CaptionFormer: Unified Segmentation, Tracking, and Captioning for Spatio-Temporal Objects

Supplementary Material

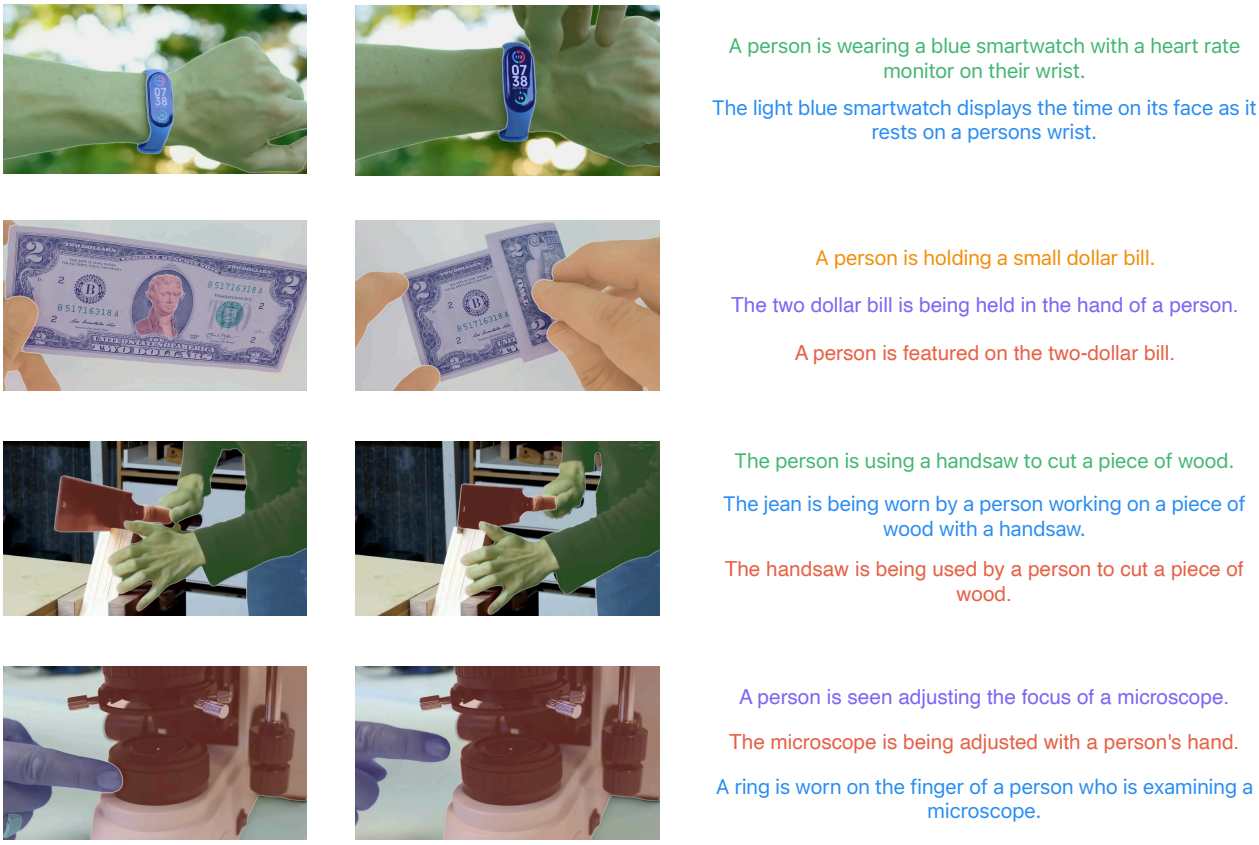


Figure 6. More qualitative examples obtained with CaptionFormer on our LV-VISCap dataset.

In this supplementary material, we show qualitative results from our method in Section A, present additional ablations in Section B, discuss failure cases and limitations in Section C, and share more details about the datasets, the method, and the implementation in Section D.

A. Qualitative results

We show qualitative DVOC results from CaptionFormer on the LV-VIS [62] dataset with (mask, caption) pairs predictions, and on the VidSTG dataset [78] with (box, caption) pairs in Figures 6 and 7 respectively. These examples show that CaptionFormer has learned to predict captions that focus on the localized objects while integrating high-level scene understanding.

On LV-VIS (Fig. 6), CaptionFormer is able to produce descriptive captions for each objects including related context, even in the case of a scene including a high number of objects. We note that, when tuned on VidSTG (see Fig. 7), CaptionFormer produces less informative and less descriptive captions. This is due to the VidSTG annotation captions being designed for grounding rather than for captioning or DVOC, and thus being only little descriptive or informative, and overlooking to the global context. In contrast, when trained on LVISCap and LV-VISCap, CaptionFormer visually generates much richer and accurate descriptions, further highlighting the value of our synthetic captions.
Overall, CaptionFormer effectively learns to jointly segment, detect, track and caption object trajectories.

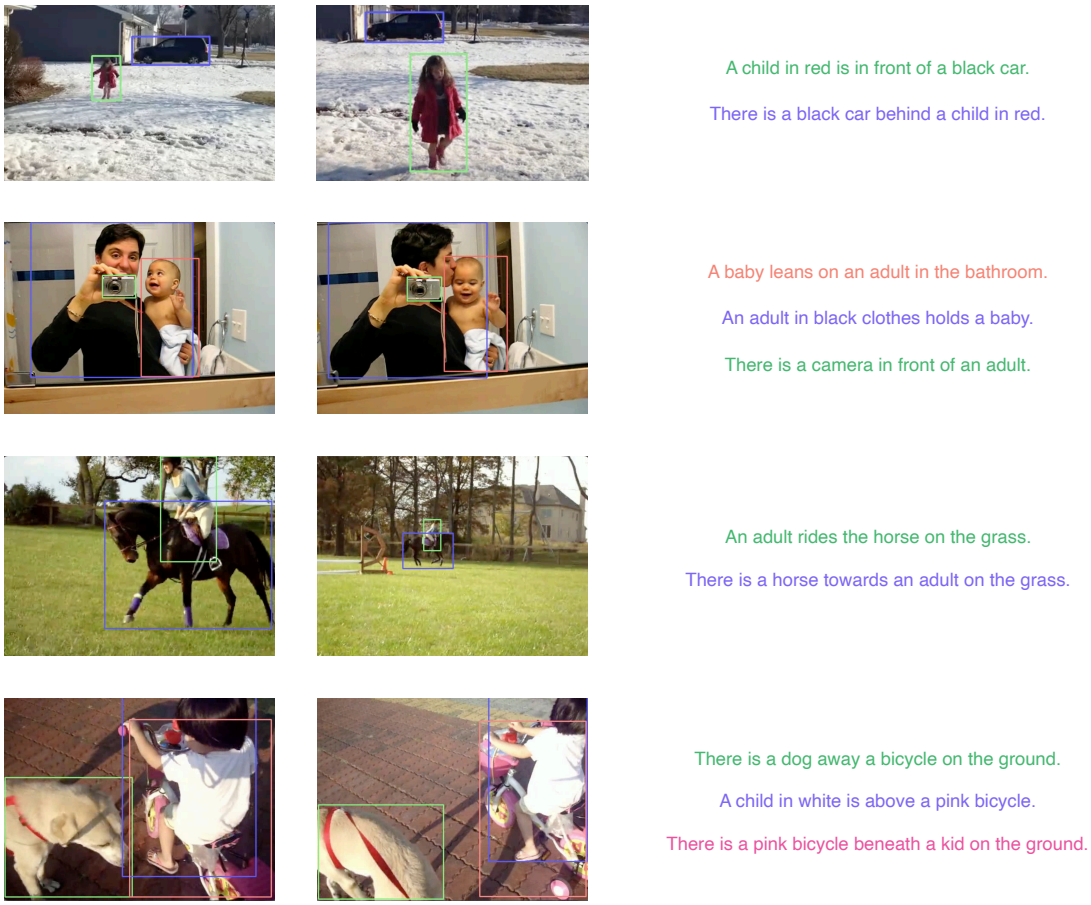


Figure 7. Qualitative examples, obtained with CaptionFormer on the VidSTG dataset.

Table 9. **Impact of the prompting strategy on caption quality**. Scores are given by a human evaluator from 0 to 2 (incorrect, partially correct, or correct) on a subset from the LV-VIS validation set, and brought to 0-100 range. For the mask visual prompt experiments, we use our best prompt with either the object's bounding boxes or center point coordinates as a localization cue in the text prompt.

Visual prompt	Prompting method	Average rating	Rating percentage		
			0	1	2
bounding boxes	single frame	26.8	66.3	13.9	19.9
	+ multiple frames	27.1	68.7	8.4	22.9
	+ detailed instructions	29.5	65.0	10.8	24.10
	+ category labels	80.7	10.8	16.9	72.3
	+ bounding box coordinates	83.1	9.6	14.5	75.9
	+ bounding box area	<u>84.3</u>	7.8	15.7	76.5
	+ few shot examples	**85.1**	9.1	11.5	79.4
mask boundaries	center point coordinates	75.9	17.5	13.2	69.3
	bounding box coordinates	77.1	15.7	14.5	69.9

B. Additional ablations

Prompting strategy. In Table 9, we show the distribution of ratings given by the human annotator depending on the prompting strategy. Using the box visual prompt yields a better focus on the queried object and more correct object captions. Importantly, giving the category labels in the prompt helps the model to generate more accurate object captions. Overall, the rate of correct captions with the best prompt indicates good quality for our synthetic object captions.

Table 10. **Impact of the inference clip on LV-VIScap.** All models are trained on LVIScap then LV-VIScap training set.

Clip length	mAP
3	26.0
4	26.1
5	<u>26.6</u>
6	26.3
7	**26.8**

Clip Length. We study the impact of the clip length on Video Instance Segmentation (VIS) performance in Table 10, and observe that increasing clip length leads to slightly better segmentation results. Overall the impact on performance is relatively small. For LV-VIS experiments, we use a clip length of $T = 5$.

Table 11. **Impact of the tracking module on VidSTG DVOC.** All models are trained on COCO, LVISCap and LV-VISCap, and finetuned on VidSTG using temporal aggregation with 8 clips.

Tracking method	CapA	DetA	AssA	CHOTA
OVFormer module [21]	51.9	**67.0**	58.2	58.7
Top-K enhanced module [82]	**52.7**	66.8	**71.0**	**63.0**

Tracking module. In Table 11, we show that the top-k enhanced tracking [82] is important for tracking objects effectively on the challenging VidSTG dataset [78], as seen in the AssA, CapA and CHOTA scores.
We attribute this difference to the numerous objects that disappear for a significant number of frames in the long videos of VidSTG. The top-K approach uses a memory bank of tracked queries that helps keeping track of these entities, while they are lost using the i to $i + 1$ tracking from OVFormer [21].

Table 12. **Zero-shot DVOC performance** on the VidSTG validation set. Our model is pre-trained on COCO, LVISCap and LV-VISCap, while DVOC-DS is pre-trained on COCO, VG, SMiT, Aug-COCO.

Method	CapA	DetA	AssA	CHOTA
DVOC-DS [81]	9.8	**51.4**	59.6	31.1
CaptionFormer (ours)	**10.4**	50.2	**71.3**	**33.4**

Generalization to Out-of-Distribution data. We report zero-shot performance on the VidSTG validation set in Table 12, and compare with DVOC-DS [81]. We observe that our model achieves better performance in the zero-shot setting for captioning and tracking. The DVOC-DS performs slightly better for detection, which can be explained by their larger pretraining set for detection.

Table 13. **Inference speed comparison** on a subset of the VidSTG validation set using a single A100 GPU.

Method	FPS
OW-VisCapTor [18]	6.4
CaptionFormer (ours)	**7.4**

Inference speed. We compare the inference speed of our model to OW-VisCapTor [18] in Table 13. The OW-VisCapTor numbers were provided by the authors for a subset of the VidSTG videos and a A100 hardware setup. We run our approach on the same subset and hardware for fair comparison.

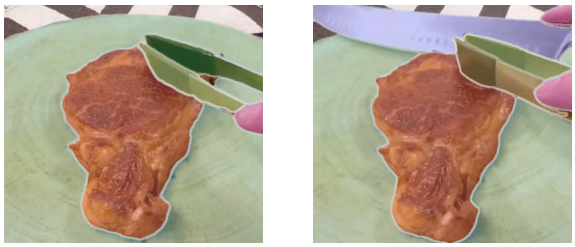
C. Failure Cases and limitations

C.1. Failure cases

We observe 3 main types of failure cases for our approach and illustrate them in Figure 8:
(i) Recognition error: in the case of ambiguous context, blurred instance or rare categories, CaptionFormer might fail to recognize the object it is describing, sometimes leading to a wrong denomination (e.g. here, a rare "pair of tongs" is incorrectly denominated as "knife").
(ii) Inconsistent captions : in similar situations, the captions produced by CaptionFormer can be inconsistent when referring to the same object in different captions.
(iii) Detection/segmentation error : In case of complex movement, appearance change or occlusion, CaptionFormer sometimes fails to detect, segment, or track objects, leading to missing captions (e.g. here, the fish is not detected in the beak of the heron, and thus has no associated caption).

C.2. Limitations

While our approach outperforms the state-of-the-art in dense video object captioning, there is still room for improving localization and captioning. Localization sometimes fails, in particular for small objects. Furthermore, the automatically generated captions are, in some cases, too generic, and can mix up two objects of the same class in the video. Future work could investigate different automatic captioning techniques for DVOC, for example based on an approach such as Ref-SAV[73], which generates captions in multiple steps to separate appearance from motion description. Eventually, objects in the videos often perform a single or few actions, and we believe that it is important for future works to build benchmarks with more complex object interactions, for instance with multiple action segments.



The knife is being used to cut a steak on a wooden cutting board.

The steak is being cut with a knife on a wooden cutting board.

A person is holding a steak with tongs on a wooden cutting board.

The **knife** is being held by a person next to a steak on a wooden cutting board.

The wooden chopping board holds a steak being cut with a knife and a pair of tongs.

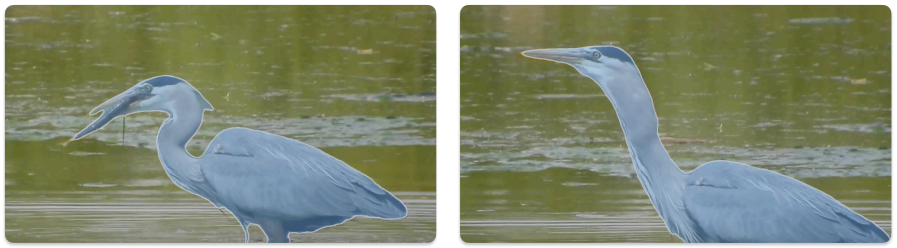
(i) Recognition error

A person is holding a roll of tape in front of them. The bracelet is worn on the wrist of a person who is holding a pink sphygmomanometer.

The yellow robe is being worn by a person who is holding a pink spinner. The earring is being help up by a person in front of a mirror.

The towel rack is mounted on a wall in a bathroom. A white bath towel hangs on the wall.

(ii) Inconsistent object categories



The gray heron is standing in the water with a fish in its beak.

(ii) Detection/segmentation error

Figure 8. Some DVOC failure cases of CaptionFormer observed on the LV-VIS dataset.

D. Additional details

D.1. Prompting strategy details

The bounding boxes used for the visual prompt are rendered with a very thin line (2px), which we observe is sufficiently thin to not impact recognition. Figure 9 shows that our prompting strategy effectively guides Gemini to attend to the image and focus on the object of interest, resulting in additional visual details being included in the generated captions. In the top example we can observe that Gemini refines the label ”person” to ”a person’s hand” and describes the interaction with the object and environment. In the bottom example the label ”gemstone” is refined by the color purple and the fact that it is covered in dirt.

The category labels given in the prompt are human-annotated ground truth. Nevertheless, we experimented with incorrect labels and Fig. 9 (red) shows that Gemini correctly identifies the resulting inconsistency.

a) Generated with ground truth category « person » :
A person's hand is reaching for and handling a binder clip while standing in front of a notebook.

b) Corrupted prompt « car » :
I'm sorry, but there is no object of class 'car' visible in the video.

c) Generated with ground truth category « gemstone »:
A purple gemstone covered in dirt is being held in someone's fingers.

d) Corrupted prompt « dog » :
I'm sorry, but I cannot fulfill the request. The object class is "dog" while the images clearly show a mineral being held by a person's hand. I'm unable to generate a relevant caption with such conflicting information.

Figure 9. **Qualitative examples** generated with our synthetic annotation approach using ground truth label and corrupted labels in the prompt.

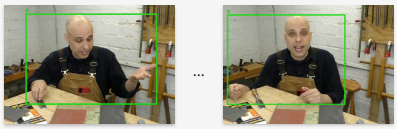
System Prompt p_s

```
"""
Generate a caption for a video, focusing on a queried object highlighted with green bounding boxes, and
semantic class provided.
It should be a rich sentence describing the object's APPEARANCE and ACTION, trajectory, or interaction with
other objects in the video frames. The other objects are not highlighted and should be mentioned only if
they interact with the query object or are relevant to the context.

# Rules
   - The single query object highlighted with bounding boxes in the frames SHOULD be the subject of the
   sentence. ex: for category "bottle": "The bottle is being inspected by a person"
   - Only facts that are visible in the video should be mentioned.
   - You should NOT mention the fact that the query object is highlighted with green bounding boxes.
   - You should RETURN A CAPTION no matter what, even if the query object is not visible in any frame.
   - If multiple objects of the same class as the query object are visible, the caption should focus
   exclusively on the single highlighted object and describe it as the singular subject of the sentence.
   - No foreign alphabets or special characters should be used in the caption. Translate foreign words if
   needed.

# Input Details
   - **Frames**: Provided sequence of 4 frames sampled from a video, in which a bounding-box highlights a
   query object
   - **Bounding Box**: Locations of the query object in the respective frames, in the format [(xmin, ymin,
   xmax, ymax),...] with each value ranging from 0 to 1000 representing a percentage of image dimensions.
   - **Area**: Area of the query object in the image, as a percentage of the total image area. This could
   help to determine wether the object is in the background. (big object class with small area)
   - **Semantic Class**: Class of the query object to be described in the caption
   - **Other classes**: Some classes of other objects in the video, These objects are not highlighted and
   should be mentioned only if relevant.

# Examples
   - **Input**: An image showing a women dressed in black and white and a dog both running on a beach with
   people in the background. The woman's short is highlighted with green bounding boxes in the video. object
   class: "short pants".
   - **Output**: "A black and white short pants is being worn by a woman running with a dog across the sandy
   beach"
"""
```

User Prompt $p_u(j)$

```
"""
Query object Bounding Box location in the respective
frames: {formatted_normalized_bbox}
Query object areas in percentage of the respective frames :
{formatted_area_pct}
Query object Class: '{cat["name"]}'
Other classes: {", ".join(other_cat_names)}
Generate a one-sentence caption, focusing on the object of
class '{cat["name"]}' highlighted by the bounding boxes.
"""
```

Visual prompt $\hat{x}_j$



$\hat{x}_j^{i_1}$ ... $\hat{x}_j^{i_4}$

Sampled frames with drawn bounding boxes $\mathcal{B}_j^{i_1}$ ... $\mathcal{B}_j^{i_4}$

Figure 10. Prompt template used to generate synthetic object captions from video segmentation annotations for the LV-VIS dataset [62]. The system prompt p_s contains general instructions, while user prompt and visual prompt $p_u(j)$ and $\hat{x}_j$ are formatted with information from each annotation.

The full prompt template used to generate our LVISCap and LV-VISCap datasets is illustrated in Figure 10. For a video x with N objects, we prompt the VLM N times, and for each object j the prompt is composed in three parts: (i) the static system prompt p_s gives general instructions for object-level caption generation, practical rules, prompting format and an example, (ii) the user prompt $p_u(j)$ depends on the example and contains textual annotation information to help the model describe objects and interactions accurately. These information notably contain target object positions, areas, category, and the categories of other objects in the scene, (iii) the visual prompt $\hat{x}_j$ consists of 4 sampled frames with drawn bounding boxes for object j.

D.2. Dataset details

LVIS [24] is a large-vocabulary image instance segmentation dataset with a long-tail distribution of 1203 annotated categories, for a total of over 2.2 million annotations in 164k natural images. The dataset is split in a training set with 100k images and 1.2M annotations, a validation set with 19k images and 244k annotations, and two test sets with 19k images each.

LV-VIS [62] is a recent large-vocabulary video instance segmentation (VIS) benchmark. It comprises $4,828$ videos with over 26k video segmentation masks from 1,196 object categories, with an average of over 5.4 objects per video. The data is split into a training set of $3,076$ videos and 16k video-level annotations, a validation set of 837 videos and 3.7k annotations, and a test set with 908 videos.

LVISCap and **LV-VISCap** denote our extensions of LVIS and LV-VIS (see Section 3.1). LVISCap extends LVIS with a total of $1,488,354$ synthetic captions, including $1,244,271$ training annotations and $244,083$ validation annotations. LV-VISCap includes a total of $19,717$ synthetic captions for $16,017$ training and $3,700$ validation annotations. Note that in the absence of annotations on the test sets of LVIS and LV-VIS, we only extend the training and validation sets with captions, and use the validation set for evaluation.

VidSTG[78] is a spatio-temporal video grounding dataset, containing $6,924$ videos with $44,808$ exhaustive trajectories annotations over 80 categories, as well as object sentence descriptions (for some objects and some timestamps only), which serve as queries for grounding. Zhou et al. [81] repurposed the dataset for DVOC by using queries as captions, and by excluding annotations without captions during evaluation. Following Zhou et al. [81], we sample 200 frames uniformly across each video for both training and evaluation. Overall, the repurposed VidSTG training set counts $28,275$ object trajectories, with $15,182$ object captions. The validation set, used for DVOC evaluation, includes 602 videos with $1,644$ captions.

Video Localized Narratives (VLN) extends existing datasets with "narrations of actors actions" in videos. We use the subset from the UVO dataset, which contains 3 sparsely annotated frames with non exhaustive captions for a total of $5,136$ training and $2,451$ validation videos.

BenSMOT contains manually collected annotations of bounding box trajectories and associated captions, focusing exclusively on humans in videos. It includes an average of 2.2 instances per video, and counts $2,284$ videos for training and $1,008$ for evaluation.

D.3. More Implementation details

The visual backbone is initialized with weights pretrained on ImageNet-21K [20] following OVFormer [21], and the Mask2Former [16] weights are trained from scratch. The OVFormer classifier uses a frozen CLIP ViT-B/32 [52] encoder. The captioning head is initialized with weights from BLIP-2 [39] with frozen LLM OPT-2.7B [76] following Chouduri et al. [18].

For all experiments except LV-VIS tuning, we first train the segmentation/detection model, then freeze it and tune the captioning head. Respectively for LVIS/VidSTG/LV-VIS we train for $440k/40k/22k$ for the first stage and $5k/2k/2k$ for the second stage. When tuning pretrained models on VidSTG/VLN/BenSMOT, we train the two stages for $(15k, 2k)/(15k, 500)/(15k, 2k)$ steps, whereas for LV-VIS, we end-to-end tune the model for $2k$ steps. Experiments are run with a batch size of 8, except when using LVIS+COCO and LV-VIS where we use a batch size of 4 and for video-level tuning of the captioning head where we use a batch-size of 1. Experiments on LV-VIS are end-to-end trainings with clip-level supervision only. For VidSTG/VLN/BenSMOT experiments we use video-level tuning for captioning with temporal aggregation, with $T_{\text{agg}} = 32/8/8$ respectively. For all experiments we train the model with a clips of size $T = 2$, and at inference use $T = 5/1/1/1$, $T_{\text{match}} = 1/100/20/40$, $K_{\text{match}} = 1/7/5/7$ for LV-VIS/VidSTG/VLN/BenSMOT experiments respectively. For the largest dataset (COCO + LVIS) the optimization takes 2 days on 4 H100 GPUs.

Following OVFormer [21], for all datasets we use an AdamW optimizer and the step learning rate schedule, with an initial learning rate of 0.0001 and a weight decay of 0.05, and apply a 0.1 learning rate multiplier to the backbone. We decay the learning rate at 0.9 and 0.95 fractions of the total number of training steps by a factor of 10. For respectively image/video datasets, we resize the shortest edge of the image to 800/480 for SwinB and 800/360 for ResNet for training and inference.