OVTrack: Open-Vocabulary Multiple Object Tracking Supplemental Material

Siyuan Li* Tobias Fischer* Lei Ke Henghui Ding
Martin Danelljan Fisher Yu
Computer Vision Lab, ETH Zürich
https://www.vis.xyz/pub/ovtrack/

In this supplementary material, we elaborate on our experimental setup, method details, and training and inference hyperparameters. Further, we provide additional ablation studies, dataset statistics, and results of our tracker and of our data hallucination strategy.

1. Dataset statistics

Since we focus on tracking an arbitrary vocabulary of classes with our tracker, we use the only large-vocabulary MOT benchmark publicly available, namely TAO [2] in all our experiments. However, to show that our method also works on other datasets, we provide a zero-shot generalization experiment on BDD100K [10] in Sec. 4 of this appendix. Furthermore, we show qualitative results on arbitrary internet videos in Sec. 5.

TAO validation set. The 833 object classes in TAO have an overlap of 482 classes in LVIS. In the validation set of TAO, 295 of the overlapping classes are present. 35 of these classes are defined as rare, which serve as our $\mathcal{C}^{\text{novel}}$. In total, there are 109,963 annotations across 988 validation sequences for evaluation, with 2,835 annotations in $\mathcal{C}^{\text{novel}}$.

TAO test set. To evaluate open-vocabulary MOT on the TAO test set, we resort to the recently published BURST [1] dataset that provides us with test set annotations for the TAO videos. This is due to the fact that the TAO test set annotations are not publicly available. However, we need the test set annotations to split the evaluation into base and novel classes. In particular, we use the instance mask annotations in BURST to create 2D bounding boxes which serve as our ground truth for evaluation on the TAO test set.

In the test set of TAO, there are 324 of the overlapping classes mentioned above present. 33 of these classes are defined as rare, which serve as our $\mathcal{C}^{\text{novel}}$. In total, there are 164,501 annotations across 1,419 test sequences for evaluation, with 2,263 annotations in $\mathcal{C}^{\text{novel}}$.

*Equal contribution.

2. Experiment details

Training details. To train OVTrack we use a two-stage training scheme. In particular, we first train the detector for 20 epochs on LVIS [6] using standard data augmentations and without hallucinated images following [4]. We use pre-trained backbone weights from [9] which are trained self-supervised for 400 epochs on ImageNet [3]. For the first stage of training, we use SGD optimizer with a learning rate of 0.02, momentum of 0.9, weight decay of 0.0001, a batch size of 16 and decay the learning rate by a factor of 10 at epochs $[8, 16]$. In the second stage of training, we train the tracking head for 6 epochs on LVIS [6] with our hallucinated reference images. We use the same optimizer and learning rate settings and decay the learning rate at epochs $[3, 5]$.

Experiment details. For the comparison on open-vocabulary MOT, all methods train using the same training schedule and dataset versions. In particular, we use LVISv1 annotations to train our model and the baselines. The baselines, namely QDTrack [5] and TETer [7] are trained according to the schedules mentioned in the respective papers, *i.e.* 24 epochs on LVIS and a subsequent fine-tuning of the tracking head on TAO for 12 epochs. We initialize the detection modules following [4]. We train our method with a similar, but shorter schedule as described above. For the closed-set MOT comparisons, we take the same model as above and compare with the numbers reported in the respective papers. For our ablation studies, we use the same 6 epoch fine-tuning as above. For data hallucination, we use the combined LVISv1 and COCO annotations as used in [2, 5, 11]. Note that for data hallucination, we only add objects with a bounding box area greater than 64^2 to A^+.

3. Method details

We provide details of our network architecture, losses, and inference scheme. For the tracking and image heads, we use a standard *4-conv-1-fc* architecture each. The text embedding and bounding box regression, share a single head with the *4-conv-1-fc* architecture, with two parallel linear

Algorithm 1 Inference pipeline of OVTrack for associating objects across a video sequence.

Input: frame index t, object candidates $r \in P$, confidence p_r, detection embeddings $\mathbf{q}_r$, and track embeddings $\mathbf{q}_\tau$ for all $\tau \in \mathcal{T}$.

```
DuplicateRemoval(P)
for r ∈ P, τ ∈ T                          # compute matching scores
    f(r, τ) = similarity(q_r, q_τ)
end for
for r ∈ P                                 # track management
    c = max(f(r))                         # match confidence
    τ_match = argmax(f(r))                # matched track ID
    if c > β and p_i > β_obj              # object match found
        updateTrack(τ_match, r, q_r, t)   # update track
    else if p_r > γ
        createTrack(r, q_r, t)            # create new track
    end if
end for
```

layers on top for text embedding and box regression outputs. In terms of network losses, we attach the formula of $\mathcal{L}_{\text{aux}}$ described in the main paper in Sec. 4.1.

$$\mathcal{L}_{\text{aux}} = \left(\frac{\mathbf{q} \cdot \mathbf{q}'}{||\mathbf{q}||||\mathbf{q}'||} - e \right)^2, \tag{1}$$

where $e = 1$ if the two samples $\mathbf{q}, \mathbf{q}' \in Q$ have the same identity and 0 otherwise. Note also that, to better align the text embeddings $\mathbf{t}_c$ with the task at hand, we use learned context vectors following [4]. This is because CLIP is trained with image-text pairs that usually contain only a single or a few instances, unlike the potentially crowded scenes encountered in MOT.

In terms of inference, we provide the formula for the bi-softmax matching that we use for association:

$$\mathbf{s}(\tau, r) = \frac{1}{2} \left[\frac{\exp(\mathbf{q}_r \cdot \mathbf{q}_\tau)}{\sum_{r' \in P} \exp(\mathbf{q}_{r'} \cdot \mathbf{q}_\tau)} + \frac{\exp(\mathbf{q}_r \cdot \mathbf{q}_\tau)}{\sum_{\tau' \in \mathcal{T}} \exp(\mathbf{q}_r \cdot \mathbf{q}_{\tau'})} \right]. \tag{2}$$

Moreover, we employ a temporal voting scheme among the frame-level object classification results to decide the final video object category in a given test sequence. Due to the different evaluation criteria of TETA and Track mAP, we use slightly different detector post-processing for inference in our experiments. For Track mAP evaluation, we set $|P| = 300$ and use class-specific non-maximum suppression (NMS). For TETA evaluation, we set $|P| = 50$ and use class-agnostic NMS. Overall, our inference scheme is illustrated in Algorithm 1.

Data generation pipeline. As stated in Sec. 4.2 and 5.2 of the main paper, we apply data augmentations in combination with our data hallucination strategy to simulate all perturbations commonly encountered in video data. We implement this process stochastically so that the image I_{ref} is generated from a random sample of transformations. The set of transformations is composed of random resize, flip, affine transformation, color jitter, mosaic, and data hallucination.

Table 1. **Open-Vocabulary MOT Track mAP comparison.** We compare to existing trackers on TAO [2] validation and test sets. All methods use ResNet50 as backbone. All methods use Faster R-CNN [8]. Only our method does not use videos for training.

Method	Base Classes			Novel Classes		
Validation set	mAP50	mAP75	mAP	mAP50	mAP75	mAP
QDTrack [5]	14.7	5.2	10.0	8.3	3.8	6.0
TETer [7]	14.1	5.1	9.6	8.5	3.9	6.2
OVTrack	**21.0**	**10.1**	**15.6**	**23.0**	**14.5**	**18.8**
Test set	mAP50	mAP75	mAP	mAP50	mAP75	mAP
QDTrack [5]	11.6	3.3	7.5	1.6	0.4	1.0
TETer [7]	11.3	3.1	7.2	1.7	0.6	1.2
OVTrack	**17.9**	**7.7**	**12.9**	**13.2**	**3.0**	**8.2**

Table 2. **Data hallucination hyperparameters.** We show that using language prompts and geometric transformation of the input image before denoising is essential to our data hallucination strategy ('DDPM', paper Sec. 4.2). We use the TAO [2] validation set.

DDPM	Lang. prompt	Geo. trans.	TETA	LocA	AssocA	ClsA
-	-	-	32.5	48.9	31.1	17.6
✓	-	-	32.6	**49.0**	30.6	17.2
✓	✓	-	32.3	48.9	30.7	17.2
✓	-	✓	32.8	48.9	32.4	17.1
✓	✓	✓	**33.3**	48.9	**32.9**	**18.0**

4. Ablation studies and additional results

Open-vocabulary MOT. We add an additional comparison to closed-set trackers in the open-vocabulary setting using the official TAO metrics in Tab. 1. We observe that also on the official Track mAP metrics, our OVTrack outperforms existing closed-set trackers by a wide margin.

Data hallucination strategy. In Fig. 1 we illustrate a variety of hyperparameters of the data hallucination process. We experiment with varying noise levels, number of iterations, and homogenization steps and choose the parameter configuration with the visually most appealing results.

In addition, we ablate the most important hyperparameters of our data hallucination strategy quantitatively in Tab. 2. We use standard data augmentations, *i.e.* random resize and horizontal flip. We observe that using hallucinated images without language prompt or geometric augmentations fails to improve the performance of the baseline trained without any hallucinated data. When adding the geometric augmentations, however, we see a clear improvement of 1.3 points in AssocA over the baseline. Further adding the language prompt to condition the hallucination process improves the result by another 0.5 points in AssocA, culminating in a 1.8 points improvement.

Zero-shot generalization. We test the ability of our

Table 3. **Zero-shot generalization.** We test our model along with two closed set baselines, QDTrack [5] and TETer [7], on the BDD100K [10] MOT validation split. We indicate the training data used to train each model. † denotes logit masking of classes not present in BDD100K.

Method	Training	TETA	LocA	AssocA	ClsA
QDTrack†	LVIS, TAO	35.6	38.1	28.5	40.2
TETer†	LVIS, TAO	36.1	36.4	31.9	40.2
QDTrack	LVIS, TAO	32.0	25.9	27.8	42.4
TETer	LVIS, TAO	33.2	24.5	31.8	43.4
Ours	LVIS	**42.5**	**41.0**	**36.7**	**49.7**
TETer	BDD100K	58.7	47.2	52.9	76.0

tracker to adapt zero-shot to another dataset in comparison to closed-set trackers. We use the large-scale MOT benchmark BDD100K [10] for this experiment. Note that BDD100K has an overlapping class taxonomy with TAO. We apply our tracker conditioned on text prompts containing the class names in the BDD100K dataset. Further, for the closed-set baselines, we provide results where we masked out the logits of classes not present in BDD100K.

Tab. 3 shows the results using the TETA metric. Our tracker exhibits a much better transfer ability, outperforming the closed-set baselines by at least 6.4 points in TETA. Our OVTrack improves over the baselines in localization, association and particularly in classification, where the gap is the biggest with 6.3 points in ClsA. Overall, we show that we are able to bridge the gap to the upper bound, *i.e*. a tracker trained on the target dataset.

5. Qualitative results

For the qualitative results in this supplementary material, we set $\gamma = \frac{1}{|\mathcal{C}|+1}$ where $\mathcal{C}$ is the number of prompts in the video to have a more rigorous detection filtering.

Data hallucination strategy. We visualize the results of different hyperparameters in our diffusion process in Fig. 1. We choose the parameters with the visually most appealing results, $\delta_0 = 0.75$, $K = 50$ and $\eta = 0.01$. We observe that choosing a too high δ_0 leads to divergence from the original image content, while too little noise leads to insufficient fidelity. Increasing the number of iterations K does not lead to an obvious improvement in visual quality, so we choose $K = 50$ to speed up the image generation process. Finally, having no homogenization, *i.e*. setting $\eta = 0.0$ leads to noticeable artifacts. On the other hand, a higher η of 0.1 leads to subtle, but significant appearance perturbation of the object, which is also undesirable for preserving its identity.

In addition, we illustrate examples of our final data hallucination strategy in Fig. 3. We visualize examples from the LVIS dataset, where in each row we plot both annotations and images and show the generated versions and the original images.

Qualitative results and failure cases. We show qualitative results and failure cases of our method in Fig. 2. We observe that our method does well on tracking, and is able to generalize even to very exotic classes, such as pikachu. However, fine-grained classification is still challenging. In particular, in the bottom row of the figure, our method fails to distinguish the sea gull from the puffin, wrongly classifying it as another sea gull. Furthermore, our detection is not perfect, as can be seen by the false negative in the 7th row ($t + 4$). In addition, the 6th row exhibits an ID switch between $t + 3$ and $t + 4$.

References

[1] Ali Athar, Jonathon Luiten, Paul Voigtlaender, Tarasha Khurana, Achal Dave, Bastian Leibe, and Deva Ramanan. Burst: A benchmark for unifying object recognition, segmentation and tracking in video. *arXiv preprint arXiv:2209.12118*, 2022. 1

[2] Achal Dave, Tarasha Khurana, Pavel Tokmakov, Cordelia Schmid, and Deva Ramanan. Tao: A large-scale benchmark for tracking any object. In *ECCV*, 2020. 1, 2

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 1

[4] Yu Du, Fangyun Wei, Zihe Zhang, Miaojing Shi, Yue Gao, and Guoqi Li. Learning to prompt for open-vocabulary object detection with vision-language model. In *CVPR*, 2022. 1, 2

[5] Tobias Fischer, Jiangmiao Pang, Thomas E Huang, Linlu Qiu, Haofeng Chen, Trevor Darrell, and Fisher Yu. Qdtrack: Quasi-dense similarity learning for appearance-only multiple object tracking. *arXiv preprint arXiv:2210.06984*, 2022. 1, 2, 3

[6] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *CVPR*, 2019. 1

[7] Siyuan Li, Martin Danelljan, Henghui Ding, Thomas E Huang, and Fisher Yu. Tracking every thing in the wild. In *ECCV*, 2022. 1, 2, 3

[8] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 2

[9] Fangyun Wei, Yue Gao, Zhirong Wu, Han Hu, and Stephen Lin. Aligning pretraining for detection via object-level contrastive learning. *NeurIPS*, 2021. 1

[10] 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 *CVPR*, 2020. 1, 3

[11] Xingyi Zhou, Tianwei Yin, Vladlen Koltun, and Philipp Krähenbühl. Global tracking transformers. In *CVPR*, 2022. 1

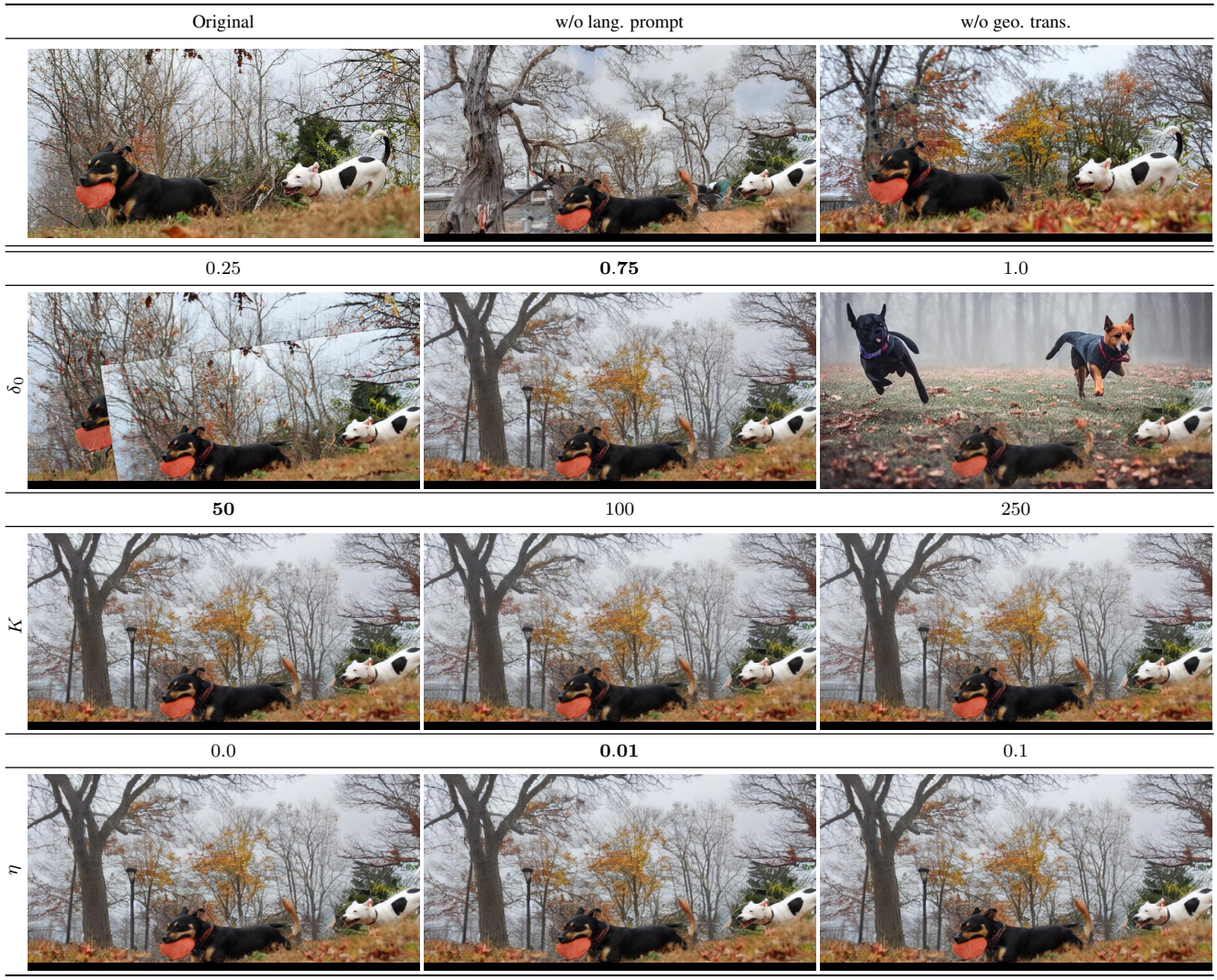

Figure 1. **Data hallucination hyperparameters.** We show the influence leaving out of language prompt and geometric transformation. In addition, we examine different values for the noise level δ_0, the number of denoising steps K and the homogenization threshold η. We indicate the value we choose for each of those parameters in bold.

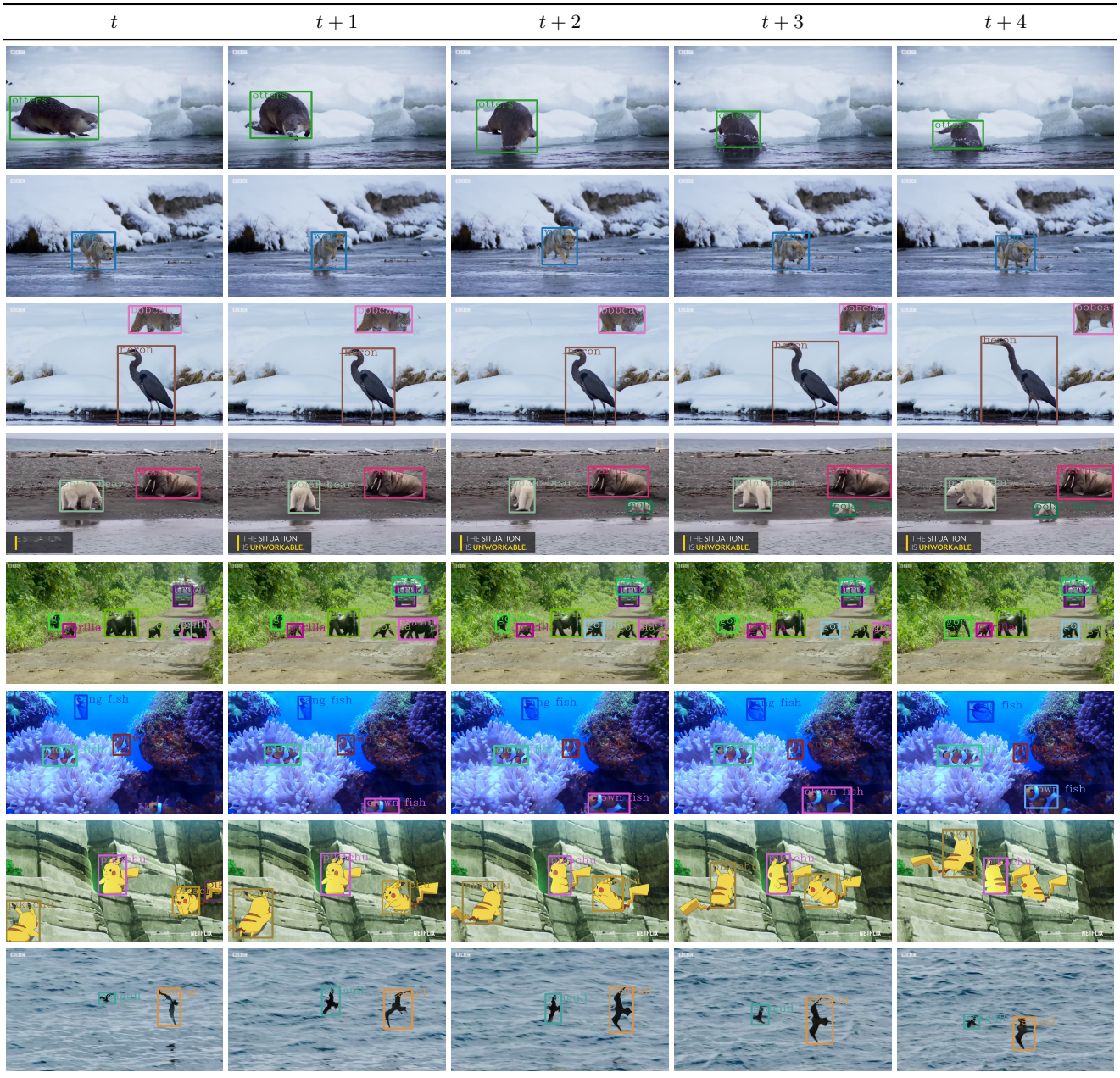

Figure 2. **OVTrack qualitative results and failure cases.** We condition our tracker on text prompts unseen during training and successfully track the corresponding objects in the videos. The box color depicts object identity. We choose random internet videos to test our algorithm on diverse real-world scenarios. The bottom row shows the difficulty of fine-grained classification, where our method fails to distinguish the puffin from the sea gull. Best viewed digitally.

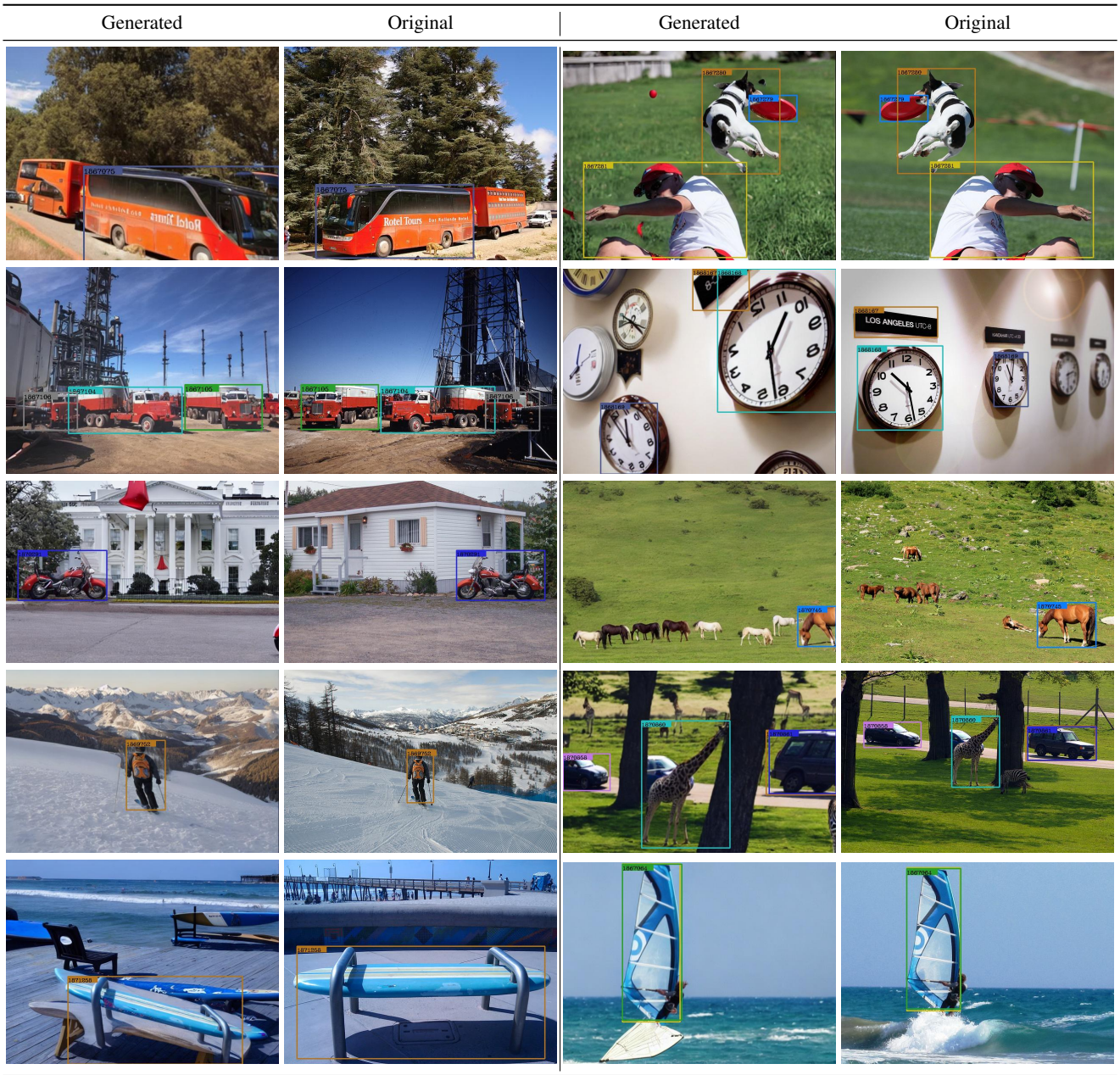

Figure 3. **Data hallucination examples.** We provide examples of our data hallucination strategy including annotations on the LVIS dataset. We plot the generated versions and the original for comparison. The ids on the bounding boxes depict the identity.