SLAck: Semantic, Location, and Appearance Aware Open-Vocabulary Tracking — Supplementary Material —

Siyuan Li[1], Lei Ke[1], Yung-Hsu Yang[1], Luigi Piccinelli[1], Mattia Segù[1], Martin Danelljan[1], and Luc Van Gool[1,2]

[1] Computer Vision Lab, ETH Zurich
[2] INSAIT

In this supplementary material, we provide additional ablation studies and results of SLAck. We also elaborate on our experimental setup, method details, and training and inference hyper-parameters. All ablations are validated in the novel split of open-vocabulary MOT benchmark [7].

1 Compare Object Motions on Different Datasets

In the main paper, we discuss the differences between pedestrian tracking and open-vocabulary tracking, which are also visually presented in Fig. 1. Instances in pedestrian tracking datasets like MOT20 [3] generally move in a linear path with minimal deformation. This simplicity allows many top-performing trackers to rely on the Kalman Filter. However, open-vocabulary tracking introduces significant challenges due to the complex, non-linear movements and substantial deformation of objects.

We also quantitatively showcase these differences in Fig. 2 by plotting kernel density estimation (KDE) based on two aspects: object displacement between consecutive frames in a video and the average Aspect Ratio Changes (ARC) of instances throughout the video. We analyze and visualize the KDE distribution by examining all instance trajectories in the MOT20 [3] and TAO [2] datasets.

To calculate object motion, we measure the displacement between two consecutive frames for each instance by computing the Euclidean distance between the centroids of their bounding boxes. For ARC, we determine the aspect ratio (width divided by height) of each bounding box per frame and then calculate the changes in aspect ratio between consecutive frames for each instance.

Fig. 2 further supports our qualitative observation from Fig. 1 that, compared to the traditional pedestrian tracking dataset MOT20, objects in open-vocabulary settings move with higher dynamism in both motion and shape. This is indicated by the broader KDE curve and the significantly larger average values for instance displacement and ARC changes.

2 Semantics and Motion Patterns

In the main paper, we explain the effectiveness of SLAck over previous state-of-the-art open-vocabulary trackers that rely solely on appearance information.

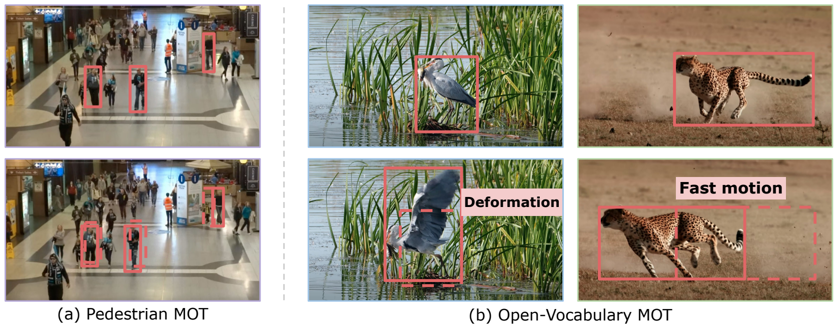

Fig. 1: Unlike conventionally pedestrian tracking such as MOT20 [3], open-world objects show high dynamicity on shape and motion, which poses significant challenges for motion-based trackers. We demonstrate this quantitatively on Fig. 2.

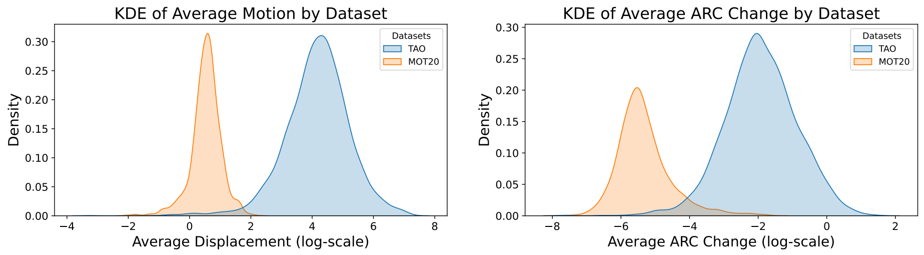

Fig. 2: Kernel density estimation (KDE) serves as a tool for comparing motion patterns between the open-vocabulary tracking dataset TAO and the conventional pedestrian tracking dataset MOT20. Given the substantial differences, we apply a logarithmic scale on the x-axis for both figures to facilitate a clearer comparison. **Left-KDE** illustrates the average motion disparities between objects in TAO and MOT20 datasets. For each instance, we compute the displacement between two consecutive frames. The open-vocabulary tracking dataset TAO showcases a broader variety of movements, as indicated by the width of the plot, and the average displacement is considerably larger than that observed in pedestrian tracking within MOT20. **Right KDE** focuses on the average Aspect Ratio Change (ARC) across the two datasets. This plot further confirms that, in open-vocabulary scenarios, objects undergo more pronounced deformation and occlusion, underscoring the complexities inherent in open-vocabulary tracking compared to traditional pedestrian tracking.

Our approach, which integrates semantic and location information, is based on the understanding that objects from different classes exhibit unique motion patterns. We now provide a quantitative demonstration using two descriptors for object motion defined earlier: average displacement and average aspect ratio changes. While this simplification doesn't capture all motion nuances, it offers a straightforward method for analyzing and visualizing motion patterns across different classes.

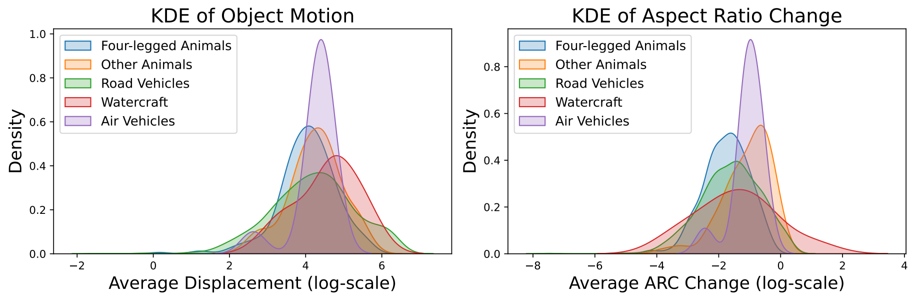

Fig. 3: Kernel density estimation (KDE) for comparing motion patterns for open-vocabulary tracking dataset TAO.

We calculate the average displacement and aspect ratio change (ARC) for each instance, then aggregate these metrics by class to uncover general motion characteristics and offer insights into the frequency and severity of deformation or occlusion for each class.

Semantic and Motion Relations for Parent Classes We categorize TAO classes into three main groups: Transport, Animal, and Static objects, focusing only on Transport and Animals—objects capable of autonomous movement. Static objects, which include items that must be carried or transported by others (e.g., personal items, sports equipment, household items, and work tools), are not analyzed due to their inherent lack of independent motion.

– **Transport** encompasses all means of transportation, organized by similarity in motion patterns. This includes road vehicles with similar motion patterns, air vehicles like airplanes and drones, and watercraft such as boats.
– **Animals** are grouped based on their locomotion methods or habitats, emphasizing similar motion patterns (e.g., four-legged land animals, birds, and fish).

Results for parent classes are shown in Fig. 3, with one figure illustrating the motion displacement density estimation and the other showing the average ARC for different categories. These figures should be considered together to fully understand the motion patterns and specific characteristics of each category. As illustrated, the distribution of motion patterns varies significantly across parent classes, confirming our assumption that objects from different classes move differently.

Semantic and Motion Relations for Child Classes Delving deeper, we examine the child categories within each parent group to analyze more specific motion patterns. Through KDE analysis, we present detailed findings for subclasses, such as road vehicles (Fig. 4), air vehicles (Fig. 5), four-legged animals (Fig. 6), and other animals (Fig. 7). Our observations reveal that even within semantically similar classes, distinct motion pattern variations exist. SLAck leverages these differences by learning the relationships between semantic categories and

Table 1: Sampling interval.

	Novel	Base
Time	AssocA	AssocA
1s	33.4	36.7
3s	**37.8**	**37.6**
5s	35.7	36.5
10s	35.7	35.5
20s	34.2	33.8

Table 2: Concat vs Add.

	Novel	Base
Model	AssocA	AssocA
Concat	35.8	**37.7**
Add	**37.8**	37.6

Table 3: img vs text.

	Novel	Base
Model	AssocA	AssocA
img	35.9	35.8
img + text	37.8	36.5
text	**37.8**	**37.6**

Table 4: Comparison with SOT method on TAO novel split.

	TETA	LocA	AssocA	ClsA	Inference Time
MixFormer	20.6	31.2	28.4	**2.4**	36h
SlAck (Ours)	**31.1**	**54.3**	**37.8**	1.3	**1.2h**

their associated motion patterns directly from the data, enhancing its tracking accuracy.

3 More Ablations

We provide more ablation studies in this section.

3.1 Temporal Sampling Intervals

We further perform ablation for temporal sampling interval, as shown in Table 1. Longer intervals risk objects disappearing, while shorter intervals fail to capture valuable cases such as occlusion.

3.2 Add vs. Concat for Feature Fusion

We compare addition against concatenation to fuse different cues in Table 2. Results show that addition yields better performance on novel classes compared to concatenation. Addition maintains fixed-size dimensionality, whereas concatenation increases dimensionality, leading to higher memory consumption and slower processing.

3.3 Distilled CLIP Features

We integrate the distilled CLIP text head features as semantic cues in the main paper. We here evaluated the integration of both distilled CLIP image and text head features in [5]. Results in Table 3 show that using only the distilled CLIP text head features as semantic cues yields the best performance.

Table 5: Inference with or without dynamic thresholding (DT).

Dynamic Thresholding	TETA	LocA	AssocA	ClsA
OVTrack *w/o* DT	27.8	48.3	33.6	1.5
Slack *w/o* DT	29.5	50.5	36.8	1.0
OVTrack *w/* DT	28.8	51.2	33.8	1.5
Slack *w/* DT	31.1	54.3	37.8	1.3

Table 6: With or without temporal encoding.

Method	TETA	LocA	AssocA	ClsA
SLAck *w/o* temporal encoding	30.6	**54.4**	36.8	0.7
SLAck *w/* temporal encoding	**31.1**	54.3	**37.8**	**1.3**

3.4 Dynamic Thresholding

We build SLAck with the same open-vocabulary detector used by OVTrack as a base. Our findings indicate that employing a fixed score threshold for open-vocabulary detection inference is suboptimal. This issue stems from the inherent design of FasterRCNN-based open-vocabulary detectors, which substitute the original classification head with a CLIP-distilled classification head. This setup calculates feature similarities among class text embeddings produced by the CLIP text encoder, followed by the softmax function to determine classification confidence for each class. Due to the introduction of novel classes during inference, the number of categories alters from the training phase, affecting the value distribution post-softmax. While OVTrack initially sets the score threshold during testing at 0.0001, our approach of utilizing a dynamic threshold, adjusted based on the number of target classes, yields improved detection outcomes. The test score threshold is determined by the following equation:

$$\text{score\_thr} = \left(\frac{1}{\text{number of classes}}\right) \times 1.001 \tag{1}$$

We report these findings in Table 5, demonstrating that this modification leads to a $+2.9$ LocA enhancement for OVTrack and a $+3.1$ LocA increase for SLAck. In Table 4 of the main paper, we referenced the OVTrack performance on the standard benchmark without incorporating our insights. Herein, we also apply dynamic thresholding (DT) to OVTrack and juxtapose it with SLAck under both with and without DT scenarios. Table 5 illustrates that SLAck surpasses OVTrack in both conditions, achieving at least a 3.2 improvement in association (AssocA).

3.5 Temporal Encoding

SLAck innovatively combines three crucial cues for the association: semantics, location, and appearances. Due to space constraints, the detailed explanation of temporal encoding was previously omitted. In this section, we elucidate temporal encoding and assess its efficacy. Temporal encoding aims to capture the differences between pairs of frames to be matched. For each frame, a unique encoding is generated for the entire frame. This is accomplished by downsampling the smallest feature maps in the Feature Pyramid Network (FPN) of the detector into a two-dimensional vector. Prior to matching objects across two frames, we compute the difference between the encodings of these frames, which constitutes the temporal encoding. This encoding reflects the changes between the two frames; if no movement occurs and the scene remains static, the difference approaches zero.

We replicate the temporal encoding for the same number of objects in the current frame and combine it with each object's fused embedding:

$$E^i_{\text{fused\_update}} = E^i_{\text{fuse}} + E^i_{\text{temporal}}, \tag{2}$$

where E^i_{fused} denotes the fused embedding for the i-th object in the frame, and E^i_{temporal} represents the temporal encoding. Table 6 demonstrates the temporal encoding's effectiveness, contributing to a $+1$ increase in AssocA for SLAck.

3.6 Compare with Single Object Tracking (SOT)

The significance of OV-MOT is its ability to support both open-vocabulary and **multiple objects**, whereas SOT focuses solely on tracking a single object. In OV-MOT, objects frequently appear and disappear, and new objects continually enter the scene, necessitating automatic handling of these scenarios, which SOT trackers cannot achieve. Additionally, OV-MOT often involves multiple objects with similar appearances appearing together, necessitating the use of motion or location cues. Moreover, OV-MOT tracks all objects at once without repeated runs as needed in SOT. Also, SOT fails to explicitly consider other objects in the scene, such as their semantics, relative location, and appearance that play a critical role in associating multiple objects. We test a strong SOT tracker MixFormer (GOT-10k) for the OV-MOT task on TAO in Table 4. Our approach significantly outperforms MixFormer. We use the same detector as ours to provide initial object boxes for MixFormer in the first frame of each video and run it repeatedly to track every object.

4 Model Details

Semantic Head The semantic head is a five-layer MLP with the GroupNorm and ReLU after each but the last layer.
Location Head The semantic head is a five-layer MLP with the GroupNorm and ReLU after each but the last layer.
Appearance Head The appearance head is a four-layer convolution with one additional MLP. After each conv layer follows a GroupNorm and ReLU.
Spatial and Temporal Object Graph (STOG) STOG is designed to process and refine fused features through a series of attentional propagation layers.

It employs a descriptor dimension of 256. The network's architecture consists of alternating layers of self-attention and cross-attention mechanisms, comprising four layers in total. Each layer leverages multi-head attention with 4 heads. This is followed by feature fusion, where the attention-derived features are concatenated with the original input features and subsequently refined through an MLP with layers configured as `[512, 512, 256]`. This MLP serves to project the enhanced feature vectors back to the original feature dimension, ensuring consistency across layers.

5 Training Detail

For SLAck-OV, we train our model using the same FasterRCNN-based open-vocabulary detector as OVTrack [7]. We conduct an inference of the pre-trained detector on pairs of sampled images using a dynamic score threshold described in Sec. 3.4. The maximum number of detection predictions per image is set to 50. Instead of the default intra-class non-maximum suppression (NMS), we apply class-agnostic NMS with an IoU threshold of 0.5. Following inference, we match the detection boxes with the sparse TAO ground truth using an IoU threshold of 0.7 for match determination. The matched detection boxes are then assigned instance IDs for association learning. It is possible for one ground truth box to match multiple detection boxes. During our differentiable optimal transport optimization using the Sinkhorn algorithm, we adjust the marginal distributions based on the sum of matches with ground truth. The number of Sinkhorn iterations is set to 100 during training.

Training images are randomly resized, keeping the aspect ratio, with the shorter side between 640 and 800 pixels. Pairs of adjacent frames, within a maximum interval of 3 seconds, are selected for training. Models are trained for 12 epochs using a batch size of 16, applying an SGD optimizer with an initial learning rate of 0.008 and weight decay of 0.0001. For SLAck-T and SLAck-L, all the hyper-parameters described above are the same as SLAck-OV except that we use a fixed score threshold of 0.0001 for the detection inference in detection-aware training.

6 Inference Detail

During inference, the shorter side of images is resized to 800 pixels. We set all detection-related hyper-parameters such as test score threshold and NMS settings, the same as during the training. We set the Sinkhorn iteration of exit to 100 and use a matching threshold of 0.2 ($match\_score\_thr$). A memory queue lasting 10 seconds ($memo\_length$) retains semantic, location, and appearance encodings of all objects in the tracklets, with objects expiring after a 10-second inactivity period. We provide a detailed tracking algorithm pseudocode in Algo. 1.

Compared to previous tracking algorithms [1, 4, 6–9], we significantly reduce the number of hyper-parameters during inference. We only need to set two hyper-parameters which are the *match_score_thr* and *memo_length* to decide when to match and how long the objects are stored in the tracklets.

7 Qualitative results

The qualitative results of SLAck are shown in Fig. 8. We choose all the novel classes to test our tracker's ability in real-world testing scenarios.

8 Limitations

SLAck is sensitive to video frame rate changes. SLAck joint considers location, object shape changes both spatially and temporally, and the frame rate changes between training and inference can introduce a huge domain gap. For example, if the model is trained on 1 FPS video and test on 30 FPS videos, the motion or shape changes can be very different.

Also, SLAck requires training with video labels. Despite the introduction of detection-aware training, which allows for end-to-end association training with sparse ground truth labels, the need for video annotations persists. This is particularly challenging in open-vocabulary tracking scenarios, where obtaining comprehensive video annotations is both difficult and costly. The scarcity of such annotations restricts the model's scalability and its applicability to a wider range of tracking tasks.

Algorithm 1 Simplified Tracking Algorithm

```
1: Initialization:
2: Initialize tracker parameters: $match\_score\_thr$, $memo\_length$, etc.
3: Initialize tracker state: $tracklets$, $fid\_tracklets$
4: procedure UpdateMemo($ids$, $bboxes$, $labels$, $app_embeds$, $cls\_embeds$, $frame\_id$)
5:     for each $id$, $bbox$, $app_embed$, $cls\_embed$, $label$ in new or updated tracklets do
6:         Update or add to $tracklets$
7:     end for
8:     Remove outdated tracklets based on $frame\_id$ and $memo\_length$
9: end procedure
10: procedure Match($bboxes$, $labels$, $app_embeds$, $cls\_embeds$, $frame\_id$, $img\_metas$)
11:     Match detections to existing objects in tracklets using the joint similarity matrix with $match\_score\_thr$
12:     Assign $ids$ to matched and newly detected tracklets
13:     return updated $bboxes$, $labels$, and $ids$
14: end procedure
15: Main tracking loop:
16: for each frame do
17:     Detect objects and extract features
18:     Match detections to tracklets
19:     Update memories with current frame data
20: end for
```

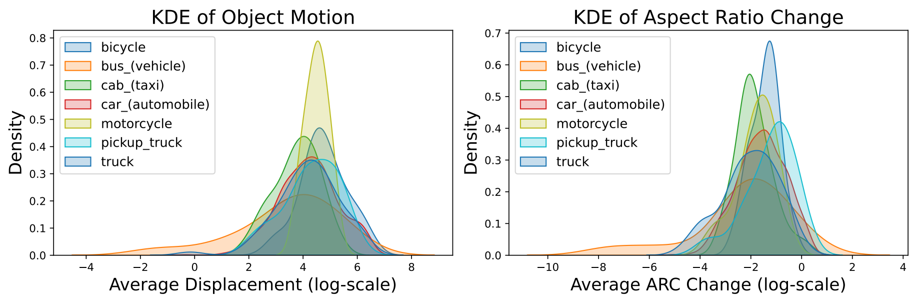

Fig. 4: The KDE of motion displacement and ARC for different classes in road vehicles.

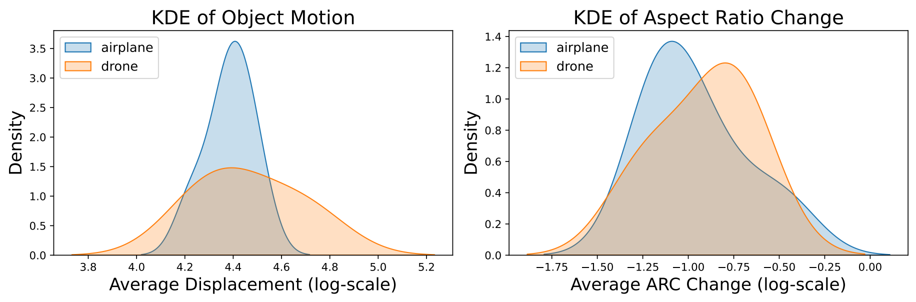

Fig. 5: The KDE of motion displacement and ARC for different classes in air vehicles.

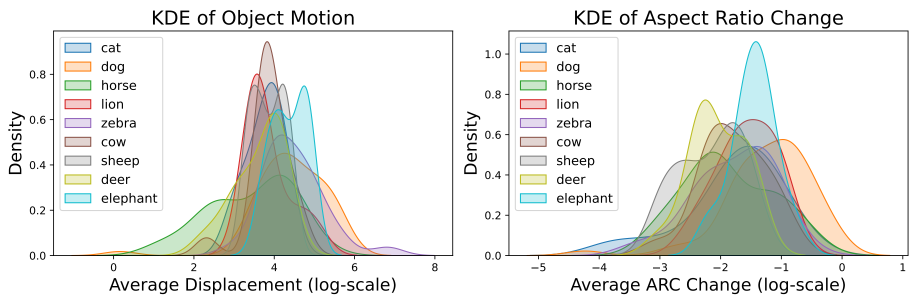

Fig. 6: The KDE of motion displacement and ARC for different classes in four-leg animals.

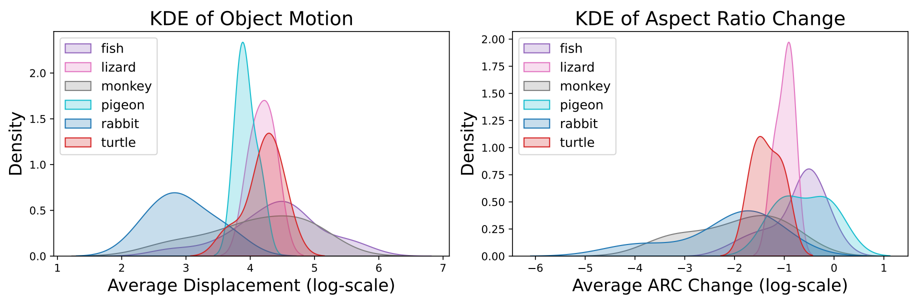

Fig. 7: The KDE of motion displacement and ARC for different classes in other animals.

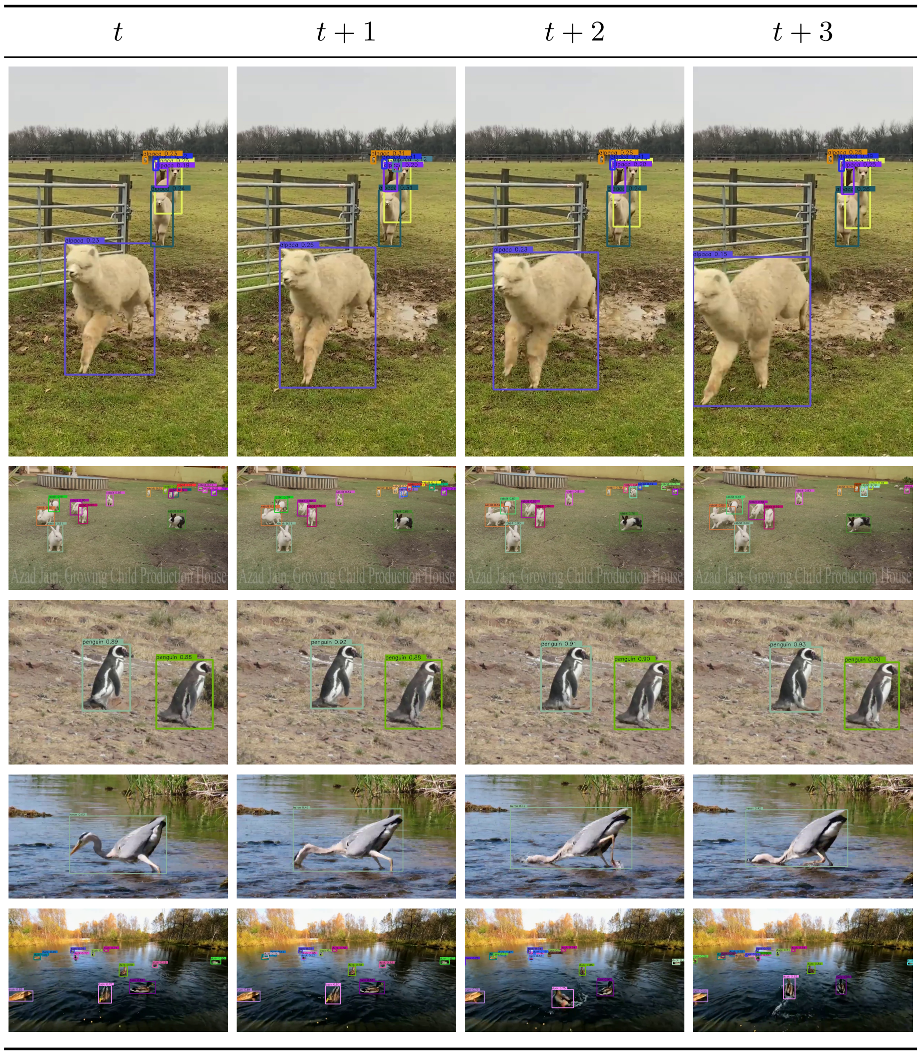

Fig. 8: Qualitative results of Open-Vocabulary Tracking. We condition SLAck on text prompts unseen during testing and successfully track the corresponding objects in the videos. The bounding box colour depicts the object's identity. We choose random internet videos to test our algorithm on diverse real-world scenarios. Best viewed digitally.

References

1. Cao, J., Pang, J., Weng, X., Khirodkar, R., Kitani, K.: Observation-centric sort: Rethinking sort for robust multi-object tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9686–9696 (2023)
2. Dave, A., Khurana, T., Tokmakov, P., Schmid, C., Ramanan, D.: TAO: A large-scale benchmark for tracking any object. In: ECCV (2020)
3. Dendorfer, P., Rezatofighi, H., Milan, A., Shi, J., Cremers, D., Reid, I., Roth, S., Schindler, K., Leal-Taixé, L.: Mot20: A benchmark for multi object tracking in crowded scenes. arXiv preprint arXiv:2003.09003 (2020)
4. Du, Y., Zhao, Z., Song, Y., Zhao, Y., Su, F., Gong, T., Meng, H.: Strongsort: Make deepsort great again. IEEE Transactions on Multimedia (2023)
5. Gu, X., Lin, T.Y., Kuo, W., Cui, Y.: Open-vocabulary object detection via vision and language knowledge distillation. arXiv preprint arXiv:2104.13921 (2021)
6. Li, S., Danelljan, M., Ding, H., Huang, T.E., Yu, F.: Tracking every thing in the wild. In: ECCV. Springer (2022)
7. Li, S., Fischer, T., Ke, L., Ding, H., Danelljan, M., Yu, F.: Ovtrack: Open-vocabulary multiple object tracking. In: CVPR (2023)
8. Pang, J., Qiu, L., Li, X., Chen, H., Li, Q., Darrell, T., Yu, F.: Quasi-dense similarity learning for multiple object tracking. In: CVPR (2021)
9. Zhang, Y., Sun, P., Jiang, Y., Yu, D., Yuan, Z., Luo, P., Liu, W., Wang, X.: Bytetrack: Multi-object tracking by associating every detection box. In: ECCV (2022)