 2018. Image-image domain adaptation with preserved self-similarity and domain-dissimilarity for person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 994–1003.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Feichtenhofer, C.; Pinz, A.; and Zisserman, A. 2017. Detect to track and track to detect. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3038–3046.

Gu, X.; Lin, T.-Y.; Kuo, W.; and Cui, Y. 2021. Open-vocabulary Object Detection via Vision and Language Knowledge Distillation. *arXiv preprint arXiv:2104.13921*.

He, J.; Huang, Z.; Wang, N.; and Zhang, Z. 2021. Learnable Graph Matching: Incorporating Graph Partitioning with Deep Feature Learning for Multiple Object Tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5299–5309.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Hu, R.; Rohrbach, M.; and Darrell, T. 2016. Segmentation from natural language expressions. In *European Conference on Computer Vision*, 108–124.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Liang, C.; Zhang, Z.; Lu, Y.; Zhou, X.; Li, B.; Ye, X.; and Zou, J. 2020. Rethinking the competition between detection and ReID in Multi-Object Tracking. *arXiv preprint arXiv:2010.12138*.

Liang, C.; Zhang, Z.; Zhou, X.; Li, B.; and Hu, W. 2022. One More Check: Making "Fake Background" Be Tracked Again. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 1546–1554.

Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollár, P. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2980–2988.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*.

Luiten, J.; Osep, A.; Dendorfer, P.; Torr, P.; Geiger, A.; Leal-Taixé, L.; and Leibe, B. 2021. Hota: A higher order metric for evaluating multi-object tracking. *International journal of computer vision*, 129(2): 548–578.

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

Milan, A.; Leal-Taixé, L.; Reid, I.; Roth, S.; and Schindler, K. 2016. MOT16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*.

Pang, B.; Li, Y.; Zhang, Y.; Li, M.; and Lu, C. 2020. Tubetk: Adopting tubes to track multi-object in a one-step training model. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 6308–6318.

Pang, J.; Qiu, L.; Li, X.; Chen, H.; Li, Q.; Darrell, T.; and Yu, F. 2021. Quasi-dense similarity learning for multiple object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 164–173.

Peng, J.; Wang, C.; Wan, F.; Wu, Y.; Wang, Y.; Tai, Y.; Wang, C.; Li, J.; Huang, F.; and Fu, Y. 2020. Chained-tracker: Chaining paired attentive regression results for end-to-end joint multiple-object detection and tracking. In *European Conference on Computer Vision*, 145–161.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 8748–8763.

Rao, Y.; Zhao, W.; Chen, G.; Tang, Y.; Zhu, Z.; Huang, G.; Zhou, J.; and Lu, J. 2022. Denseclip: Language-guided

dense prediction with context-aware prompting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18082–18091.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, 28: 91–99.

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

Shuai, B.; Berneshawi, A.; Li, X.; Modolo, D.; and Tighe, J. 2021. SiamMOT: Siamese Multi-Object Tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12372–12382.

Sun, P.; Jiang, Y.; Zhang, R.; Xie, E.; Cao, J.; Hu, X.; Kong, T.; Yuan, Z.; Wang, C.; and Luo, P. 2020. TransTrack: Multiple-Object Tracking with Transformer. *arXiv preprint arXiv:2012.15460*.

Tian, Z.; Shen, C.; Chen, H.; and He, T. 2019. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9627–9636.

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

Wang, Y.; Weng, X.; and Kitani, K. 2020. Joint Detection and Multi-Object Tracking with Graph Neural Networks. *arXiv preprint arXiv:2006.13164*.

Wang, Z.; Liu, X.; Li, H.; Sheng, L.; Yan, J.; Wang, X.; and Shao, J. 2019. Camp: Cross-modal adaptive message passing for text-image retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 5764–5773.

Wang, Z.; Zheng, L.; Liu, Y.; Li, Y.; and Wang, S. 2020. Towards real-time multi-object tracking. In *European Conference on Computer Vision*, 107–122.

Wojke, N.; Bewley, A.; and Paulus, D. 2017. Simple online and realtime tracking with a deep association metric. In *2017 IEEE international conference on image processing (ICIP)*, 3645–3649.

Wu, J.; Cao, J.; Song, L.; Wang, Y.; Yang, M.; and Yuan, J. 2021. Track to Detect and Segment: An Online Multi-Object Tracker. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12352–12361.

Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhudinov, R.; Zemel, R.; and Bengio, Y. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, 2048–2057.

Xu, Y.; Ban, Y.; Delorme, G.; Gan, C.; Rus, D.; and Alameda-Pineda, X. 2021. TransCenter: Transformers with Dense Queries for Multiple-Object Tracking. *arXiv preprint arXiv:2103.15145*.

Yu, E.; Li, Z.; and Han, S. 2022. Towards Discriminative Representation: Multi-view Trajectory Contrastive Learning for Online Multi-object Tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8834–8843.

Yu, E.; Li, Z.; Han, S.; and Wang, H. 2022. Relationtrack: Relation-aware multiple object tracking with decoupled representation. *IEEE Transactions on Multimedia*.

Yu, F.; Chen, H.; Wang, X.; Xian, W.; Chen, Y.; Liu, F.; Madhavan, V.; and Darrell, T. 2020. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2636–2645.

Zeng, F.; Dong, B.; Wang, T.; Chen, C.; Zhang, X.; and Wei, Y. 2021. MOTR: End-to-End Multiple-Object Tracking with TRansformer. *arXiv preprint arXiv:2105.03247*.

Zhang, Y.; Jin, R.; and Zhou, Z.-H. 2010. Understanding bag-of-words model: a statistical framework. *International journal of machine learning and cybernetics*, 1(1): 43–52.

Zhang, Y.; Wang, C.; Wang, X.; Zeng, W.; and Liu, W. 2021. Fairmot: On the fairness of detection and re-identification in multiple object tracking. *Int J Comput Vis*, 1–19.

Zhou, X.; Koltun, V.; and Krähenbühl, P. 2020. Tracking objects as points. In *European Conference on Computer Vision*, 474–490.

Zhou, X.; Wang, D.; and Krähenbühl, P. 2019. Objects as points. *arXiv preprint arXiv:1904.07850*.

Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; and Dai, J. 2020. Deformable DETR: Deformable Transformers for End-to-End Object Detection. In *International Conference on Learning Representations*.