Supplementary material

To complement the main paper, this supplementary material assembles additional results, analyses, and implementation details. Sections A and B provide additional qualitative visualizations and expanded quantitative metrics, respectively. Section C discusses the model and data limitations. Section D reports comprehensive dataset statistics. Section E details the GROVE architecture and training setup, while Section F describes our automatic annotation pipeline. Section G outlines the human-annotation protocol, and Section H lists the exact prompts used to curate spatio-temporally grounded captions.

A. Additional qualitative results

Figures 8 and 9 show qualitative results of our GROVE model (Section 4 in the main paper), pre-trained on the HowToGround1M dataset and finetuned on the iGround training set (2013 examples). The results are shown on the iGround test set. In the figures' captions we discuss some of the benefits of our model. Additional qualitative results showcasing the predictions of our approach overlaid over the input videos are shown in the **supplementary video** (available at https://ekazakos.github.io/grounded_video_caption_generation/). Figure 10 shows the main failure modes of our model.

B. Additional quantitative results

Detailed analysis for the ablations of automatic annotation. We replace each stage of our automatic annotation method with an alternative. Results are shown in Table 7 in the main paper. In Stage 1, we replace the still-image model [32] with an alternative still-image grounded caption generation method. This approach leverages GIT [42] for frame-level captioning, Llama3 [9] for extracting noun phrases from the caption, and OWLv2 [25] for their bounding box localisation within each frame. We call this alternative "b. Alt. Stage 1 (F)". We also evaluate a video-level variant "c. Alt. Stage 1 (V)", where we replace the GIT captioner with VideoLlama3 [46]. To ablate Stage 2 ("d. Alt. Stage 2"), we provide the LLM with full captions from Stage 1 instead of extracting Subject-Verb-Object triplets from the caption to assess the impact of additional context. To ablate Stage 3 ("e. Alt. Stage 3"), we incorporate CoTracker3 [15], a SOTA visual point tracking method to provide temporal association of bounding boxes across frames. Using 5 uniformly sampled frames and their bounding box predictions from Stage 1, we track objects in between with CoTracker3 and associate the resulting tracks with noun phrases from the caption.

Results are reported in Table 7 in the main paper, where we compare the alternative automatic annotation methods on the iGround validation set. The frame-level alternative Stage 1 (row b.) performs better in captioning due to GIT's superior performance but performs noticeably worse for grounding. This is because our Stage 1 still-image grounding model [32] is explicitly trained for grounding, unlike GIT, Llama3, and OWLv2, which are not trained jointly and may underperform due to various factors–such as Llama3 extracting non-groundable noun phrases or OWLv2 missing objects. A similar trend is observed for the video-level alternative Stage 1 (row c.). Compared to our proposed method, the alternative Stage 2 (row d.) underperforms across all metrics except AP50. This is because the full-caption input yields fewer predictions as the LLM trims its output to the most salient objects, reducing recall but improving precision–hence the slightly higher AP50. In contrast, the SVO-based input in our proposed automatic annotation method leads to slightly longer captions (12 vs.11 words) with more noun phrases (3.3 vs. 3.0), leading to more object predictions and higher recall. This reflects a typical precision-recall trade-off. The alternative Stage 3 (row e.) underperforms in grounding due to tracker drift caused by abrupt viewpoint changes. Overall, on average (the last column in Table 7 in the main paper), our proposed method achieves the best performance.

Comparison with the state of the art on YouCook-Interactions and GroundgYouTube datasets. In Table 8, we evaluate GROVE on YouCookInteractions and GroundingYouTube datasets, outperforming the previous SOTA by large margins.

Method	YouCook-Interactions	GroundingYouTube
What When and Where (S3D) [4]	53.98	60.62
What When and Where (CLIP) [4]	58.35	56.98
GROVE	**68.67**	**72.14**

Table 8. Comparison with SOTA on YouCook-Interactions [38] and GroundgYouTube[4] datasets.

C. Limitations

Although our proposed datasets and model advance the state of the art in grounded video captioning and spatio-temporal sentence grounding, they also reveal avenues for future exploration, stemming from the following limitations.

Scaling to long videos. Despite achieving state-of-the-art results on VidSTG by running inference in a sliding-window manner over videos up to three minutes, the training phase remains memory-bound: we can supply the model with only eight frames per clip. This is sufficient for the short clips in HowToGround1M and iGround (8-10 seconds), where eight frames corresponds to about 1 fps sampling. For VidSTG's much longer videos, however, uniform sampling of only eight frames introduces large gaps between frames and prevents the model from seeing fine-grained temporal dependencies during training. Closing this discrepancy will require methods that can train directly on larger frame spans or more

	Method	METEOR	CIDER	AP50	Recall
All	Auto. annotation	12.3	31.7	26.9	19.3
	GROVE	**19.7**	**92.6**	**42.0**	**26.9**
Hard	Auto. annotation	08.1	07.3	22.3	14.7
	GROVE	**14.4**	**41.3**	**36.0**	**18.9**

Table 9. Results of our automatic annotation method (Auto. annotation) and the complete proposed model (GROVE) on the entire iGround validation set (All) and for a subset (about 10% of data) with challenging similar referring expressions (Hard).

efficient representations of extended temporal context (*e.g.* memory).

Complex referring expressions. We examined the iGround validation set and discovered that roughly 10% of its videos contain more than one object whose referring expressions (and appearance) are highly similar. We designate this challenging portion of the data as the "Hard" subset. In Table 9, we compare GROVE with our automatic annotation method in this subset. Although GROVE still surpasses the automatic annotation method on this subset, the marked drop in performance of both methods on the "Hard" subset (comparing to "All", *i.e.* the full validation set) reveals that reliably disambiguating closely related referring expressions remains still a challenge. These results suggest opportunities to refine both the model architecture and the automatic annotation method to better handle such fine-grained cases.

D. Dataset Statistics

Table 10 reports the statistics of both the HowToGround1M pre-training dataset and the iGround manually annotated set. Word clouds of the natural language descriptions from those datasets are shown in Figure 7.

Statistic	HowToGround1M	iGround
Avg num frames per video	44.6	40.1
Avg duration (seconds)	7.9	8.0
Avg num instances per video	80.1	118.1
Total num instances	80,092,775	421,588
Avg box width × height	243.7 × 172.6	174.9 × 135.5
Avg tube length (frames)	6.4	29.0
Avg caption length (words)	12.1	15.4

Table 10. Statistics of HowToGround1M and iGround datasets.

E. Details of the GROVE model

Model architecture. Figure 3 in the main paper shows the different components of our approach. The Global Video Encoder, $\mathcal{V}_e(\cdot)$, outputs video features, o_e, which are pooled spatio-temporally, resulting in the video prompts. These are projected to a language embedding space with $VL(\cdot)$. The LLM, $\mathcal{LM}(\cdot)$, ingests a multimodal prompt consisting of

Figure 7. Word cloud for (a) HowToGround1M dataset and (b) iGround dataset.

video and language tokens. The LLM is prompted to generate a caption for the video by tagging the noun phrases that correspond to objects and appending them with detection tokens (shown with red and green in the LLM's generated caption in Figure 3 The LLM's output hidden states that correspond to the generated caption are projected to queries (using $LQ(\cdot)$). The queries corresponding to the detection tokens are fed to the bounding box decoder $D(\cdot)$. The Grounding Video Encoder, $\mathcal{V}_g(\cdot)$, outputs fine-grained video features, which are also fed to the decoder. The decoder performs cross-attention frame-wise between the queries and the outputs of $\mathcal{V}_g(\cdot)$, o_g, which are used as keys/values. Finally, the prediction heads output bounding box predictions and temporal objectness scores for each object at each frame. This objectness score is used to predict the presence/absence of the object in each video frame and is of major importance for the grounded video caption generation task. Details about the visual backbones $\mathcal{V}_e(\cdot)$ and $\mathcal{V}_g(\cdot)$ as well as details about the LLM $\mathcal{LM}(\cdot)$ including the format of its multimodal inputs and its vocabulary are given next.

Projection layers. We project the outputs of the Global Video Encoder and the output hidden states of the LLM with MLPs, $o_{p'} = VL(o_p)$ and $o_q = LQ(o_l)$, where $VL(\cdot)$ projects the visual features to an embedded language space, while $LQ(\cdot)$ projects the LLM's hidden states to queries. $o_{p'}$ is the LLM's visual input while o_q is input to the bounding box decoder that is described next.

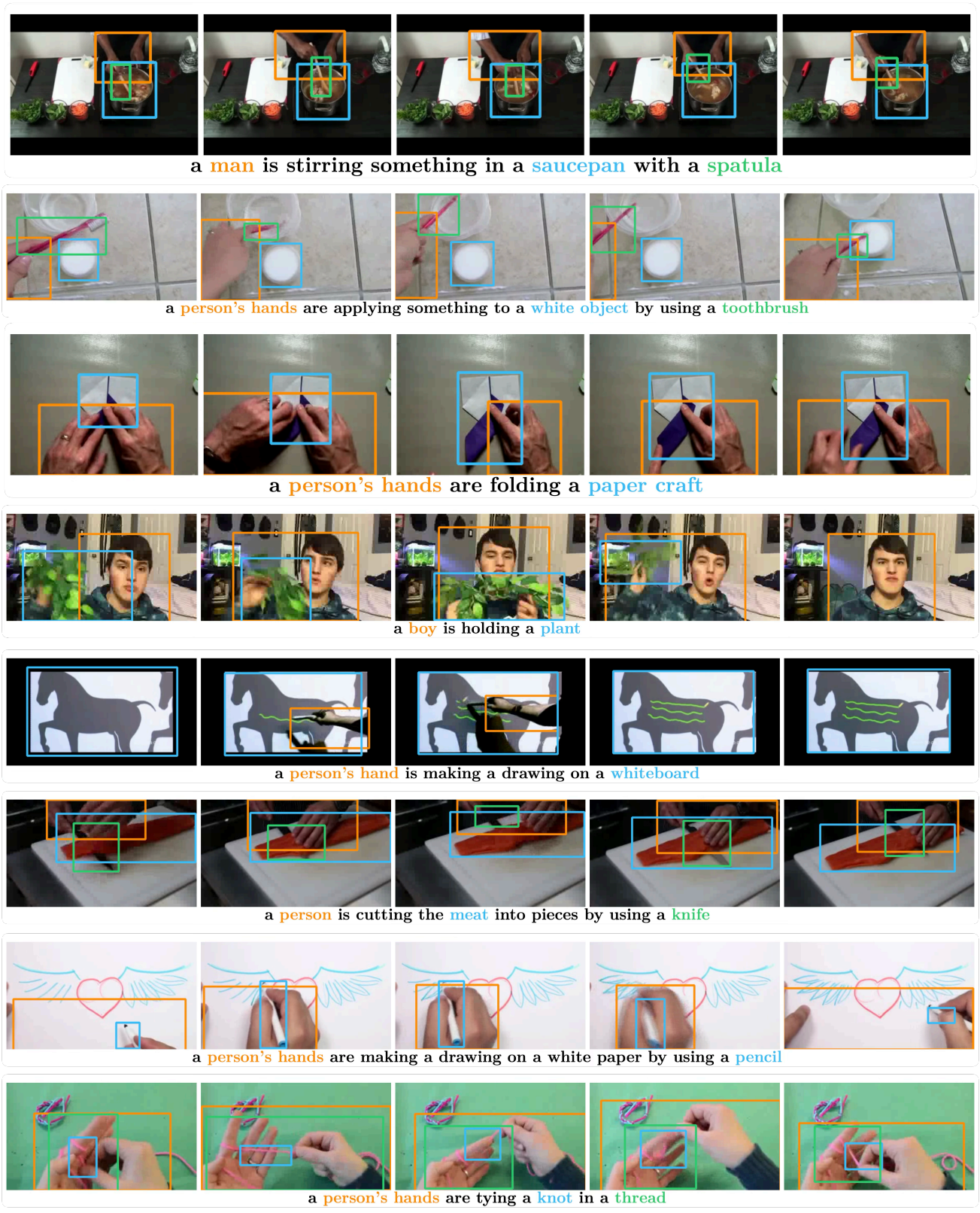

Figure 8. Qualitative results of our GROVE model on the (unseen) iGround test set. The colour-coded sentence fragments are spatio-temporally localised in the video with the bounding boxes colour coded with the same colour. The results demonstrate that: (i) our model can localise even small objects such as a pen or a tooth brush; (ii) objects are consistently labelled across frames despite changes of viewpoint or scale; (iii) the model focuses on the human and the interacted objects; (iv) the model can successfully ground multiple objects in the video. **Additional results are shown in the supplementary video** (available at https://ekazakos.github.io/grounded_video_caption_generation/).

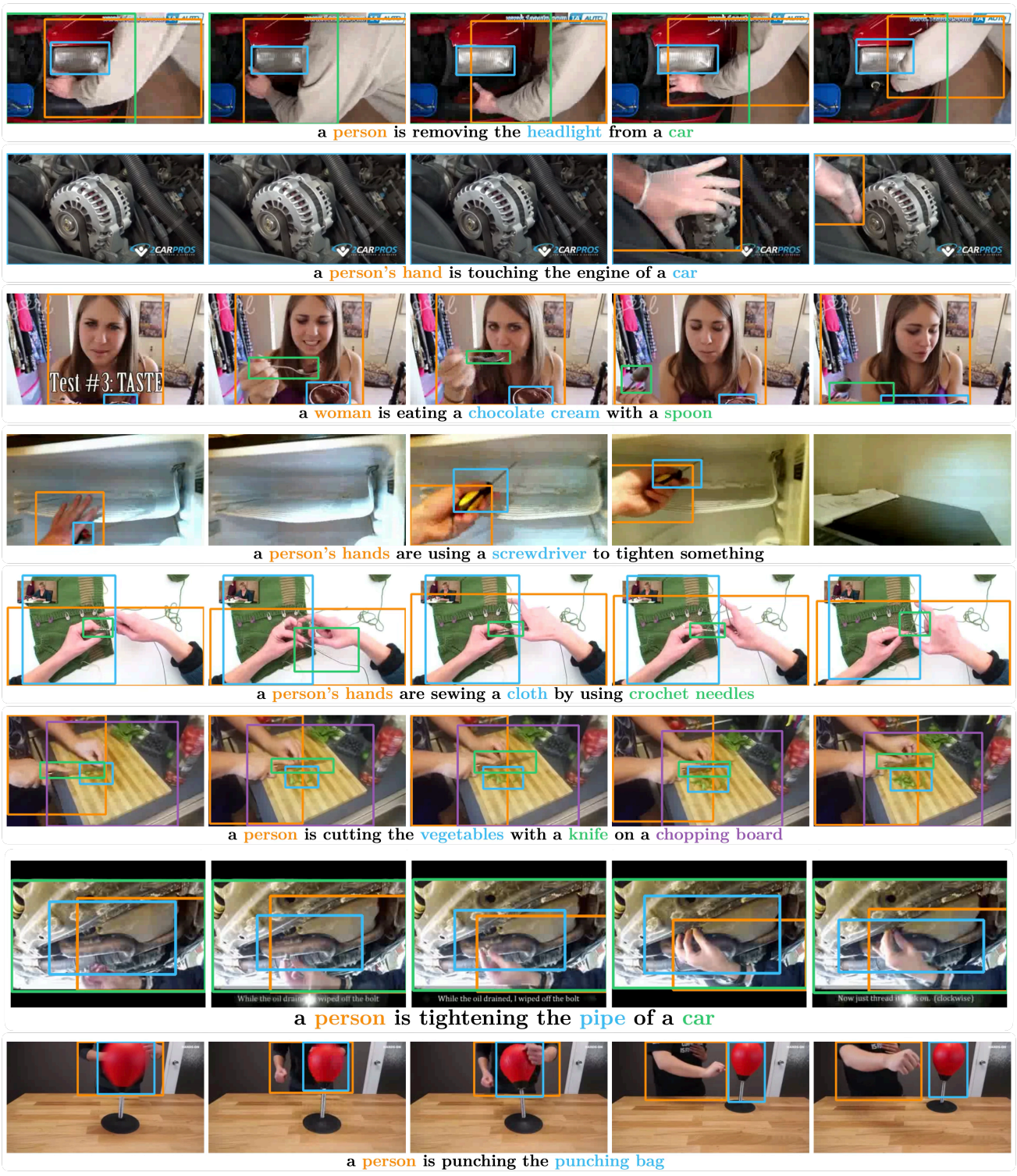

Figure 9. Additional qualitative results of our GROVE model on the (unseen) iGround test set. The colour-coded sentence fragments are spatio-temporally localised in the video with the bounding boxes colour coded with the same colour. In addition to the model's properties discussed in Fig. 8, GROVE is capable of predicting whether an object is present in a certain frame via the temporal objectness head; in the second example there are no bounding box predictions for the hand in the first three frames while in the fourth example there are no predictions for the hand and the screwdriver in the second and fifth frame. **Additional results are shown in the supplementary video** (available at https://ekazakos.github.io/grounded_video_caption_generation/).

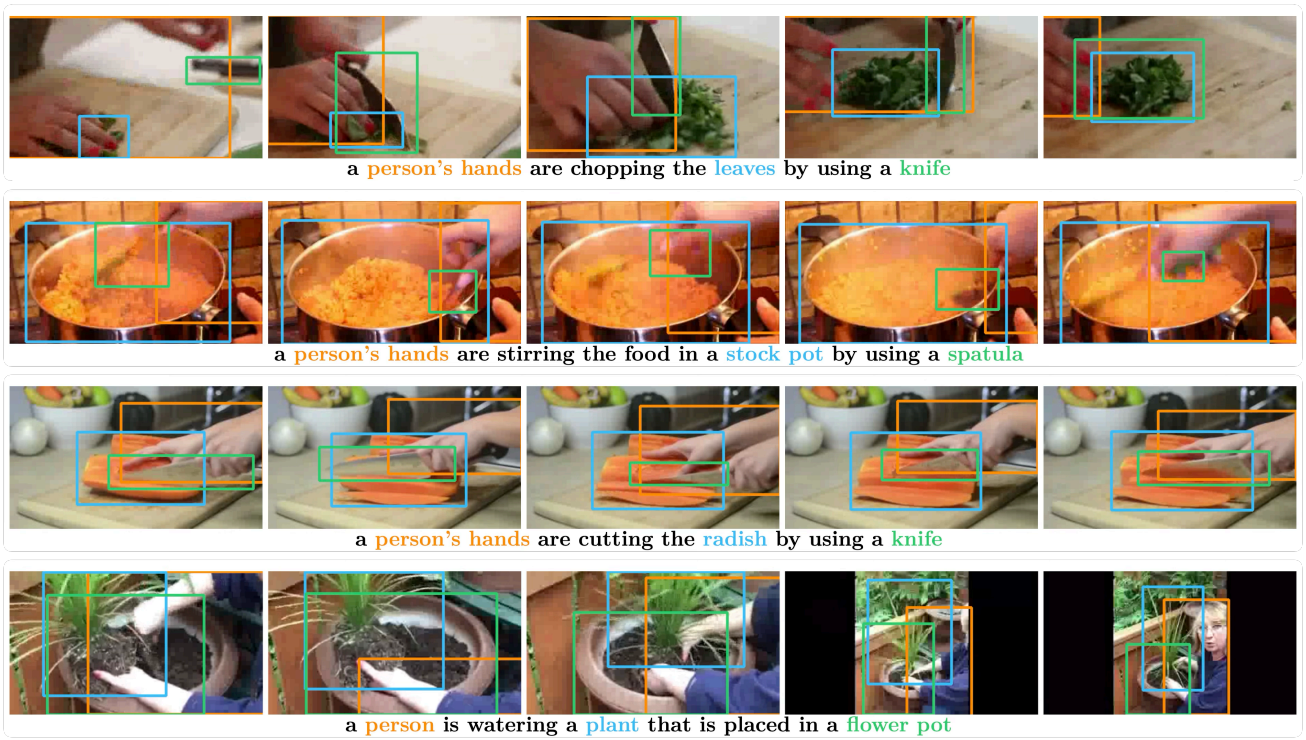

Figure 10. **Qualitative results for the main failure modes of our GROVE model on the (unseen) iGround test set.** The colour-coded sentence fragments are spatio-temporally localised in the video with the bounding boxes colour coded with the same colour. We identify four main failure modes: (i) temporal objectness mispredicts the presence of an object (first row, last frame for the knife), (ii) inaccurate predictions of object location (second row, third and last frames for the spatula), (iii) misclassification of object (third row, model predicts "radish" for the pumpkin), and (iv) misclassification of action (last example, model predicts "watering" for planting).

Backbones. GROVE consists of two video encoders and a multimodal LLM as its main backbones. The Global Video Encoder $\mathcal{V}_e(\cdot)$, takes as input a video $v \in \mathbb{R}^{T\times H1\times W1}$ and produces an output $o_e \in \mathbb{R}^{T\times \frac{H1}{p}\times \frac{W1}{p}}$, where p is the patch size of the underlying visual transformer. Its purpose is to provide a holistic representation of the video that will be ingested by the LLM. The Grounding Video Encoder $\mathcal{V}_g(\cdot)$, takes as input a video $v \in \mathbb{R}^{T\times H2\times W2}$, where $W2 > W1$ and $H2 > H1$. It produces $o_g \in \mathbb{R}^{T\times \frac{H2}{p}\times \frac{W2}{p}}$. o_g is used to ground phrases from the caption to the visual content, which is performed by the bounding box decoder that is described later. The input video to the Grounding Video Encoder is of larger spatial resolution than that of the Global Video Encoder for enhanced localisation capability. Finally, the LLM $\mathcal{LM}(\cdot)$ takes as input a multimodal sequence $s \in \mathbb{R}^{L\times D}$ and produces an output o_l of the same size. Its input is of the form `The <video> provides an overview of the video. Could you please give me a description of the video? Please respond with interleaved bounding boxes for the corresponding parts of the answer.` `<video>` is replaced by the output of $\mathcal{V}_e(\cdot)$, and therefore the LLM ingests mixed language and visual tokens. We also augment the LLM's vocabulary with a detection token `<DET>`, prompting the model to generate responses with `<DET>` tokens by the phrases that correspond to objects to be detected in the video.

Loss function. Our loss function is a combination of a language modelling loss and losses relevant to video object detection. The language modelling loss is a Cross-Entropy loss applied on o_l. For object detection, we follow DETR [3] and use a gIoU loss [34] and an L1 loss applied on p_{bb}. Different than [3], the losses are applied per frame and summed over frames. Moreover, the losses are applied only to the objects that appear in the frame (rather than each object in the caption) using the ground-truth temporal objectness scores. The representation that we use for the bounding boxes is `[x,y,w,h]` and their coordinates are normalised with the dimensions of the video. Finally, we employ a binary cross-entropy loss on p_{tobj}. Our loss is, hence, defined

as:

$$\mathcal{L}_{LM} = CE(o_l) \quad (1)$$
$$\mathcal{L}_{gIoU} = gIoU(p_{bb}, gt_{bb}) \quad (2)$$
$$\mathcal{L}_{L1} = L1(p_{bb}, gt_{bb}) \quad (3)$$
$$\mathcal{L}_{tobj} = BCE(p_{tobj}, gt_{tobj}) \quad (4)$$
$$\mathcal{L} = \lambda_{LM} \times \mathcal{L}_{LM} + \lambda_{gIoU} \times \mathcal{L}_{gIoU} \quad (5)$$
$$+ \lambda_{L1} \times \mathcal{L}_{L1} + \lambda_{tobj} \times \mathcal{L}_{tobj}, \quad (6)$$

where gt_{bb} are the ground truth boxes and gt_{tobj} are the ground truth objectness scores and λ are the weights for the losses.

Training/inference. We realise the Global Video Encoder $\mathcal{V}_e(\cdot)$ with a CLIP-L [31] model with an input of 336×336 and a patch size of 14. The Grounding Video Encoder $\mathcal{V}_g(\cdot)$ is instantiated with a SAM [17] encoder and the bounding box decoder $D(\cdot)$ is a SAM-based decoder, the same as in GLaMM [32]. The LLM $\mathcal{LM}(\cdot)$ is a Vicuna-7B model [7]. During training we keep $\mathcal{V}_e(\cdot)$, $\mathcal{V}_g(\cdot)$ and $\mathcal{LM}(\cdot)$ frozen. $\mathcal{V}_g(\cdot)$ originally takes as input 1024× 1024 images. As this is too large to fit in memory for videos, we instead use 512×512 video spatial resolution, while we interpolate the positional encodings of $\mathcal{V}_g(\cdot)$ and fine-tune them. Adapters are 3D spatiotemporal convolutional layers with a kernel of size $3 \times 3 \times 3$ and a stride of 1. We apply adapters to every 3 layers of $\mathcal{V}_e(\cdot)$ and to all global attention layers of $\mathcal{V}_g(\cdot)$. The bounding box head h_{bb} is an MLP with two FC layers and a ReLU activation function in between, while the temporal objectness head h_{tobj} is a linear layer. Both prediction heads employ a sigmoid activation function. We apply a threshold of 0.5 to the temporal objectness scores. Both the adapters and the prediction heads are randomly initialised. We use $T = 8$ frames for the videos during both training and testing. During training we perform random sparse sampling of frames by splitting the video in 8 segments and randomly drawing a frame from each segment while during testing we pick the centre frame of each segment.

We train GROVE for 20 epochs using a batch size of 128. We use a learning rate of 5×10^{-5} with warmup for the first 100 training steps and linearly decay the learning rate for the rest of training. We do not apply any weight decay or spatial data augmentation. We use $\lambda_{LM} = 1$,$\lambda_{gIoU} = \lambda_{L1} = \lambda_{tobj} = 2$.

Details of VidSTG and ActivityNet-Entities experiments. For VidSTG [49] and ActivityNet-Entities [51], we do not use the temporal objectness head. That is because in VidSTG the spatio-temporal tubes are continuous within the segments' boundaries, while ActivityNet-Entities provides annotations for a single frame per object and in the rest of the frames the objects might still be present but without annotation, and thus should not be modelled as absent. As the task in VidSTG entails predicting the spatio-temporal bounding boxes *given* a short description, we provide the short descriptions as input to our GROVE model during both training and inference in a teacher-forcing setup. For evaluating GROVE on VidSTG *without* observing any VidSTG data during training (GROVE with FT: ✗, Table 4 in the main paper), we pre-train GROVE on HowToGround1M. Each HowToGround1M caption is rewritten–by prompting Llama-3–into both of VidSTG's sentence styles, declarative and interrogative. Every transformed sentence is then paired with a single bounding box per frame, chosen as the box of the first subject or object it mentions. This supervision reshapes HowToGround1M's annotation distribution to mirror VidSTG's, allowing GROVE to achieve strong performance without relying on any VidSTG training data.

F. Details of the automatic annotation method

Multiple people in the video. Our automatic annotation method can handle multiple subjects in a video as long as one of the two following conditions are met: a) the subjects are described with a distinct language, *e.g.* 'man with green jumper' and 'man with blue shirt', or b) the subjects are within a Subject-Verb-Object relationship even when described with the same terms, *e.g.* ('person', 'dances', 'with', 'person') which would produce 'A person dances with another person'. If neither conditions are met, the caption aggregation (Stage 2) may merge the two subjects into one.

Association of verbs and objects is naturally performed through the Subject-Verb-Object triplets. For example, given two relationships: ('man', 'cuts', 'onions') and ('woman', 'stirs', 'food', 'in', 'pot'). The LLM-based caption aggregation step (Stage 2) has sufficient information to associate the man with the action of cutting the onions and the woman with stirring the food.

Additional details of Stage 3. We provide additional details of the procedure of Stage 3 using the example from Figure 2 in the main paper, right. The object in the woman's hands is described as 'a green beverage' and 'a glass of green liquid' across different frames. Stage 2 has provided the video-level noun phrases 'a woman' and 'a beverage'. Stage 3 is formulated as a classification problem where each one of 'a green beverage' and 'a glass of green liquid' are the inputs to be classified in one of the classes {'a woman', 'a beverage', $\emptyset$} and thus associated with the right bounding box. The class $\emptyset$ represents the "None" class, *i.e.* when an input does not belong to any of the known classes and it is useful for noisy inputs.

G. Protocol for human annotations

In Figure 11, we describe the annotation guidelines for annotating the training/validation/test sets of the GROVE dataset.

The annotation criteria have been extensively discussed

Annotation Guidelines:

1. Video Selection:
   - You will be provided with a larger set of videos than needed.
   - Your first task is to select clips that are considered 'interesting' based on criteria that will be discussed further. An 'interesting' video typically includes dynamic events or actions that are clear and distinguishable despite the low video quality. In those events/actions people usually interact with objects, e.g. 'A man is cutting an onion using a knife'. 'Non-interesting' events are typically static, e.g. a person simply standing/sitting and talking. Non-interesting events are also events with ambiguous actions taking place, i.e. generic/abstract actions that cannot be described concisely or actions for which the annotator is unsure about what is happening in the video.
2. Video Annotation:
   - For each selected video clip, write a concise, one-sentence description of the main event taking place in the clip. If the action is too complex, use at most two sentences for describing it, but prioritise one-sentence descriptions.
   - Focus only on the objects that humans interact with rather than describing densely every object in the scene.
   - To enrich the language descriptions, also describe properties of objects such as color, shape, etc, e.g. 'blue cup' or 'red onion'. It is not strictly necessary to always describe the object's property but only when deemed important by the annotator.
   - When you are unsure about the object being used, you can simply describe it as 'object'. If object is unknown but the category of the object is known, please describe the object using its category, e.g. 'food'.
   - When there are two or more humans in the scene, use one of their characteristics to distinguish them, e.g. 'the woman in the red shirt standing next to the woman in the green shirt is putting a strawberry on a cocktail glass'.
   - If there are multiple actions happening consecutively, describe all of them and their associated objects. E.g. 'a person is doing action-1 using object-1, then doing action-2 with an object-2'. As shown in the example, you can use 'then' for connecting temporally adjacent actions.
   - Provide bounding boxes for humans/different objects mentioned in your description. These bounding boxes should be applied to all frames where the objects are visible.
   - Label each bounding box with a short phrase directly from your sentence description (e.g., 'a brown dog', 'personś hands').
   - It is not necessary that each object appears in each frame of the video. For example, a person might be using a tool, then leaving it down and using another tool. In this case, you would annotate with bounding boxes the first tool for the first half of the video and the second tool for the second half. Another common case is that objects or the person might disappear and then reappear. In this case, again all instances of the object must be annotated, so you should be careful about objects leaving the scene as they might enter the scene again later.
   - If there are many small objects, e.g. mushrooms in a pan, use a single bounding box labelled as 'mushrooms'.
   - There are cases where two or more bounding boxes are needed for objects of the same type: a) one bounding box for each human hand when both are used to perform an action, b) one bounding box for each tool/container/appliance etc of the same type that the human is using, e.g. when they are placing food in two dishes, or pouring the content of a shaker in two cocktail glasses.
   - Descriptions: Must be accurate and written in fluent English. Suitable for either native speakers or highly proficient English speakers.
   - Bounding Boxes: Ensure that bounding boxes accurately encompass the objects for the entirety of their visibility within the clip. The bounding boxes should be consistent and smooth across frames, maintaining size and position as closely as possible given the movement of the object and video quality. An exception is when there are abrupt viewpoint changes of the camera, which might result in objects abruptly changing position and size across neighbouring frames.

Figure 11. Annotation guidelines for the manually annotated iGround dataset.

with the annotation provider and the annotators have been trained based on those criteria prior to commencing the annotation process. We have also performed a pilot annotation project with the annotation provider on 10 video clips with several rounds of careful checking and feedback. Moreover, the annotation provider performed regular quality reviews on the annotations to ensure that the annotation criteria have been met.

H. Prompts for automatic curation of spatio-temporally grounded captions

The full prompt for the **Stage 2 (Video-level caption aggregation)** of our automatic annotation approach (Section 3 in the main paper) is shown in Figure 12 and the full prompt for **Stage 3 (Temporally consistent bounding box annotation)** in Figure 13.

System Instructions

Generate a dynamic, video-level description based on frame-level inputs. The inputs include actions performed in individual frames in the form of Subject-Verb-Object (SVO) triplets along with prepositions and prepositional objects. The SVO triplets describe how actions are performed and how objects interact. Your output should be a concise narrative in 1 sentence, focusing on the most salient actions depicted across the frames. Enclose the exact text of relevant objects within tags.
Input format:

```
[['subject': 'subject_text', 'verb': 'action_text', 'object': 'object_text',
'prepositions_objects': [('preposition', 'prepositional_object')],],]
```

Output format:

```
A Python dictionary with a key 'CAPTION', and as a value a dynamic description of the video content.
```

Infer motion from static descriptions. E.g. 'image shows a person holding a spoon and a bowl' implies 'person is stirring food in a bowl'. Enclose the human and the most frequent object that is used to perform the action within tags. If there is no human, enclose the two most frequent objects within tags.

User Input 1

SVO:

```
[['image', 'shows', 'cup'], ['bowl', 'is']],
[['person', 'holding', 'spoon'], ['spoon', 'is', 'bowl'],
[['image', 'shows', 'spoon', ('inside', 'bowl')]],
[['person', 'seen'], ['person', 'holding', 'spoon'], ['spoon', 'used'],
 ['spoon', 'stir', 'food', ('in', 'bowl')]],
[['person', 'holding', 'spoon'], ['spoon', 'is', 'bowl']],
[['person', 'holding', 'spoon'], ['spoon', 'is', 'bowl']],
[['person', 'holding', 'spoon'], ['spoon', 'is', 'bowl']],
['image', 'shows', 'spoon', ('in', 'bowl')]],
[['image', 'shows', 'bottle'], ['bottle', 'positioned', ('beside', 'bowl')]],
[['image', 'shows', 'bottle'], ['bottle', 'positioned', ('beside', 'cup')]],
[['image', 'shows', 'bottle'], ['image', 'placed', ('on', 'counter')],
 ['bottle', 'positioned', ('beside', 'bowl')]]]
```

Assistant Response 1

```
{'CAPTION': '<p>A person</p> is stirring <p>food in a bowl</p> using a spoon'}
```

User Input 2

SVO:

```
[['hand', 'using', 'cutting board']],
[['woman', 'using', 'cutting board'], ['woman', 'make', 'craft project']],
[['child', 'using', 'craft cutter'], ['child', 'cut', 'object']],
[['child', 'using', 'craft cutter'], ['child', 'cut', 'paper']],
[['woman', 'using', 'craft cutter'], ['woman', 'cut', 'object']],
[['woman', 'using', 'scissors pair'], ['woman', 'cut', 'piece', ('of', 'paper')]],
[['hand', 'using', 'scissors pair'], ['hand', 'cut', 'piece', ('of', 'paper')]],
[['woman', 'using', 'scissors pair'], ['woman', 'cut', 'piece', ('of', 'paper')]],
[['woman', 'using', 'craft cutter'], ['woman', 'cut', 'object']],
[['woman', 'using', 'craft cutter'], ['woman', 'cut', 'plate']]]
```

Assistant Response 2

```
{'CAPTION': '<p>A woman</p> is cutting <p>an object</p> using a craft cutter'}
```

New User Input

SVO: {input_svo}

Figure 12. The full prompt for Stage 2 (Video-level caption aggregation) of our automatic annotation approach (Section 3 in the main paper).

System Instructions

You are tasked with classifying humans and objects to a set of given categories.
Input format:

```
Human/Object (string), set of categories (lists of strings).
```

Output format:

```
A Python dictionary with a key 'CATEGORY', and as a value the predicted category of the human/object.
```

Use 'None' if the human/object doesn't belong to any of the categories. DO NEVER classify a human as the object category and vice versa.

User Input 1

Input: 'person'
Categories: ['a woman', 'her hair']

Assistant Response 1

```
{'CATEGORY': 'a woman'}
```

User Input 2

Input: 'table'
Categories: ['a person', 'a bowl']

Assistant Response 2

```
{'CATEGORY': 'None'}
```

User Input 3

Input: 'a piece of food on a plate'
Categories: ['a woman', 'a meal']

Assistant Response 3

```
{'CATEGORY': 'a meal'}
```

User Input 4

Input: 'a hand'
Categories: ['a person', 'food on a plate']

Assistant Response 4

```
{'CATEGORY': 'a person'}
```

User Input 5

Input: 'a man in a white shirt and black apron is also present'
Categories: ['a person', 'food']

Assistant Response 5

```
{'CATEGORY': 'a person'}
```

New User Input

Input: `{input_object}`
Categories: `{input_categories}`

Figure 13. The full prompt for Stage 3 (Temporally consistent bounding box annotation) of our automatic annotation approach (Section 3 in the main paper).