SemTrack: A Large-scale Dataset for Semantic Tracking in the Wild (Supplementary Material)

Pengfei Wang[1,*], Xiaofei Hui[1,2,*], Jing Wu[1,*], Zile Yang[1,*], Kian Eng Ong[1,*], Xinge Zhao[1], Beijia Lu[1], Dezhao Huang[3], Evan Ling[3], Weiling Chen[4,‡], Keng Teck Ma[3], Minhoe Hur[3], and Jun Liu[1,2,†]

[1] Singapore University of Technology and Design
[2] Lancaster University
[3] Hyundai Motor Group Innovation Center in Singapore (HMGICS)
[4] Tianqiao and Chrissy Chen Institute

1 Details of Collection and Annotation of Data and Checking of Annotations

1.1 Data Collection

In order to achieve comprehensive tracking of various types of targets in different scenarios, we use videos from publicly available datasets, *i.e.* YFCC100M [15], TAO [1], ImageNet-VidVRD [14], VidOR [13], HACS [21], AVA [5], GOT-10K [7], and ILSVRC2016 [12]. These videos encompass different types of indoor and outdoor scenes in a wide range of scenarios, ranging from the daily life encounters and interactions of individuals to commercial and factory settings to outdoor settings (*e.g.*, desert, forest, and river). Having a diverse representation of objects and interactions in a wide range of complex and challenging scenarios that reflects the diversity in the real world is helpful to develop and evaluate robust and generalizable tracking models that are of great practical value to handle various complex real-world scenarios, thus benefitting a large and diverse range of applications.

We first curate the list of actions and interactions from AVA [5] and form a vocabulary list of interaction classes (see Supp. Sec 2.3 for more details). We define each interaction to minimize ambiguity. Using this list as a starting point, we ask our annotators to screen the videos and shortlist suitable videos that contain rich semantic trajectories. The criteria for the selection include:

1. The video must contain at least one target.
2. Interactions seen in the video should be found in our initial list of interaction classes. If there are other interactions, the annotator will define them and propose adding them to the vocabulary list, so that this list can be expanded to comprehensively cover a wide range of interactions that occur in real life. All the annotators meet again to standardize and refine this vocabulary list.

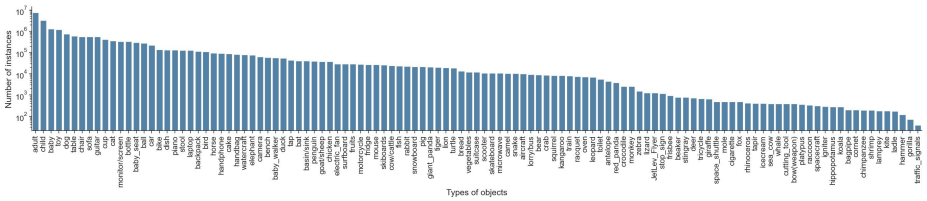

Fig. 1: Distribution of different object classes in terms of number of instances

Table 1: Different types of interactions in SemTrack

SuperCategory	Interaction
Movement-related and daily activities	bite, carry, chase, close, cut, fall_off, feed, fill(glass), fly_with, grab, hit, hold, kick, knock, lick, lift, light(cigarette), open, pat, point_to, press, pull, push, release, smell, squeeze, throw, use, wave
Hygiene	clean, trim_nails
Sports	swim_with, catch(ball), serve(ball), shoot
Music	play(instrument)
Commuting	ride, drive, get_off, get_on
Social interactions and display of affection	lean_on, caress, fight, follow, hold_hand_of, hug, kiss, play_with, shake_hand_with, shout_at, speak_to, wave_hand_to

1.2 Data Annotations

The entire annotation process is conducted using the Label Studio platform [16]. We leverage existing annotations in the datasets mentioned in the earlier subsection and tailor them for our semantic tracking.

Selection of target: In each video, the object with rich semantic information is selected as the target. If there is another object that has rich interactions with some other affiliated object, it can also be selected as a target. In other words, there can be multiple target objects in a video clip.

Target initialization sentence: The target initialization sentence is a statement that describes the target in its first frame of occurrence so that it can be used to detect and localize the target for the model to start tracking (*i.e.*, target tracking initialization). We manually describe the target based on its attribute, including its color and position. The combination of the target's attributes and actions (*e.g.*, stand, sit) gives rise to a rich diversity of target descriptions.

Annotation of bounding boxes and trajectories of target and affiliated objects: Based on the selected target in the video, the objects that interact with the target are annotated as affiliated objects of the target. We follow AVA style [5] to manually annotate the bounding boxes of the target and affiliated objects across frames. If the source dataset contains trajectories, we use those existing trajectories.

Annotation of object classes of affiliated objects: We annotate the object classes of the affiliated objects.

Annotation of interactions: We annotate the interactions between the target and its affiliated object based on the predefined interaction vocabulary list.

1.3 Checking of Annotations

We conduct two rounds of checking.

In round 1, each annotator will check the videos that are different from the ones that he/she has annotated. They will ensure that:

- every target initialization sentence only defines one target and can clearly identify and distinguish it from other objects,
- the bounding box is tightly bounded around the target or affiliated objects, and the annotated bounding boxes (e.g., not severely occluded) are considered valid following previous datasets (e.g., [1, 7, 12, 13]),
- the object classes are correctly identified,
- and the interaction classes are correctly annotated.

In round 2, the video samples are re-distributed to a separate set of annotators (*i.e.*, the annotator in round 1 is different from round 2) to check for the same items as mentioned in round 1. If there is any discrepancy, a more experienced annotator will take the average coordinate value of their bounding boxes and disambiguate the annotation.

2 Statistics of SemTrack

2.1 Various Scenes

Our SemTrack dataset covers 12 different types of scenes, including indoor and outdoor scenes.
The indoor scenes include:

1. Home (*e.g.*, kitchen, bathroom, bedroom),
2. Mall (*e.g.*, supermarket),
3. Factory,
4. Classroom,
5. Stage (*e.g.*, stage performance).

The outdoor scenes include:

1. Desert,
2. Forest,
3. Transportation vehicle (*e.g.*, train, bus),
4. Road (*e.g.*, street, walkway),
5. Outdoor sports arenas (*e.g.*, horse riding, skiing),
6. Zoo and safari,
7. River, beach, underwater (*e.g.*, scuba diving).

2.2 Diverse types of targets and affiliated objects

Our SemTrack dataset contains annotations of 115 different types of objects, spanning 10 different supercategories. Fig. 1 shows the distribution of various categories. The 10 supercategories are as follows:

1. Person
2. Animals (*e.g.*, rabbit, cat, and dog)
3. Home Stuff
   - Furniture (*e.g.*, sofa)
   - Electronics (*e.g.*, monitor, oven)
   - Kitchen wares (*e.g.*, bottle, cup)
   - Toiletries (*e.g.*, basin, tap)
   - Baby items (*e.g.*, baby seat, toy)
   - Other general items (*e.g.*, cutting tool)
4. Personal items (*e.g.*, handbag, suitcase)
5. Food (*e.g.*, ice-cream)
6. Tools (*e.g.*, hammer)
7. Sports (*e.g.*, ball, baseball bat)
8. Musical instruments (*e.g.*, piano)
9. Road and vehicle (*e.g.*, bicycle, car)
10. Weapon (*e.g.*, hunting bow)

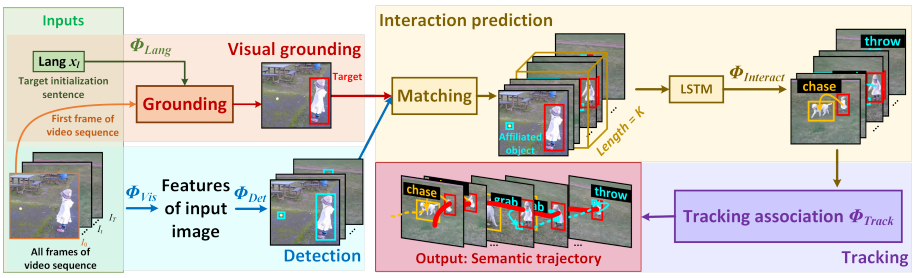

Fig. 2: Architecture of SemTracker model consisting of four modules: visual grounding, object detection, interaction prediction, and object tracking.

2.3 Interaction Vocabulary

We select videos that contain interactions that commonly happen in an individual's daily life, ranging from general daily interactions and encounters to social interactions to commuting to sports. Details of the interactions are listed in Table 1.

3 Further Elaboration of our SemTracker Model

3.1 Model Architecture

Our SemTracker consists of four modules as shown in Fig. 2 — a visual grounding module, an object detection module, an interaction prediction module, and a multiple object tracking module.

Visual grounding module. The visual grounding module is used to locate the target in the first frame of the video sequence. We follow [3, 17] and use the pre-trained language model BERT [8] to extract language embedding of the input sentence and predict the location of the target.

Object detection module. The object detection module is applied to detect the objects in each frame, and obtain their locations and object classes. These detections are then used in the interaction prediction module and the object tracking module. We adopt the YOLOX [4] model for object detection.

After obtaining the location of the objects, RoI Align Φ_{RoI} [6] is applied to extract the intermediate features $f_{t,i}$ corresponding to the detected bounding boxes (t represents the frame index, i represents the index of the detected object). These intermediate features are then used to determine the target and predict the interaction classes in the interaction prediction module.

Interaction prediction module. The interaction prediction module aims to determine the target in each frame and predict the interactions between the target and affiliated objects. The target is determined by matching the image patch from the visual grounding module and the detected objects in each frame. We use the cross-correlation function to perform the matching following [3]. The object with the highest matching score is determined as the target. The other objects are designated as the candidate affiliated objects.

After obtaining the target and the candidate affiliated objects, the model then aims to find out whether the candidate affiliated objects are interacting with the target and what interactions they are having. Intuitively, information in the preceding frames can be useful in predicting the interaction in the subsequent frames. Hence, here we adopt LSTM to process the information in consecutive frames. Specifically, the intermediate features of both the target and the candidate affiliated objects in the previous frames are fed into the LSTM modules. We construct three parallel LSTM modules, with each taking in intermediate features obtained at a different scale in the feature pyramid network (FPN) [4, 10] in the object detection module. The output features of the three LSTM modules are then concatenated and fed into an interaction prediction head [19] to predict the interaction class of each pair of the target and the candidate affiliated objects. We also include a "None" class to indicate that there is no interaction between the target and the candidate affiliated object during this process. Hence, only those candidate affiliated objects that have valid interactions with the target (*i.e.*, having predicted interaction class that is not "None") are then determined and confirmed as the affiliated object.

Multiple object tracking module. Now that the target and the affiliated objects are identified and confirmed, we then obtain the positional trajectories

of these objects with the object tracking module. We adopt the ByteTrack [20] model to associate the bounding boxes across frames for each object.

In summary, the SemTracker model is made up of these four modules to predict the semantic trajectory of the target (*i.e.*, bounding boxes of the target, bounding boxes and classes of the affiliated objects of this target, and the types of interactions between them over time).

3.2 Network Training and Experiment Details

Overall loss function. The overall loss function of our SemTracker model (Equation 1) is the sum of the object detection loss $\mathcal{L}^{det}$ (same as YOLOX [4]) and the interaction prediction loss $\mathcal{L}^{int}$.

$$\mathcal{L} = \mathcal{L}^{det} + \mathcal{L}^{int} \tag{1}$$

The interaction prediction loss is the cross-entropy loss between the predicted interaction class p_i^{int} and the ground truth interaction class y_i^{int}:

$$\mathcal{L}^{int} = (-\frac{1}{N_{obj}} \sum_{i=1}^{N_{obj}} [y_i^{int} \log(p_i^{int})]) \tag{2}$$

where N_{obj} refers to the total number of interaction classes.

More implementation details. The dataset is split into 80% for training, 10% for validation, and 10% for testing. We release the split information in the dataset. In this paper, we evaluate the method on a subset of the testing set as representatives especially focusing on people (e.g., adult and baby). The network is trained on four NVIDIA 3090 graphic cards for 10 hours. The batch size is 32. We train the model with the initial learning rate set at 2e-5 and weight decay at 0.0005. Adam optimizer is used with a momentum of 0.9. In our meta-learning method, we set the learning rates α and β (in Equation 3 in the main paper) to 0.00005 and 0.001 respectively.

4 Our Proposed Evaluation Metric

We propose a new evaluation metric Semantic Tracking-mAP (ST-mAP) to evaluate the performance of methods for Semantic Tracking. Each predicted sample consists of the position of the target and an affiliated object, as well as the interactions between them, which is compared with the ground truth. If there are no interactions at a position in a trajectory for the ground truth, the affiliated object position and interaction are considered empty.

To calculate the ST-mAP, we need to determine the true positive, false negative, and false positive. In order to determine these values, we perform a primary one-to-one matching for all the ground truth and prediction using the Hungarian algorithm [9] that optimizes the match based on the 3D IoU of the bounding boxes and the object classes following Track-mAP [1] and penalizes those predictions that are missed or extraneous. This will ensure that each of

the ground truth trajectory is matched to at most one predicted trajectory and vice versa. Then, we calculate ST-mAP with consideration of the accuracy of the predicted bounding boxes of the target object, the accuracy of the predicted bounding boxes of the affiliated object, and the accuracy of the predicted interaction classes. The details are elaborated below.

1. $IoU_{3D}(G^{tar}, P^{tar})$: The 3D IoU [1] between the predicted bounding boxes and ground truth bounding boxes across the frames for the target is larger than a threshold (*e.g.*, 0.5).
2. $IoU_{3D}(G^{aff}, P^{aff})$: The 3D IoU [1] between the predicted bounding boxes and ground truth bounding boxes across the frames for the affiliated object is larger than a threshold (*e.g.*, 0.5).
3. γ_{int}: The percentage of correctly predicted interactions along the trajectory is larger than a threshold (*e.g.*, 0.7).

An illustration of the calculation of the three conditions for a trajectory to be considered as true positive is shown in Fig. 3. The calculation of the 3D IoU for both the target and the affiliated object along the trajectory follows the calculation that is used in [1].

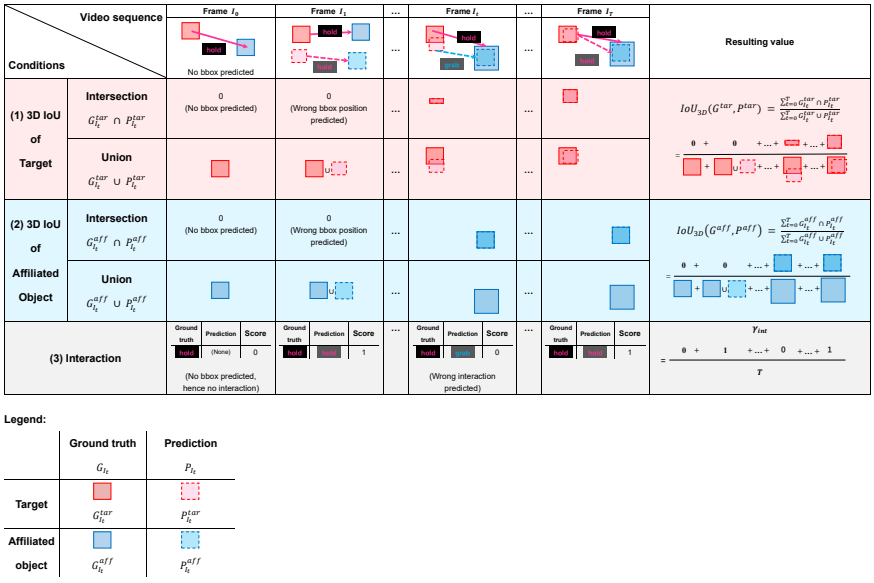

Fig. 3: Illustration of the calculation of the three conditions for a sample to be considered as true positive.

Hence, those prediction that are correctly matched to ground truth are now considered as truth positive. Conversely, those ground truth that are not matched (*i.e.*, missed predictions) are defined as false negative, and those prediction that are not matched (*i.e.*, extraneous predictions) are defined as false positive. Based on the values of truth positive, false negative, false positive, the precision-recall curve can be plotted. Thus, the ST-AP and ST-mAP (Equation 4 and 5 in the main paper) can be calculated accordingly.

5 Additional Ablation Studies and Further Analysis

5.1 Ablation Studies on Target Initialization

There are different ways to initialize the target in our SemTrack dataset, for example:

- using the target initialization sentence to initialize the tracking of the target in each video [17],
- or using the target's bounding box (bbox) in the first frame to initialize tracking in each video [11, 18].

To evaluate how a model's performance is affected by how the target is initialized, we conduct experiments with the following settings:

- **(1) sentence initialization** that uses the target initialization sentence to initialize each target;
- **(2) bbox initialization** that uses the ground truth bounding box in the first frame to initialize each target;
- **(3) ground truth bbox for all frames** that provides the ground truth bounding boxes of the target for all frames.

Table 2: Results of how target initialization affects our SemTracker model (that uses YOLOX-X [4] and ByteTrack [20]).

Setting	Initialization	Evaluation Metrics		
		Track-mAP ↑	HOTA ↑	ST-mAP ↑
1	sentence initialization	10.656	18.854	8.668
2	bbox initialization	12.344	20.513	10.154
3	ground truth bbox for all frames	18.423	24.054	16.703

The experimental results are listed in Table 2. From the results, we observe that how a target is initialized does influence the overall performance in our proposed Semantic Tracking benchmark. Specifically, we observe that using setting 3 (ground truth target bounding box for all frames), the performance is improved greatly.

Comparing the results of settings 1 (sentence initialization) and 2 (bbox initialization), we see that initializing the target using setting 1 (sentence initialization) is more challenging than 2 (bbox initialization). With the different settings available in our SemTrack dataset, our dataset shall facilitate the community to develop and evaluate robust and accurate tracking models.

5.2 Results of Single Object Tracking Metrics

As our SemTrack dataset can also be used to evaluate the traditional Single Object Tracking (SOT) task, we evaluate our dataset on this task that tracks the target's positional trajectory. More specifically, we perform a One Pass Evaluation (OPE) and evaluate the target's positional trajectory based on widely used SOT metrics [11,17]: Success, Precision, and Normalized Precision. We conduct experiments whereby the target is initialized either by a **target initialization sentence** or a **bounding box**. The results are shown in Table 3.

Table 3: Results of single object tracking.

Initialization	Evaluation Metrics		
	Success ↑	Norm-Precision ↑	Precision ↑
target initialization sentence	0.33	0.41	0.20
bounding box	0.36	0.53	0.25

While the results show that it may be more challenging to track a target using target initialization sentence, considering how target initialization sentence for tracking can tremendously benefit the user (*i.e.*, more convenient and intuitive for a user [2,17]), it is still worthwhile to build more robust tracking models that leverage the benefits of using target initialization sentence. Our challenging dataset shall facilitate the community to develop and evaluate robust and accurate tracking models.

References

1. Dave, A., Khurana, T., Tokmakov, P., Schmid, C., Ramanan, D.: TAO: A Large-Scale Benchmark for Tracking Any Object, pp. 436–454 (10 2020). https://doi.org/10.1007/978-3-030-58558-7_26
2. Fan, H., Bai, H., Lin, L., Yang, F., Chu, P., Deng, G., Yu, S., Huang, M., Liu, J., Xu, Y., et al.: Lasot: A high-quality large-scale single object tracking benchmark. International Journal of Computer Vision **129**, 439–461 (2021)
3. Feng, Q., Ablavsky, V., Bai, Q., Sclaroff, S.: Siamese natural language tracker: Tracking by natural language descriptions with siamese trackers. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2021)
4. Ge, Z., Liu, S., Wang, F., Li, Z., Sun, J.: Yolox: Exceeding yolo series in 2021. arXiv preprint arXiv:2107.08430 (2021)

5. Gu, C., Sun, C., Ross, D.A., Vondrick, C., Pantofaru, C., Li, Y., Vijayanarasimhan, S., Toderici, G., Ricco, S., Sukthankar, R., Schmid, C., Malik, J.: Ava: A video dataset of spatio-temporally localized atomic visual actions. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6047–6056 (2018). https://doi.org/10.1109/CVPR.2018.00633
6. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 2961–2969 (2017)
7. Huang, L., Zhao, X., Huang, K.: Got-10k: A large high-diversity benchmark for generic object tracking in the wild. IEEE Transactions on Pattern Analysis and Machine Intelligence (2019)
8. Kenton, J.D.M.W.C., Toutanova, L.K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of naacL-HLT. pp. 4171–4186 (2019)
9. Kuhn, H.W.: The hungarian method for the assignment problem. Naval research logistics quarterly **2**(1-2), 83–97 (1955)
10. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 936–944 (2017). https://doi.org/10.1109/CVPR.2017.106
11. Muller, M., Bibi, A., Giancola, S., Alsubaihi, S., Ghanem, B.: Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 300–317 (2018)
12. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV) **115**(3), 211–252 (2015). https://doi.org/10.1007/s11263-015-0816-y
13. Shang, X., Di, D., Xiao, J., Cao, Y., Yang, X., Chua, T.S.: Annotating objects and relations in user-generated videos. In: Proceedings of the 2019 on International Conference on Multimedia Retrieval. pp. 279–287. ACM (2019)
14. Shang, X., Ren, T., Guo, J., Zhang, H., Chua, T.S.: Video visual relation detection. In: ACM International Conference on Multimedia. Mountain View, CA USA (October 2017)
15. Thomee, B., Shamma, D.A., Friedland, G., Elizalde, B., Ni, K., Poland, D., Borth, D., Li, L.J.: Yfcc100m: The new data in multimedia research. Commun. ACM **59**(2), 64–73 (jan 2016). https://doi.org/10.1145/2812802, https://doi.org/10.1145/2812802
16. Tkachenko, M., Malyuk, M., Holmanyuk, A., Liubimov, N.: Label Studio: Data labeling software. https://github.com/heartexlabs/label-studio (2020-2022), open source software available from https://github.com/heartexlabs/label-studio
17. Wang, X., Shu, X., Zhang, Z., Jiang, B., Wang, Y., Tian, Y., Wu, F.: Towards more flexible and accurate object tracking with natural language: Algorithms and benchmark. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 13763–13773 (June 2021)
18. Wu, Y., Lim, J., Yang, M.H.: Online object tracking: A benchmark. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2411–2418 (2013)
19. Zellers, R., Yatskar, M., Thomson, S., Choi, Y.: Neural motifs: Scene graph parsing with global context. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5831–5840 (2018)
20. Zhang, Y., Sun, P., Jiang, Y., Yu, D., Weng, F., Yuan, Z., Luo, P., Liu, W., Wang, X.: Bytetrack: Multi-object tracking by associating every detection box (2022)

21. Zhao, H., Torralba, A., Torresani, L., Yan, Z.: Hacs: Human action clips and segments dataset for recognition and temporal localization. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 8668–8678 (2019)