A. Appendix / Supplemental Material

A.1. Qualitative Results
A.1.2 Strefer-Synthesized Data
A.1.2 Strefer-Trained Model
A.2. Strefer Details
A.3. Model Details
A.3.1 Architecture Overview
A.3.2 Video Token Representation
A.3.3 Masklet Reference Token Representation
A.3.4 Timestamp Reference Token Representation
A.4. Implementation Details
A.5. Evaluation Details
A.6. Training-Free Methods Experimental Results
A.7. Limitations and Future Directions

A.1. Qualitative Results

We provide qualitative results that effectively showcase the strengths and limitations of our method and model.

A.1.1. Qualitative Results of Strefer-Synthesized Data

We present qualitative results of our **Strefer**-synthesized data in Fig. 6 and Fig. 7, as well as qualitative results of our novel referring masklet generation pipeline within **Strefer** in Fig. 8, Fig. 9, Fig. 10 and Fig. 11. Observations of the qualitative results are presented in the figure captions; more discussions are available in Sec. 4.3 of the main paper and Appendix A.7.

A.1.2. Qualitative Results of Strefer-Trained Model

We present qualitative results of our final **Strefer**-trained model in comparison to the 'Baseline' model (from our quantitative result tables). Specifically, results are shown in Fig. 12 and Fig. 13 for the task of **Video Regional Captioning/Description**; Fig. 14, Fig. 15, Fig. 16, Fig. 17 and Fig. 18 for the task of **Video Regional QA**; and Fig. 19, Fig. 20 and Fig. 21 for the task of **Timestamp-Referred Video QA**. Three failure cases are also shown in Fig. 22, Fig. 23 and Fig. 24. Observations of the qualitative results are presented in the figure captions; more discussions are available in Sec. 4.3 of the main paper and Appendix A.7.

A.2. Strefer Details

Notably, in the design of **Strefer**, we choose masks to accommodate diverse, free-form spatial references from users (e.g., points, scribbles, etc.), which can be readily converted into masks using off-the-shelf tools like SAM2 [43].

We present our designed Referring Masklet Generation Pipeline in Algorithm 1 and the Video Clipper in Algorithm 2. On the right, we list the prompt used for the Video LLM-based Active Entity Recognizer.

Algorithm 1 Referring Masklet Generation Pipeline

Require: Video $\mathcal{V}$, Referring Expressions $\mathcal{R} = [r_1, \ldots, r_n]$, Generalized Nouns $\mathcal{G} = [g_1, \ldots, g_n]$
Ensure: Masklets aligned to each referring expression $r_i \in \mathcal{R}$

procedure GENERATEMASKLETS($\mathcal{V}, \mathcal{R}, \mathcal{G}$)
  $\mathcal{F}, S \leftarrow$ SAMPLEANDREORDERFRAMES($\mathcal{V}$)
  $f^*, D_{f^*} \leftarrow$ SELECTINITIALFRAME($S, \mathcal{G}, \mathcal{R}$)
  $\mathcal{M} \leftarrow$ BIDIRECTIONALSEGMENTATIONTRACKING($\mathcal{F}, f^*, D_{f^*}$)
  $\mathcal{M} \leftarrow$ ASSIGNEXPRESSIONSTOMASKLETS($\mathcal{M}, f^*, \mathcal{R}, \mathcal{G}$)
  return $\mathcal{M}$
end procedure

procedure SAMPLEANDREORDERFRAMES($\mathcal{V}$)
  $\mathcal{F} \leftarrow$ sample frames from $\mathcal{V}$
  $S \leftarrow$ reorder $\mathcal{F}$ using a middle-first recursive strategy
  return $(\mathcal{F}, S)$
end procedure

procedure SELECTINITIALFRAME($S, \mathcal{G}, \mathcal{R}$)
  max_count $\leftarrow -1$, $f^* \leftarrow S[-1]$, best_frame $\leftarrow S[0]$, best_detections $\leftarrow \emptyset$
  for each frame $f \in S$ **do**
    $D_f \leftarrow$ GROUNDINGDINO(f, `set`($\mathcal{G}$))
    if $|D_f| >$ max_count **then**
      max_count $\leftarrow |D_f|$, best_frame $\leftarrow f$, best_detections $\leftarrow D_f$
    end if
    if $|D_f| \geq |\mathcal{R}|$ **then**
      return (f, D_f)
    end if
  end for
  if $f^* == S[-1]$ **and** best_frame $\neq f^*$ **then**
    $f^* \leftarrow$ best_frame, $D_{f^*} \leftarrow$ best_detections
  end if
  return (f^*, D_{f^*})
end procedure

procedure BIDIRECTIONALSEGMENTATIONTRACKING($\mathcal{F}, f^*, D_{f^*}$)
  Define forward sequence: $\mathcal{F}^{\rightarrow} = [f^*, f^*{+}1, \ldots]$
  Define backward sequence: $\mathcal{F}^{\leftarrow} = [f^*, f^*{-}1, \ldots]$
  Initialize $\mathcal{T} \leftarrow []$
  for each clip $\mathcal{C} \in \{\mathcal{F}^{\rightarrow}, \mathcal{F}^{\leftarrow}\}$ **do**
    $\mathcal{M}_{\mathcal{C}} \leftarrow$ SAM2($\mathcal{C}, D_{f^*}$, `video`)
    Append $\mathcal{M}_{\mathcal{C}}$ to $\mathcal{T}$
  end for
  return MERGE_TRACKING_RESULTS($\mathcal{T}$)
end procedure

procedure ASSIGNEXPRESSIONSTOMASKLETS($\mathcal{M}, f^*, \mathcal{R}, \mathcal{G}$)
  Partition $\mathcal{M}$ into groups based on $\mathcal{G}$
  for each group G in partitioned $\mathcal{M}$ **do**
    $\mathcal{B}_G \leftarrow$ available bounding boxes on frame f^* in group G
    for each referring expression $r_i \in \mathcal{R}$ **do**
      $b_i \leftarrow$ REXSEEK($f^*, \mathcal{B}_G, r_i$)
      Assign r_i to mask in G corresponding to box b_i
    end for
    Post-process assignments to ensure valid mapping
  end for
  return $\mathcal{M}$
end procedure

Prompt P.1: Entity Recognizer

Prompt: What "active" scene entities can you identify from the video? An entity refers to an object, and "active" scene entities are scene objects that have any dynamic behaviors, such as actions, interactions with others, or movements. Please compile a list of clearly visible "active" scene entities from the video. Use entity appearance in concise description to distinguish one "active" scene entity from another if possible.

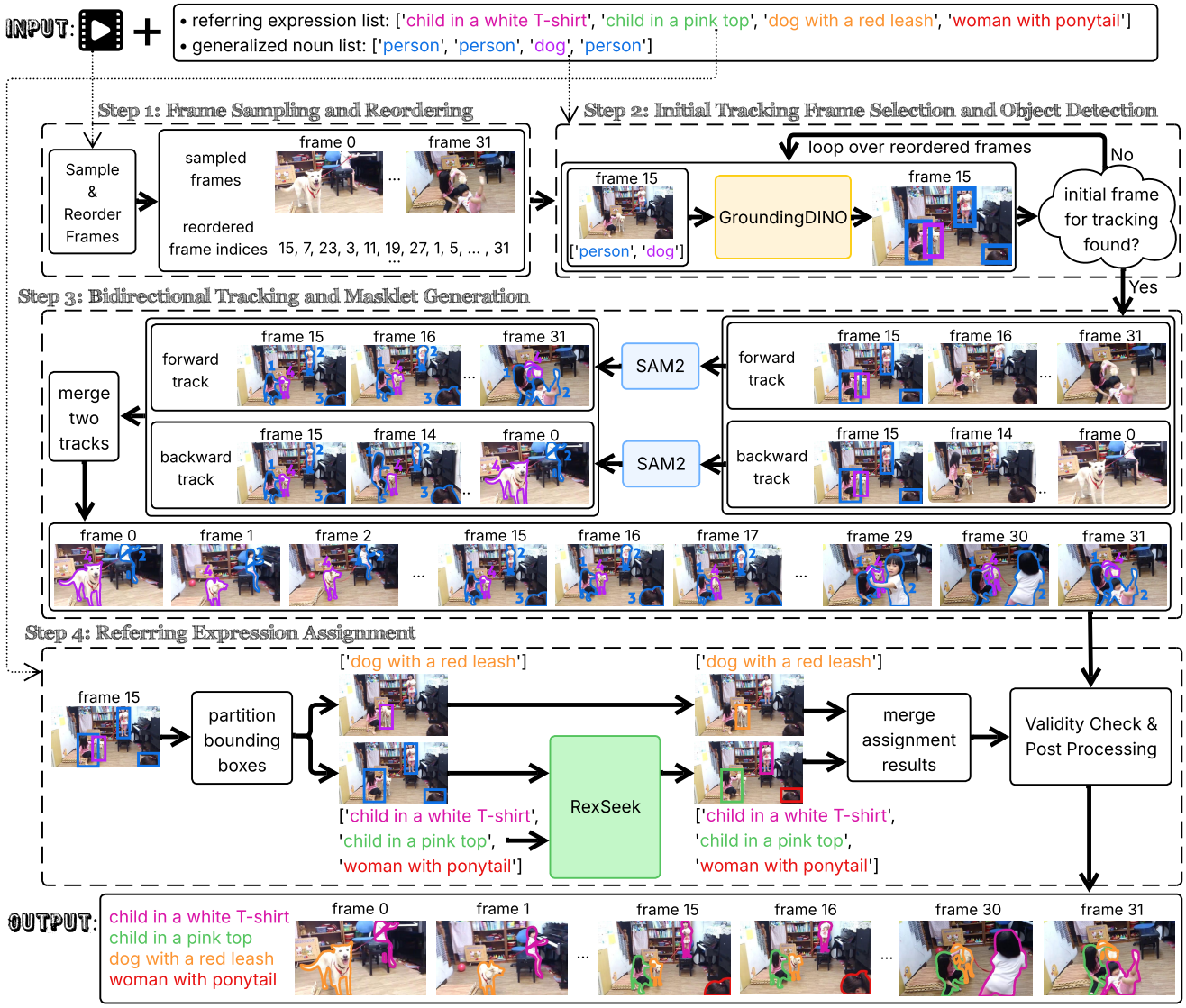


Figure 4. **Overview of our novel Referring Masklet Generation Pipeline within Strefer.** This pipeline produces tracked segmentation masks from videos with complex structures based on multi-word natural language referring expressions. Our pipeline is carefully crafted to address key limitations overlooked by prior works [22, 37, 68] by orchestrating complementary strengths of the state-of-the-art pixel-level vision foundation models to achieve more effective results. It handles challenging scenarios, including multiple same- or similar-category entities described differently, entities absent in the first frame, and entities that temporarily exit and re-enter the scene.

A.3. Model Details

We synthesize video instruction tuning data to enable Video LLMs fine-grained, mask-level comprehension at any specific regions and any timestamps for a given video. These capabilities are achieved by tuning a general, coarse-level next-token-prediction Video LLM with plug-and-play enhancements for spatiotemporal localized understanding.

To support detailed region-level understanding, we incorporate the spatiotemporal object encoder design from VideoRefer [68], which enables the model to understand fine-grained masks and masklets. For precise timestamp-level comprehension, we introduce learning special temporal tokens inspired by GroundedLLM [50], allowing the model to interpret specific moments in time properly.

We integrate these plug-and-play modules into BLIP-3-Video [45]. The full model architecture is described in detail below.

A.3.1. Architecture Overview

The Video LLM processes a video and a user's multimodal query to generate a textual response. A multimodal query consists of the textual component of the question, a masklet

Algorithm 2 Video Clipping Pipeline

procedure CLIPVIDEO($video$)
  $B \leftarrow$ PYSCENEDETECT($video$, threshold=20)
  if $B = \emptyset$ **and** DURATION($video$) ≥ 3 sec **then**
    $E \leftarrow$ GETEMBEDDINGS($video$);
    $D \leftarrow$ PAIRWISEDISTANCES(E)
    $T \leftarrow$ CLUSTERINGAUTOTHRESHOLD(D, 1.7);
    $C \leftarrow$ HIERARCHICALAGGLOMERATIVECLUSTERING(E, T)
    $B \leftarrow$ EXTRACTCLIPTIMESTAMPBOUNDARIES(C)
  end if
  return B
end procedure

procedure CLUSTERINGAUTOTHRESHOLD(D, f)
  $m \leftarrow \text{mean}(D)$; $s \leftarrow \text{std}(D)$; $M \leftarrow \max(D)$
  return $\min(m + f \cdot s,\ M)$
end procedure

along with its associated frames referring to a specific region within the video, and optionally, one or more specific timestamps within the video.

The architecture of the Video LLM is illustrated in Fig. 3 of the main paper. At a high level, the LLM processes four types of input tokens: (i) *visual tokens*, which encode the global context of the video; (ii) *region tokens*, which represent specific visual regions referenced in the user query (e.g., a mask or masklet); (iii) *timestamp/temporal tokens*, which indicate particular temporal locations within the video; and (iv) *text tokens*, which represent the textual content of the query itself. These tokens are jointly fed into the LLM, which then auto-regressively generates a textual response.

The construction of visual, region, and timestamp tokens from raw inputs—namely, the video and multimodal user query—is detailed in Appendix A.3.2, Appendix A.3.3, and Appendix A.3.4, respectively.

A.3.2. Video Token Representation

Given an input video $x_{\text{v}} \in \mathbb{R}^{T_v \times 3 \times H_v \times W_v}$, where T_v is the number of frames and H_v, W_v are the height and width of the frames, a visual encoder extracts the video's global visual features $f_{\text{v}} \in \mathbb{R}^{t_v \times d_v \times h_v \times w_v}$.

A Video-Language Connector is then applied on top of the visual encoder to project the global visual features into a sequence of visual tokens $e_{\text{v}} \in \mathbb{R}^{L_v \times d}$, where d represents the dimensionality of the language model's input token space, and L_v is the number of visual tokens of a video. This connector aligns the visual features to the input space of a language model while preserving semantics relevant for multimodal understanding. In some designs (e.g., BLIP-3-Video [45]), the connector also incorporates a token compression module to reduce the number of tokens, improving efficiency without sacrificing critical information.

A.3.3. Masklet Reference Token Representation

Our modified Video LLM is designed to understand user queries about videos that involve spatial or spatiotemporal, local regional references. To support diverse, free-form spatial reference from users (e.g., points, scribbles, etc.), we standardize them by converting these free-form spatial references into regional masks before processed by the model. This approach is effective because many forms of spatial reference can be easily transformed into masks using off-the-shelf tools like SAM2 [43].

Mask and Masklet. A regional mask is represented as a 2D binary matrix $\mathbb{R}^{H_m \times W_m}$, where H_m and W_m are the height and width of the image containing the region of interest, with a value of 1 inside the region and 0 outside. When extended over time, a temporal sequence of such regional masks $x_{\text{r}} \in \mathbb{R}^{T_m \times H_m \times W_m}$ is referred to as a masklet. Since a mask is special case of masklet with only one frame, we describe the masklet feature extraction process below.

Masklet Token Representation. Leveraging the same visual encoder, our model extracts image feature maps $f_{\text{m}} \in \mathbb{R}^{t_m \times d_v \times h_m \times w_m}$ for the frames that contain the masklet x_{r}. The masklet x_{r} and its corresponding frames' feature maps f_{m} are then processed by a Region-Language Connector, which outputs region tokens $e_{\text{r}} \in \mathbb{R}^{L_r \times d}$ that are aligned to the language space, where L_r is a predefined number of region tokens.

The Region-Language Connector begins by resizing the binary masklet x_{r} via bilinear interpolation to match the spatial (and temporal if the visual encoder condenses the time axis) dimensions of f_{m}, yielding a resized masklet of shape $\mathbb{R}^{t_m \times h_m \times w_m}$. A Mask Pooling operation is then applied: average pooling is performed over the spatial locations within the mask region for each frame, producing a pooled feature representation $p \in \mathbb{R}^{t_m \times d_v}$. This representation can be interpreted as a sequence of t_m region tokens, each of dimensionality d_v.

To reduce the temporal redundancy, a Temporal Token Merge module condenses the t_m tokens into L_r representative ones ($L_r < t_m$). Specifically, for $p \in \mathbb{R}^{t_m \times d_v}$, cosine similarities are computed between each pair of temporally adjacent tokens:

$$\mathbf{s}_{i,i+1} = \frac{p^i \cdot p^{i+1}}{\|p^i\| \cdot \|p^{i+1}\|}, \quad 0 \leq i < t_m - 1 \tag{1}$$

This yields a similarity vector $\mathbf{s} \in \mathbb{R}^{t_m - 1}$. A similarity threshold θ is then selected as the L_r-th largest value in $\mathbf{s}$. Next, the sequence p is processed sequentially from the beginning to the end to form token groups. An initially empty group is created and the first token in p is added to it. For each index i from 0 to $t_m - 2$, if $\mathbf{s}_{i,i+1} \geq \theta$, then p^{i+1} is added to the current group. Otherwise, the current group is finalized, and a new group is initiated with p^{i+1}.

This process produces exactly L_r token groups. Each group is finally merged into a single representative token by averaging the embeddings of all tokens within the group.

Finally, the resulting L_r tokens, each in $\mathbb{R}^{d_v}$, are pro-

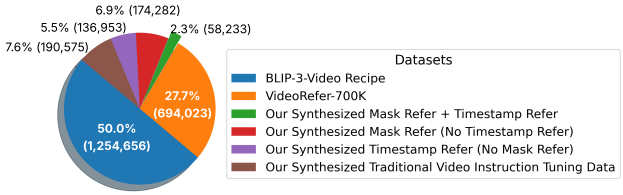


Figure 5. Data composition of our final recipe, used in our experiments in Sec. 4 of the main paper.

jected into the language embedding space via an MLP, producing the final region tokens $e_{\mathrm{r}} \in \mathbb{R}^{L_r \times d}$.

Note that the Temporal Token Merge module is bypassed when the user query involves only a single frame mask (as opposed to a masklet).

A.3.4. Timestamp Reference Token Representation

By design, our model is effective in scenarios where users may refer to specific times within videos in their queries. However, LLMs often struggle with interpreting numerical values [46]. To address this challenge, we adopt the Temporal Token Representation method introduced in Grounded-VideoLLM [50], which discretizes continuous time into a sequence of temporal tokens, making time-related reasoning more manageable for LLMs.

Suppose the video has a duration of L seconds. We divide it into M equal-length, non-overlapping, and non-spacing segments, resulting in $M+1$ anchor points that span from the start to the end of the video. These anchor points, labeled from $<0>$ to $<M>$, represent evenly spaced temporal positions throughout the video. Each specific timestamp within the video is mapped to an anchor point and then encoded as a temporal token. For example, $<0>$ marks the beginning of the video, while $<M>$ represents the end. These $M+1$ anchor points are added to the LLM's vocabulary (by expanding the LLM's vocabulary), enabling unified modeling of time alongside text. Mathematically:

A specific continuous timestamp τ can be easily converted to a temporal token $<t>$ and vice versa:

$$t = \mathrm{Round}\left(M \cdot \frac{\tau}{L}\right), \quad \tau = L \cdot \frac{t}{M} \tag{2}$$

In this way, specific timestamps in the user query are converted into timestamp anchor points. Both text and timestamp anchor points are then mapped to embeddings through the extended word embedding layer of the LLM, forming interleaved text tokens and temporal tokens.

In our model tuning and evaluation experiments, since our Video LLM processes 32 input frames, we set $M = 31$ to learn 32 temporal tokens.

A.4. Experimental Implementation Details

To evaluate the quality of our synthesized instruction data, we integrate it into a base video instruction tuning recipe, which combines the video instruction-tuning data used by BLIP-3-Video [45] with VideoRefer-700K [68]. Our baseline is the model tuned on this base recipe. The video instruction-tuning data used by BLIP-3-Video comprises data from multiple sources, including Mira [21], VideoInstruct-100K [35], MSVD-QA [57], MSRVTT-QA [57], ActivityNet-QA [66], TGIF-QA [19], and NExT-QA [56]. VideoRefer-700K is a recently released instruction-tuning dataset for video mask and masklet referring tasks, but it lacks timestamp-referring instructions.

The full model is tuned except the visual encoder. The visual encoder is not fine-tuned due to insufficient data, which prevents effective training. Since the encoder is designed to extract complex visual patterns and features from raw RGB signals, it requires a large amount of data to generalize well. To be specific, we start from the pretrained image-comprehension vision LLM, the pre-trained BLIP-3 [61] model, with additional untrained architectural enhancements from BLIP-3-Video [45], as well as those described in Sec. 3.3 of the main paper. We adapt BLIP-3 for video and masklet comprehension by fine-tuning the full model illustrated in Fig. 3 of the main paper, except for the visual encoder, using 32 frames per video and 32 temporal tokens. Other hyperparameters, such as learning rate and batch size, were selected based on downstream evaluation results for the baseline. However, when tuning the model using recipes integrated with our data, we did not change any hyperparameters from those used in the baseline. The training takes roughly 1 day and requires 3×8 H200 GPUs. The resulting model has 4B parameters.

In the baseline and ablation models, if its training data lacks mask-referring instructions, the corresponding modules are excluded; likewise, timestamp-related modules are omitted if timestamp-referring instructions are not present in training. Therefore, the 'Baseline Ablation' model presented in the result tables shares the same architecture as BLIP-3-Video [45]; the model does not include the plug-and-play modules described in Sec. 3.3 of the main paper, as its training data lacks instructions that refer to masks or timestamps. The 'Baseline' model presented in the result tables does not have 'Timestamp Conversion' or an extended LLM vocabulary for learning special temporal tokens (see Fig. 3 of the main paper) due to the lack of instruction data involving specific timestamps. We report all results for GPT-4o and GPT-4o-mini as presented in [68].

Our synthetic data includes full-length masklets per referring entity, but for efficient training, we sample a single mask on a random frame per instruction-response pair. At evaluation, we use the full masklet. Training with full masklets is expected to further improve performance.

A.5. Evaluation Details

We describe the evaluation benchmarks for Mask-Referred Regional Description, Mask-Referred Regional QA, and Timestamp-Referred Video QA below, as these represent less common evaluation settings for Video LLMs.

VideoRefer-Bench^{D} [68] assesses the model's ability to describe an entity across a video, given a mask or masklet of that entity. The benchmark comprises 400 videos from the test set of Panda-70M [6].

To evaluate performance on this benchmark, we use the following instruction template: "`Please give a detailed description of the highlighted object <region> in the video.`" The word `<region>` is substituted with model-extracted regional tokens if the model has built-in mechanisms to extract regional features.

The model evaluation is performed by GPT-4o by assigning scores to the generated predictions on a scale range from 0 to 5 across the following four dimensions [68]:

- Subject Correspondence: This dimension evaluates whether the subject of the generated description accurately corresponds to that specified in the ground truth.
- Temporal Description: This aspect analyzes whether the representation of the object's motion is consistent with the actual movements.
- Appearance Description: This criterion assesses the accuracy of appearance-related details, including color, shape, texture, and other relevant visual attributes.
- Hallucination Detection: This facet identifies discrepancies by determining if the generated description includes any facts, actions, or elements absent from reality, like imaginative interpretations or incorrect inferences.

VideoRefer-Bench^{Q} [68] evaluates a model's ability to answer video entity-related questions, given one or more entities' masks or masklets within a video. The benchmark includes 1,000 multiple-choice questions spanning 198 videos sourced from various datasets, including the test set of MeViS [12], A2D-Sentences [14], and Refer-YouTube-VOS [53]. Questions are crafted to assess different dimensions of understanding, including Basic Questions, Sequential Questions, Relationship Questions, Future Predictions and Complex/Reasoning Questions.

Sequential Questions typically ask about entity action and ordering; Basic Questions typically concern attributes like object color. Relationship Questions involve more than one object regions in the question. Future Predictions involve weakly grounded reasoning about forthcoming events. Notably, models generally perform best on Complex/Reasoning Questions, making this category the easiest despite its name.

We use the following instruction template: "`Please answer the following question about the <region>. {question}`".

Timestamp-based Yes/No QA on QVHighlights is a task that repurposes existing annotations from the video highlight detection dataset, QVHighlights [23]. Specifically, for each annotated segment—defined by a start and end timestamp and an associated language description—we construct a question prompt in the following form: '`Does the following description accurately reflect what happens in the video between <start_time> and <end_time>? Description: {description}. Respond with 'Yes' or 'No' only.`" Each of these prompts is assigned the ground truth answer "`Yes`".

To generate negative (i.e., "`No`") samples, we randomly select segments from the same video that do not overlap with any annotated intervals. To ensure this, we first expand each annotated timestamp by a buffer of 5 seconds on both sides, then merge overlapping intervals to form a set of excluded ranges. We then identify all remaining gaps in the video timeline that lie outside these excluded regions. From these valid gaps, we randomly select a new segment that satisfies a minimum duration of 10 seconds. A description from an annotated segment is then paired with this unrelated time window to form a mismatched QA example with the correct answer "`No`".

We ensure a balanced answer distribution, with almost 50% of the samples labeled as "`Yes`" and 50% as "`No`".

For each question, we substituted `<start_time>` and `<end_time>` with their corresponding timestamps. For models that do not learn temporal tokens, timestamps are represented by default in the `HH:MM:SS.xxx` format. For models that do learn temporal tokens, we use temporal tokens to substitute `<start_time>` and `<end_time>`. For example, if a model learns 32 temporal tokens and the video's duration is 90 seconds, a timestamp like `00:00:19.228` is converted to `<7>`.

A.6. Training-Free Methods Experimental Results

It is worth noting that incorporating the plug-and-play modules and modify the architecture of the pre-trained general-purpose Video LLM for space-time referring is not strictly necessary. We explore training-free approaches—SoM [63] for masklet comprehension and NumberIt [55] for timestamp understanding.

SoM: Mask-Overlay-Frame Prompting. We follow the implementation of the Set-of-Mark (SoM) method from VideoRefer [11] to apply masks to video frames, as originally proposed by [63]. We also changed the question prompt into: "`I have outlined an object with a red contour in the video. Please describe the object in detail.`" The results are presented in Table 5. After applying SoM, the average performance on the Mask-Referred Video Regional

Mask-Referred Regional Description (VideoRefer-Bench^{D} [68])	Samples Added (%)	**Avg.**	Subject Correspondence	Temporal Description	Appearance Description	Hallucination Detection
Baseline Ablation Model	N/A	2.7308	3.5200	2.4235	2.5639	2.4160
Baseline Ablation Model + SoM [63]	N/A	2.6593	3.5600	2.1834	2.3576	2.5363

Table 5. **Regional Description** results on VideoRefer-Bench^{D} before and after applying the training-free method, SoM [63].

Timestamp-Referred QA (Yes/No)	Samples Added (%)	QVHighlights [23]
Baseline Ablation: Video Instruction-Tuning Data [45]	N/A	0.5297
Baseline Ablation: Video Instruction-Tuning Data [45] + NumberIt [55]	N/A	0.5301
Baseline: Base Recipe (1,948,679 samples)	N/A	0.5288
Baseline: Base Recipe (1,948,679 samples) + NumberIt [55]	N/A	0.5311
+ $\mathcal{G}6$ + $\mathcal{G}7$ + $\mathcal{G}8$ + Remaining $\mathcal{G}5$ (Sampled) + $\mathcal{G}1$	28.73%	**0.6031**
+ $\mathcal{G}6$ + $\mathcal{G}7$ + $\mathcal{G}8$ + Remaining $\mathcal{G}5$ (Sampled) + $\mathcal{G}1$ + NumberIt [55]	28.73%	**0.6041**

Table 6. **Results of timestamp-based Yes/No QA on QVHighlights** before and after applying the training-free method, NumberIt [55].

Description task decreases, but performance increases on certain metrics, e.g., Subject Correspondence.

Our analysis reveals that the effectiveness of training-free SoM is highly sensitive to the way masks are rendered on the video frames. In our initial implementation, we used thicker mask boundaries and semi-transparent red fill color. This approach led to severe hallucinations by the model, which often misinterpreted masked regions as merely red-colored objects. In contrast, the SoM implementation from VideoRefer [11] uses thinner boundaries and fully transparent fills, resulting in significantly improved performance over our version. Nevertheless, the performance remains lower than the baseline without any SoM masking.

NumberIt: FrameID-Overlay-Frame Prompting. Similar to SoM, NumberIt [55] overlays the frame ID at a specific location on each frame. We overlaid the frame ID in red, following the authors' suggestion, and placed each ID in the top-left corner of the corresponding frame (the resulting rendering effect is similar to Fig. 20). We also modified each question as follows: `The red numbers on each frame represent the frame number. Does the following description accurately reflect what happens in the video between <frame_start> and <frame_end>? Description: {description}. Respond with 'Yes' or 'No' only.` For each question, we substituted `<frame_start>` and `<frame_end>` with their corresponding frame IDs. The results are listed in Table 6. The performance on timestamp-based Yes/No Video QA in QVHighlights shows a slight improvement after applying the training-free method, NumberIt.

In summary, using a pretrained general-purpose Video LLM for space-time referring tasks does not necessarily require modifying the model architecture or fine-tuning the model. Training-free approaches such as SoM and NumberIt can help the model perform mask-referring or timestamp-referring tasks, though their performance gains may be limited. We hypothesize that incorporating these techniques during model fine-tuning—while preserving the original architecture—may lead to performance gains [55], which we leave as future work.

A.7. Limitations and Future Directions

Strefer synthesized data is not error-free. For example, in the last blue-highlighted video segment shown in Fig. 6 of the Appendix, **Strefer** identifies the woman as not present. However, a human viewer would easily identify the woman in that segment while watching the video, despite the fact that she is largely occluded and the frames are mostly occupied by the child. This error also occurs because we did not employ a more complex, dense video captioning framework that leverages *inter-segment* information, such as a hierarchical [74] or differential video captioning [5] method. We actually tried these approaches, but they did not yield better results than clip-by-clip captioning using current open-source models. We also experimented with several alternative open-source mid-scale Video LLMs, including Qwen2.5-VL-7B-Instruct, LLaVA-NeXT-Video-34B, LLaVA-OneVision-7B, and Tarsier-7B. Ultimately, we selected Tarsier-34B, as it appeared to provide more accurate, action-centric descriptions.

Scenes characterized by high visual clutter and significant dynamic variations continue to pose substantial challenges. Fig. 11 illustrates a failure case of our referring masklet generation—the woman is not segmented in the first four frames. This highlights that videos with heavy motion blur and long-range dependencies remain challenging to handle. The issue stems from the tracking limitations of SAM2, the tracking and segmentation model we employ which is sensitive to the selection of the start tracking frame. Fig. 7 presents another example of tracking and segmentation failure—the child in the black shirt lacks associated masks in frames 10 and 11, despite being clearly visible.

Strefer may inherit limitations from the underlying models used in its modular components, such as pixel-level foundation models, LLMs, and Video LLMs. For instance, LLMs and Video LLMs are known to hallucinate, potentially introducing misleading information into annotated video metadata and synthesized question–answer pairs. Similarly, the extraction of pixel-level information may be less reliable in videos with highly similar individuals or densely populated scenes (e.g., crowded urban environments), which the pixel-level foundation models we used could not reliably handle. Despite these challenges, the modular nature of **Strefer** positions it well to benefit from future improvements in its underlying models—including LLMs, Video LLMs, and grounding vision foundation models—as they continue to evolve.

Strefer involves multiple models, which may hinder exact reproduction of its pseudo-annotation, data synthesis as well as our model training procedures for research groups with limited resources. However, **Strefer** is a multi-stage modular framework, wherein individual model components can be substituted with more computationally efficient alternatives, enabling flexible adaptation under varying resource constraints.

In terms of the limitations inherent to models trained on **Strefer**-synthesized data, Fig. 22 shows a typical failure case where the region indicated by the mask is neither the primary foreground object nor centrally positioned in the frame. This reflects a common issue across all models, including baselines, which tend to exhibit both a foreground main object bias and a center bias. Furthermore, Fig. 23 illustrates another failure case, underscoring the persistent difficulty in capturing fine-grained action semantics.

Future work may further improve **Strefer** by refining individual modules—for example, improving the video clipping pipeline to produce entity-centric segments, and incorporating feedback-verification mechanisms to minimize hallucinated content in video metadata and instruction-following pairs. Additionally, given the potential for error propagation in our modular framework, as well as the nature of synthetic data, which may not fully match real human question distributions, future research is encouraged to develop effective filtering strategies for the synthetic instruction-tuning data as a final quality assurance or adjustment step in the data engine, thereby enabling more efficient, reliable, and robust model training.

For the development of the referring and reasoning video model, our current trained models are limited to mask-based spatial referring and is not trained for other types of spatial references such as points, boxes, and scribbles. However, since these other forms of spatial references are inherently sparser than masks and can be easily derived from object segmentation masks, an avenue for future research is to explore transforming the current mask-level instruction-tuning data produced by **Strefer** into alternative data formats, and to train models that can comprehend more diverse forms of user spatial references.

Moreover, the LLM backbone of models we trained on our data is based on the pretrained microsoft/Phi-3-mini-4k-instruct [1]. As with many other Video LLMs, the performance of our model is heavily influenced by the capabilities of the underlying LLM. We encourage further research into training Video LLMs on **Strefer**-synthesized data using larger and more powerful models. However, this typically demands significantly more tuning data and greater computational resources.

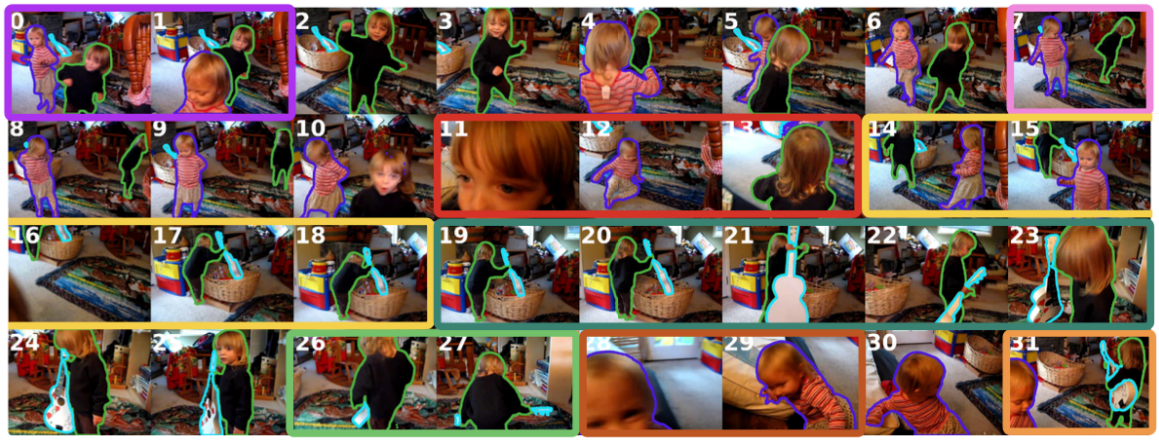
In our experiments, we conducted rather limited exploration of the optimal training data mixture. As a result, the current composition may not represent the most effective setup for fostering broad, balanced and transferable skills. Future work could focus on systematically optimizing the data composition, which is likely to result in more substantial and consistent performance gains across diverse benchmarks and metrics.

Finally, our model is grounded at the perception level rather than at the output response generation level. While grounding at the output level offers a more direct path to interpretable video-language reasoning, it requires training data with high-fidelity spatiotemporal annotations. At present, the boundary of the fine-grained space-time information generated by **Strefer** may lack the precision required to reliably supervise such models.

Our work centers on *referring* understanding-where the model leverages fine-grained spatiotemporal cues as conditional input, and our models are trained using synthesized instruction-tuning data, rather than being directly supervised by pseudo-annotated dense video metadata. This setup is inherently more robust to moderate imperfections—such as missing entities or imprecise temporal boundaries. For example, as shown in Fig. 7, even when the temporal span of the masklet for the child in the black shirt is incomplete, the associated instruction-response pair remains accurate and meaningful. This resilience arises because referring understanding does not require an exhaustive coverage of the language-described pixel-level space-time information, making it more adaptable under the current data limitations.

Looking ahead, advancing output-level spatiotemporal grounding in Video LLMs holds significant promise for improving their generalization, reliability and fine-grained spatiotemporal reasoning skills. We encourage future work to pursue this direction by leveraging more accurate spatial-temporal annotations aligned with language, ideally enabled by an enhanced, scalable, and automated data generation pipeline.

Figure 6. **Example of Strefer-Synthesized Instruction-Response Pairs (left) and Pseudo-Annotated Video Metadata (right).** Each instruction begins with the prefix: "Please answer the following question about the <region>" (and the prefix is omitted in the figure). For each instruction-response pair, the boundary of the object mask referred to by <region> is shown next to the pair and highlighted in color. **Strefer** automatically clips the video into segments and pseudo-annotates the video metadata—including active entities, their locations (as masklets), and action timelines—for complex video scenarios, such as scenes containing multiple entities of the same category, and cases where entities do not appear in the first frame, or temporarily exit and re-enter the frame; based on the video metadata, it generates instruction-response pairs, requiring no legacy annotations. Though current implementation of **Strefer** does not any use proprietary models, without the need to annotate large volumes of new videos, instruction data from **Strefer** empowers models for space-time referring and spatiotemporal reasoning (*ref.* Table 2, 3, and 4).



Strefer

00:00:00.000 ~ 00:00:02.044:
child in the black shirt
- The child in the black shirt is moving their arms and body, possibly dancing or playing.

child in the striped shirt
- The child in the striped shirt is moving their arms and body, possibly dancing or playing.

toy guitar
- None

Strefer

00:00:19.228 ~ 00:00:21.688:
child in the black shirt
- The child in the black shirt is moving their feet and body, possibly dancing or playing on a rug.

child in the striped shirt
- The child in the striped shirt is standing and slightly shifting their weight while looking down at the floor.

toy guitar
- None

Strefer

00:00:32.407 ~ 00:00:39.539:
child in the black shirt
- The child in the black shirt is moving their head and body.

child in the striped shirt
- The child in the striped shirt is sitting on the floor, then stands up and begins to walk.

toy guitar
- None

Strefer

00:00:39.539 ~ 00:00:51.301:
child in the black shirt
- The child in the black shirt walks towards a basket, picks up a toy guitar from the basket, and holds it.

child in the striped shirt
- The child in the striped shirt walks towards the camera, turns around, and walks away.

toy guitar
- The toy guitar is being held and manipulated by one of the children, who appears to be examining or playing with it.

Strefer

00:00:51.301 ~ 00:01:06.9:
child in the black shirt
- The child in the black shirt is interacting with a toy guitar, moving it from a basket to the floor.

child in the striped shirt
- None

toy guitar
- The toy guitar is being lifted out of a basket by a child.

Strefer

00:01:15.2 ~ 00:01:18.287:
child in the black shirt
- The child in the black shirt is kneeling on the floor, holding a white object, and appears to be interacting with the floor or something on it.

child in the striped shirt
- None

toy guitar
- The toy guitar is being held by a person and is being moved around.

Strefer

00:01:24.918 ~ 00:01:26.503:
child in the black shirt
- None

child in the striped shirt
- The child in the striped shirt is sitting on the floor, leaning forward, and appears to be interacting with something out of frame. The child's movements suggest engagement and focus on the activity.

toy guitar
- None

Strefer

00:01:26.503 ~ 00:01:30.007:
child in the black shirt
- The child in the black shirt is playing a guitar.

child in the striped shirt
- The child in the striped shirt is sitting on the floor, turning their head to the right, and then turning back to face the camera.

toy guitar
- The toy guitar is being played by a child, with movements suggesting strumming and handling of the instrument.

Strefer — **Instruction:** What was the person doing between 00:00:51.301 and 00:01:06.9 in the video?
Response: Interacting with a toy guitar, moving it from a basket to the floor.

Strefer — **Instruction:** What did the person do at the end of the video?
Response: Playing a guitar.

Strefer — **Instruction:** During which part of the video is it seen dancing or playing energetically?
Response: The beginning.

Strefer — **Instruction:** What was the child doing after standing up from sitting on the floor?
Response: After the child in the striped shirt stood up from sitting on the floor, the child began to walk.

Strefer — **Instruction:** Who lifted it out of a basket before it was being played with by a child?
Response: A child.

Figure 7. Example of **Strefer**-synthesized instruction-response pairs (bottom) and video metadata (top).

Figure 8. **Qualitative Results for Referring Masklet Generation.** In this video, our method accurately generates masklets corresponding to the input referring expressions. In contrast, GroundedSAM2 [44] fails to differentiate between the two children and also fails to detect and track the woman, who appears midway through the video and occupies only a small portion of the frames.

Prompt = “man in the grey shirt.man in the black jacket.”

Video Frames

Ours

man in the grey shirt | man in the black jacket

Prior Method - GroundedSAM2 [44]

man | man

Figure 9. **Qualitative Results for Referring Masklet Generation.** In this video, our method accurately generates masklets corresponding to the input referring expressions. In contrast, GroundedSAM2 [44] fails to differentiate between the man in the grey shirt and the man in the black jacket.

Figure 10. **Qualitative Results for Referring Masklet Generation.** Our method accurately generates masklets corresponding to the input referring expressions. In contrast, GroundedSAM2 [44] converts the input text prompt into the class names *groom officiant*, *groom*, *bride*, and *bridesmaid*. GroundedSAM2 then fails to detect the bridesmaid in the video, while incorrectly assigning the class name *bridesmaid* to the actual bride, *groom* to the officiant, and *groom officiant* to the groom.

Figure 11. **Failure Results for Referrin Masklet Generation.** Our method fails to consistently track the woman on a bicycle throughout the video, while GroundedSAM2 [44] fails to detect, track, and differentiate the individuals referenced in the input text prompt. Videos with heavy motion blur and long-range dependencies remain challenging to handle.

Please give a detailed description of the highlighted object <region> in the video.

Ground Truth: The table in the video is a long, wooden table with a polished surface. It is situated in a formal setting, likely a courtroom or a meeting room. The table has several nameplates in front of it, indicating the positions of the individuals seated behind it.

Baseline: The object in question is a man dressed in a suit and tie, seated at a desk. He is positioned on the left side of the image, with his back to the camera. Throughout the video, he remains seated, maintaining a composed and professional demeanor. His attire is formal, consisting of a dark suit, white shirt, and a tie. The man's posture is upright, and he appears to be engaged in a serious discussion or meeting. The background behind him is subdued, with a focus on the man and his immediate surroundings.

The object in question is a table. Initially, it is seen with a man in a suit sitting at it, holding a cup. The table is positioned in a formal setting, likely a conference room or a meeting hall. The man appears to be engaged in a discussion or a meeting, as he is seated and holding a cup, possibly containing a beverage. The table itself is wooden and has a polished finish, with a few nameplates and papers placed on it, indicating a formal meeting or conference.

Figure 12. **Qualitative Results**. This sample is drawn from VideoRefer-Bench^{D}, designed to assess a model's performance on the task of **Mask-Referred Regional Description**. The boundary of the region referred to by the mask in this sample is highlighted in purple. While the video includes several individuals as prominent foreground elements, the masklet specifically refers to the table, not the people. The baseline model, however, fails to interpret the mask correctly and mistakenly answers that the referred object is a man. In contrast, the model trained on **Strefer**-generated data accurately identifies the masklet-referred region as a table.

Please give a detailed description of the highlighted object <region> in the video.

Ground Truth: The parking lot is a large, open area with multiple cars parked in designated spaces. The surface is smooth and appears to be made of asphalt. The parking lot is bordered by orange and white barriers on one side, and there are several orange cones placed throughout the area.

Baseline: A car is parked on the left side of the road. It is a white vehicle with a blue stripe running along its side. The car is stationary and does not appear to be in motion.

The object in question is a parking lot. Initially, it is depicted as a vast, open space with a smooth, dark asphalt surface. The parking lot is bordered by a series of orange and white barriers, which are arranged in a neat, parallel fashion, creating a clear demarcation for the area. The barriers are evenly spaced, ensuring a uniform appearance.

Figure 13. **Qualitative Results**. This sample is drawn from VideoRefer-Bench^{D}, designed to assess a model's performance on the task of **Mask-Referred Regional Description**. The boundary of the region referred to by the mask in this sample is highlighted in purple. The baseline model tends to have a foreground bias, fails to interpret the masklet correctly, and mistakenly responds that the referred region is a car. In contrast, the model trained on **Strefer**-generated data accurately identifies the masklet-referred region as a parking lot.

What action is <object0><region> performing? Pick the best option:
(A) Sitting and eating
(B) Turning around and moving to the right
(C) Lying down to eat
(D) Walking to the leftmost part

Ground Truth: (B) Turning around and moving to the right

Baseline: (A) Sitting and eating

(B) Turning around and moving to the right

Figure 14. **Qualitative Results**. This sample is drawn from VideoRefer-Bench^{Q}, designed to assess a model's performance on the task of **Mask-Referred Regional QA**. The boundary of the region referred to by the mask in this sample is highlighted in purple. The model trained on **Strefer**-generated data correctly identifies the masklet-referred region and action.

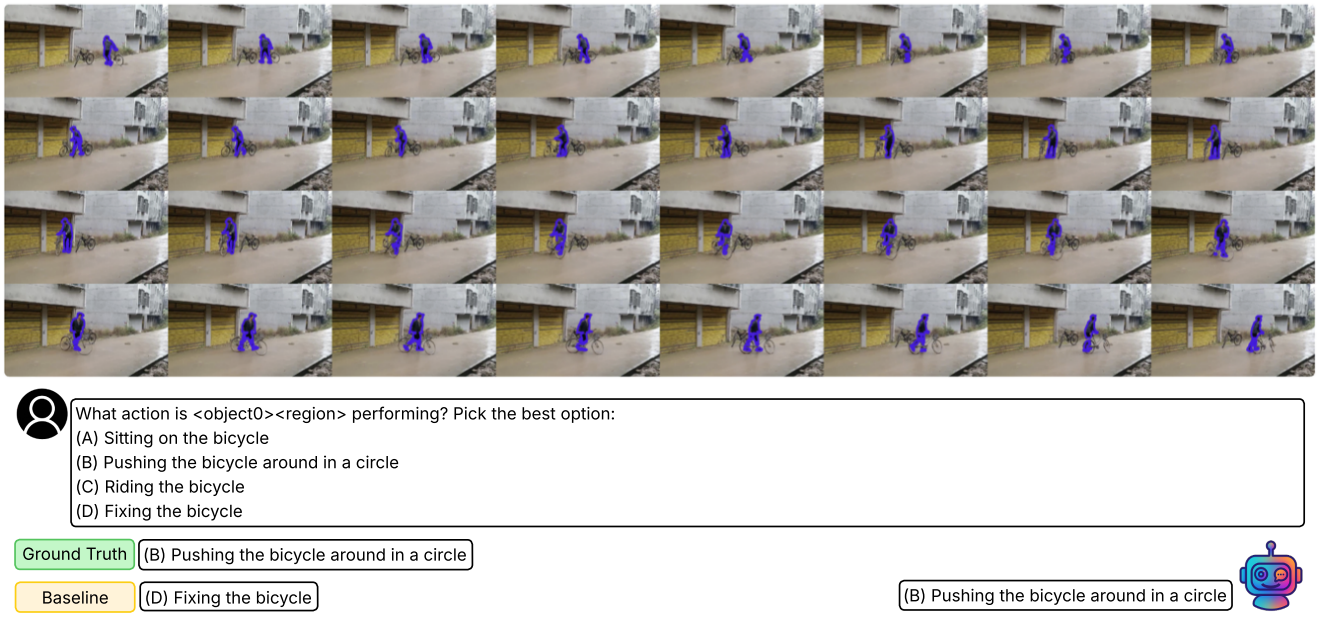
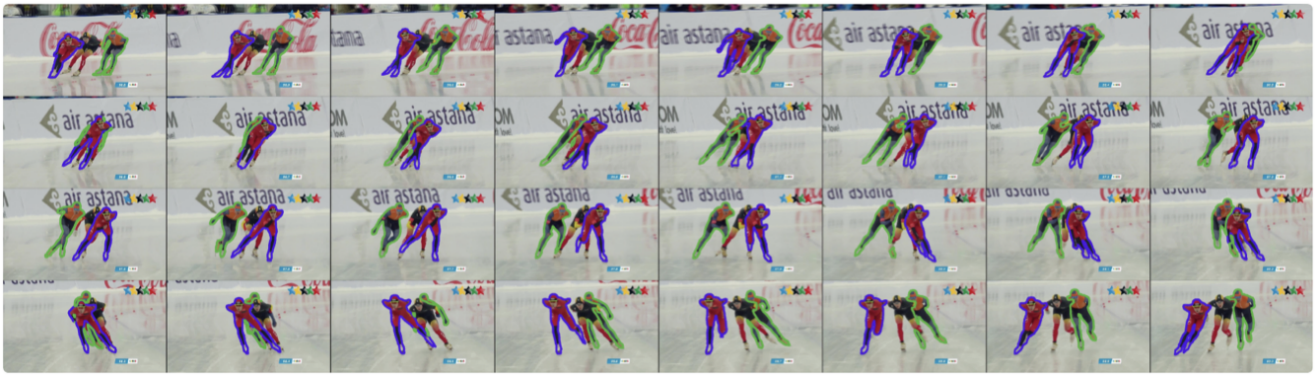


Figure 15. **Qualitative Results**. This sample is drawn from VideoRefer-Bench^{Q}, designed to assess a model's performance on the task of **Mask-Referred Regional QA**. The boundary of the region referred to by the mask in this sample is highlighted in purple. In this sample, the model must demonstrate fine-grained spatiotemporal action understanding due to the small size of the mask and the subtle motion differences between the correct and negative options. The model trained on **Strefer**-generated data successfully identifies both the region referred to by the masklet and the corresponding action.

Figure 16. **Qualitative Results**. This sample is drawn from VideoRefer-Bench^{Q}, designed to assess a model's performance on the task of **Mask-Referred Regional QA**. The boundary of the region referred to by the mask in this sample is highlighted in purple. The model trained on **Strefer**-generated data correctly identifies the masklet-referred region and action.



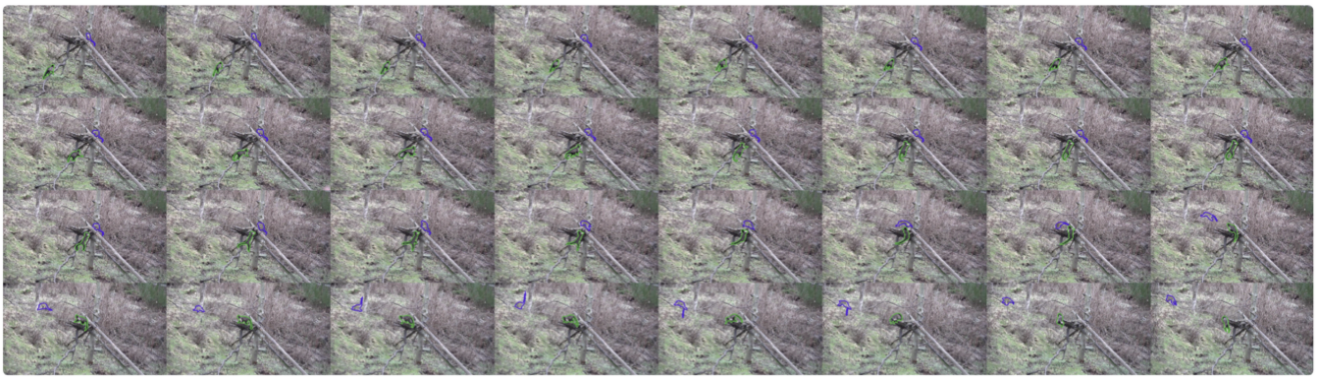
How does <object1><region>'s position relate to <object3><region>'s position? Pick the best option:
(A) <object1> is in front of <object3>
(B) <object1> is behind <object3>
(C) <object1> is beside <object3>
(D) <object1> is not visible in relation to <object3>

Ground Truth (A) <object1> is in front of <object3>

Baseline (B) <object1> is behind <object3>

(A) <object1> is in front of <object3>

Figure 17. **Qualitative Results**. This sample is drawn from VideoRefer-Bench^{Q}, designed to assess a model's performance on the task of **Mask-Referred Regional QA**. This sample presents a multi-masklet scenario, with two masklets referring to two different individuals. The boundary of the <object1> region is highlighted in purple, and <object2> is highlighted in green. The model trained on **Strefer**-generated data correctly answers this multi-masklet reference question by effectively analyzing the relationship between the two masklets within the video context.



How does the position of <object1><region> relate to <object2><region>? Pick the best option:
(A) <object1> is above <object2> on the tree
(B) <object1> is below <object2> on the tree
(C) <object1> and <object2> are on the same branch
(D) <object1> is on a different tree than <object2>

Ground Truth (A) <object1> is above <object2> on the tree

Baseline (B) <object1> is below <object2> on the tree

(A) <object1> is above <object2> on the tree

Figure 18. **Qualitative Results**. This sample is drawn from VideoRefer-Bench^{Q}, designed to assess a model's performance on the task of **Mask-Referred Regional QA**. This sample presents a multi-masklet scenario, with two masklets referring to two different individuals. The boundary of the <object1> region is highlighted in purple, and <object2> is highlighted in green. Kindly zoom in, as the regions are relatively small and may be difficult to discern. The model trained on **Strefer**-generated data correctly answers this multi-masklet reference question by effectively analyzing the relationship between the two masklets within the video context.

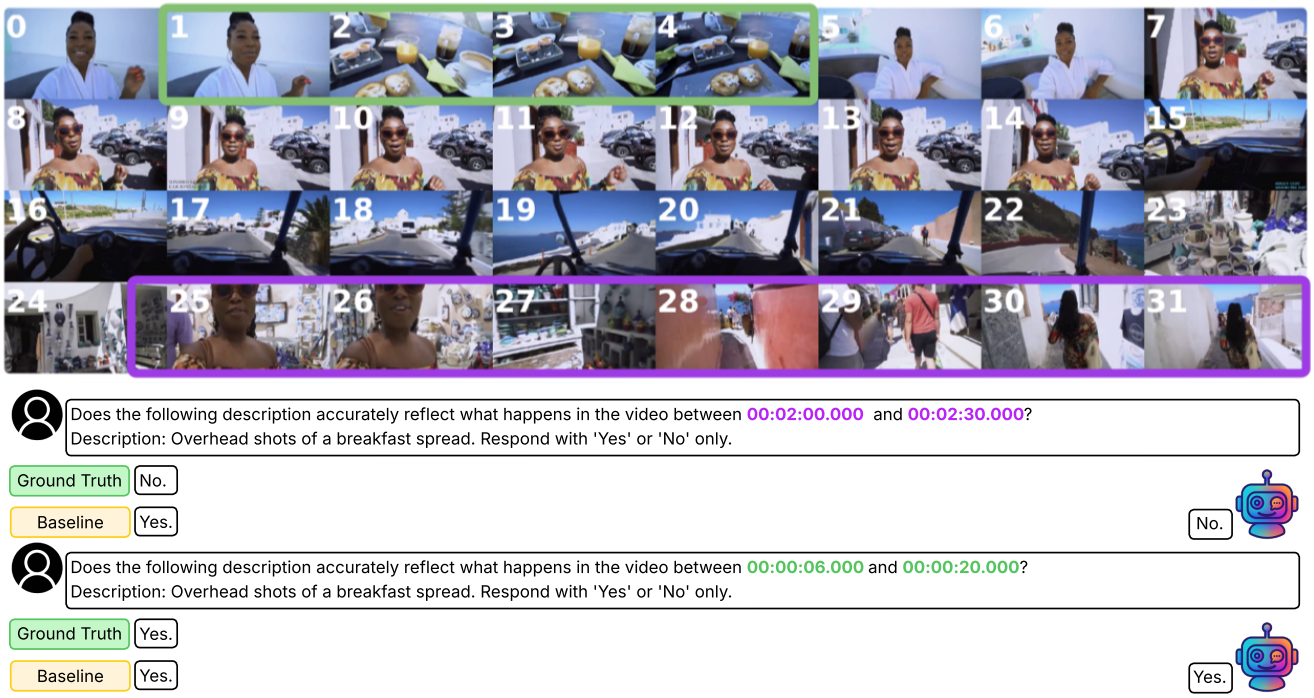


Figure 19. **Qualitative Results**. This sample is drawn from QVHighlights, using our repurposed task designed to assess a model's performance on **Timestamp-Referred Video QA**. The segment boundaries corresponding to the timestamps in the first and second questions are highlighted in purple and green, respectively. The model trained on our **Strefer**-generated data correctly answers both questions, demonstrating superior understanding of precise moments and segments in videos compared to the baseline.

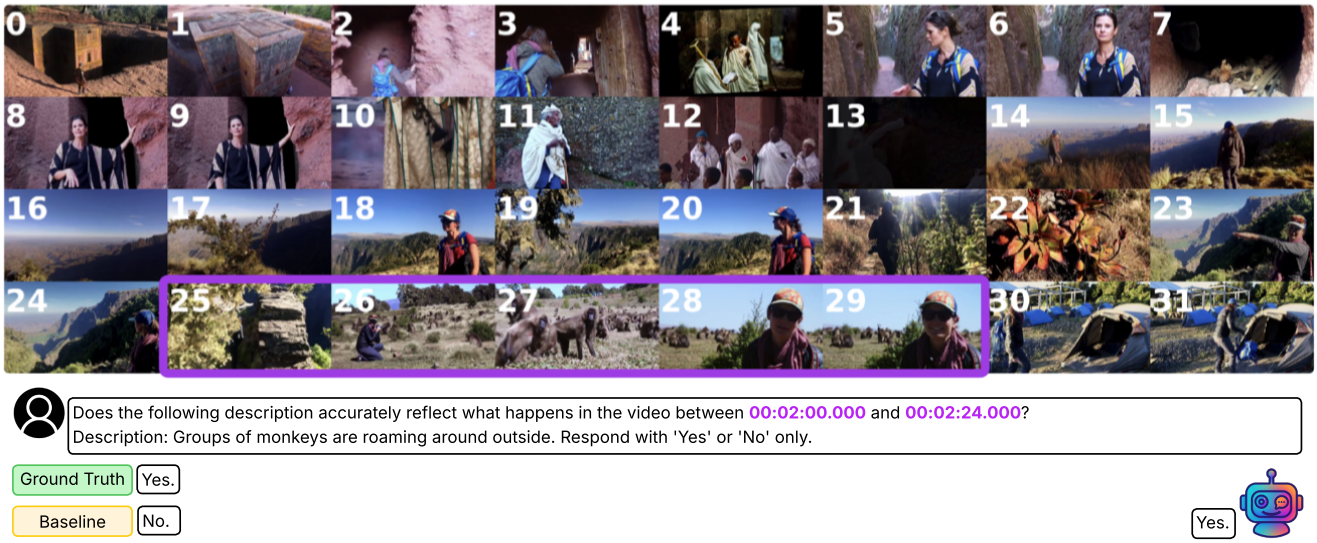
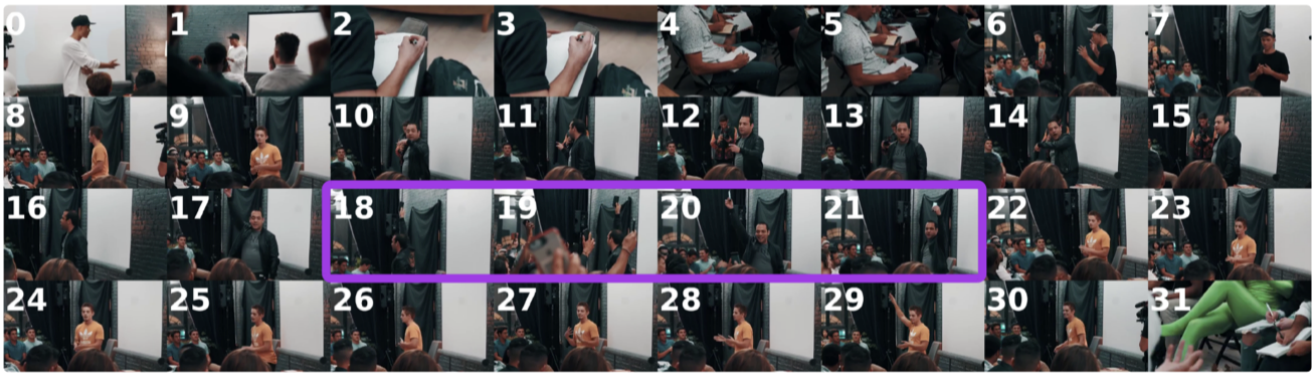


Figure 20. **Qualitative Results**. This sample is drawn from QVHighlights, using our repurposed task designed to assess a model's performance on **Timestamp-Referred Video QA**. The boundary of segment corresponding to the timestamps in the question is highlighted in purple. The model trained on our **Strefer**-generated data correctly answers the question, demonstrating superior understanding of precise moments and segments in videos compared to the baseline.



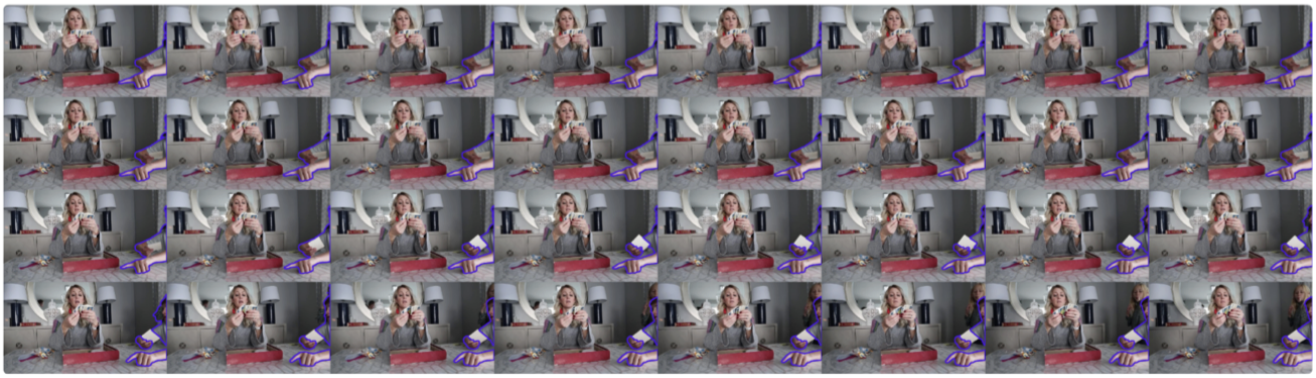
Does the following description accurately reflect what happens in the video between 00:01:28.000 and 00:01:42.000?
Description: People in a crowd during a meting raising their hands. Respond with 'Yes' or 'No' only.

Ground Truth: Yes.

Baseline: No.

Yes.

Figure 21. **Qualitative Results**. This sample is drawn from QVHighlights, using our repurposed task designed to assess a model's performance on **Timestamp-Referred Video QA**. The boundary of segment corresponding to the timestamps in the question is highlighted in purple. The model trained on our **Strefer**-generated data correctly answers the question, demonstrating superior understanding of precise moments and segments in videos compared to the baseline.



Please give a detailed description of the highlighted object <region> in the video.

Ground Truth: The boy in the video is wearing a light-colored shirt. He is sitting on the right side of a table and appears to be playing a card game. He holds a card in his right hand and points his left hand toward the woman across the table. His attention is focused on the game and he appears to be actively participating in the activity.

Baseline: A woman is sitting at a table, holding a deck of cards. She is wearing a gray sweater and has blonde hair. The table is white and has a marble pattern. There is a large mirror behind her, reflecting the room. The woman is shuffling the cards, and her hands are in motion as she deals them.

A person is sitting at a table, holding a deck of cards. The individual is wearing a gray sweater and has blonde hair. The person is engaged in a card game, with their hands actively manipulating the cards. The table is white and has a marble-like pattern. In the background, there is a large mirror reflecting the scene, and a lamp is visible on the left side of the table. The person's focus is on the cards, and they appear to be in the midst of a game, possibly a card game like "Cards Against Humanity" or a similar game. The person's movements are deliberate and focused, indicating their engagement in the game.

Figure 22. **Failure Results**. This sample is drawn from VideoRefer-Bench^{D}, designed to assess a model's performance on the task of **Mask-Referred Regional Description**. The boundary of the region referred to by the mask in this sample is highlighted in purple. The masklet is intended to refer to the boy on the right, but he is mostly out of view, while a woman appears prominently in the center of the video. Both the baseline model and the model trained on **Strefer**-generated data fail to correctly interpret the masklet.

Please give a detailed description of the highlighted object <region> in the video.

Ground Truth: The woman in the video is engaged in cooking activities in the kitchen, wearing white clothes and a blue hat, standing in front of the counter cutting vegetables.

Baseline: A woman in a black coat is standing in a kitchen, facing a woman in a white coat. She is engaged in cooking, with a pot on the stove in front of her. The woman in the black coat is focused on her task, while the woman in the white coat is also involved in the cooking process. The kitchen setting is well-lit, with various kitchen utensils and ingredients visible on the countertop. The woman in the black coat remains stationary, while the woman in the white coat occasionally moves around, possibly adjusting the stove or handling other cooking tasks.

A woman in a white sweater and blue hat is standing in a kitchen. She is engaged in cooking activities, using a frying pan on the stove. Her attire is casual, with a white sweater and a blue hat, and she appears focused on her task. The kitchen setting is well-lit, with various kitchen utensils and ingredients visible in the background. The woman's movements are deliberate as she stirs the contents of the frying pan, indicating her active involvement in the cooking process.

Figure 23. **Failure Results**. This sample is drawn from VideoRefer-Bench^{D}, designed to assess a model's performance on the task of **Mask-Referred Regional Description**. The boundary of the region referred to by the mask in this sample is highlighted in purple. While the model trained on **Strefer**-generated data correctly identifies that the masklet refers specifically to the woman in the white sweater, it incorrectly responds that her action is "frying pan".

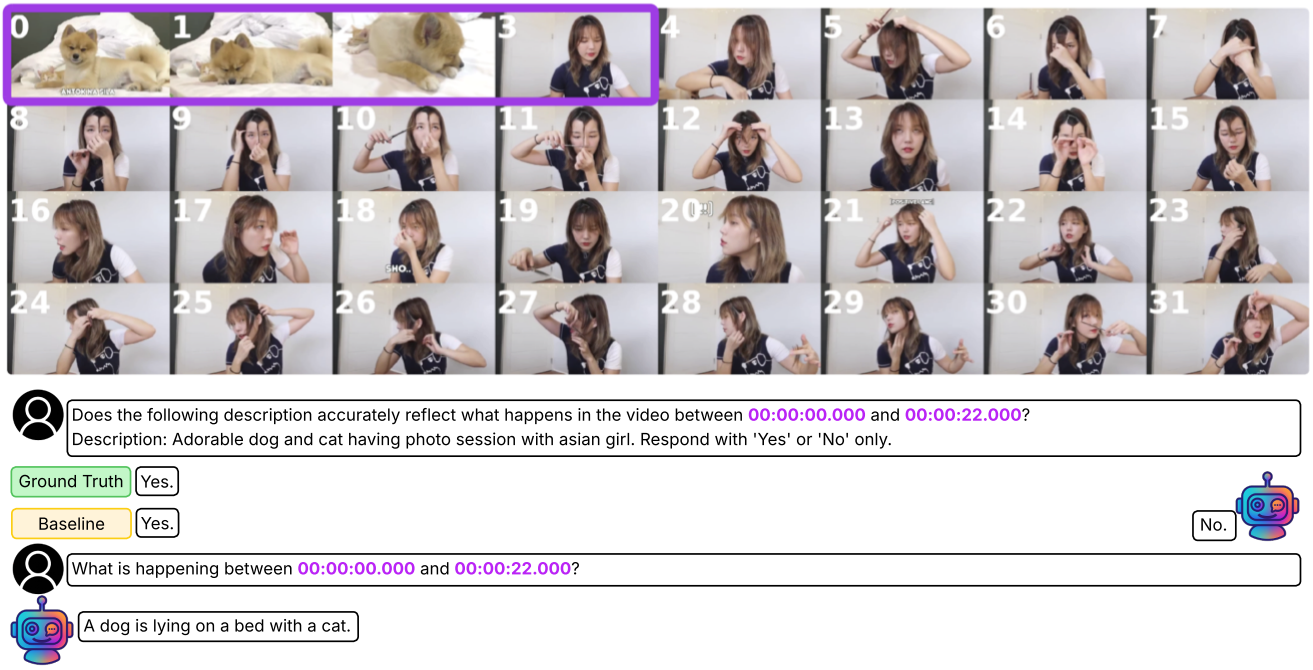


Figure 24. **Failure Results**. This sample is drawn from QVHighlights, using our repurposed task designed to assess a model's performance on **Timestamp-Referred Video QA**. The boundary of segment corresponding to the timestamps in the question is highlighted in purple. Although the model trained on our **Strefer**-generated data fails to answer the question correctly, it does accurately recognize that the segment shows a dog lying on a bed with a cat. We suspect the model's failure stems from its disagreement with the description, which is not fully grounded in the visual content—for example, the video segment does not clearly depict a photo session involving the dog, the cat, and the girl.

Type ID & Task	Frames	Source	Format	Example Question-Answer Pair	Mask-Refer Version
1. Ask the model to describe the behavior of entities that are present in a segment of the video.	Frames are extracted only from the segment of the video.	Template	OE	Question: <video>What is happening to the woman? Answer: The woman is engaged in a dance with the man, involving spins and turns. She is lifted off the ground by the man during the dance.	Question:<video>Please answer the following question about the <region>. What is happening to her? Answer: The woman is engaged in a dance with the man, involving spins and turns. She is lifted off the ground by the man during the dance.
2. Ask the model to describe the behavior of entities that are not present in a segment of the video; the model should respond with uncertainty (e.g., "Sorry, I'm not sure").	Frames are extracted only from the segment of the video.	Template	OE	Question: <video>What is currently happening to the person in a green hoodie? Answer: The person in a green hoodie seems to be not clearly visible.	N/A
3. Ask a yes/no question about the presence of an entity in a segment of the video; if present, the model should describe its behavior; if absent, the model should respond with uncertainty.	Frames are extracted only from the segment of the video.	Template	OE	Question: <video>Were you able to see a woman in a black jacket? Answer: Yes. The woman walks towards the child seated on the sofa.	N/A
4. Ask a yes/no question about the presence of an entity in a segment of the video; the model should respond with a concise "Yes" or "No" only.	Frames are extracted only from the segment of the video.	Template	OE	Question:<video> Is there a woman in a black jacket? Answer only "Yes" or "No". Answer: Yes.	N/A
5. Ask the model to identify the correct temporal order in which entities first appear in the video from multiple choices.	Frames are extracted from the full video.	Template	MCQ	Question:<video> Which order shows their first appearance in the video? (A) child interacting with the plant bed, child holding a bag and a toy, child walking across the lawn (B) child holding a bag and a toy, child approaching a plant bed, child interacting with the plant bed (C) child approaching a plant bed, child holding a bag and a toy, child interacting with the plant bed (D) child walking across the lawn, child holding a bag and a toy, child interacting with the plant bed Answer: (B)	N/A
6. Ask the model to describe the behavior of entities that may or may not be present in a specific time range of the video; the question refers to a time range.	Frames are extracted from the full video.	Template	OE	Question:<video> Could you explain what the girl in the yellow coat is doing between 00:00:05 and 00:00:12.210? Answer: The girl in the yellow coat is carefully watering plants in a garden.	Question:<video>Please answer the following question about the <region>. Could you explain what she is doing between 00:00:05 and 00:00:12.210? Answer: The girl in the yellow coat is carefully watering plants in a garden.
7. Ask the model to describe what happened generally or to a specific entity during a specific time range in the video; the question refers to a time range.	Frames are extracted from the full video.	LLM	OE & MCQ	Question:<video> What else did the woman interviewing the man do between 00:00:00 and 00:00:07.007? Answer: The woman interviewing the man is talking as well.	Question:<video>Please answer the following question about the <region>. What else did she do between 00:00:00 and 00:00:07.007? Answer: The woman interviewing the man is talking as well.
8. Ask the model to describe what happened generally or to a specific entity during a coarse time range in the video (e.g., throughout the video, beginning, middle, or end).	Frames are extracted from the full video.	LLM	OE & MCQ	Question:<video> What else did the woman interviewing the man do in the beginning of the video? Answer: The woman interviewing the man is talking as well.	Question:<video>Please answer the following question about the <region>. What else did she do in the beginning of the video? Answer: The woman interviewing the man is talking as well.
9. Ask the model to identify when a specific behavior or event occurs within the video; expect the model to answer with a coarse time range in the video (e.g., throughout the video, beginning, middle, or end).	Frames are extracted from the full video.	LLM	OE & MCQ	Question:<video> During which part of the video was the child in pink dress riding a tricycle? Answer: The beginning.	Question:<video>Please answer the following question about the <region>. During which part of the video was this person riding a tricycle? Answer: The beginning.
10. Ask the model to describe the behavior of an entity before/during/after something else occurs.	Frames are extracted from the full video.	LLM	OE & MCQ	Question:<video> What is the adult doing while the child is riding a tricycle? Answer: The adult is watching and walking behind the child.	Question:<video>Please answer the following question about the <region>. What is he doing while the child is riding a tricycle? Answer: The adult is watching and walking behind the child. -
11. Ask the model to identify the entity involved before/during/after something else occurs.	Frames are extracted from the full video.	LLM	OE & MCQ	Question: Who is walking behind the child in blue while the child is riding a tricycle? Answer: An adult wearing a black shirt.	Question:<video>Please answer the following question about the <region>. Who is walking behind this child while the child is riding a tricycle? Answer: An adult wearing a black shirt.

Table 7. **Details of Strefer-synthesized video instruction data**. The table details the question task types, their visual inputs, QA generation sources, formats, examples, and the mask-referring versions of the QAs. 'OE' denotes open-ended QA, and 'MCQ' indicates multiple-choice QA.

References

[1] Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024. 15

[2] Ali Athar, Xueqing Deng, and Liang-Chieh Chen. Vicas: A dataset for combining holistic and pixel-level video understanding using captions with grounded segmentation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 19023–19035, 2025. 2

[3] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 2

[4] Brandon Castellano and contributors. PySceneDetect: Video Scene Cut Detection. https://www.scenedetect.com/, 2025. Version 0.6.6 (released March 9, 2025). 5

[5] Lin Chen, Xilin Wei, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Zhenyu Tang, Li Yuan, et al. Sharegpt4video: Improving video understanding and generation with better captions. *Advances in Neural Information Processing Systems*, 37:19472–19495, 2024. 3, 14

[6] Tsai-Shien Chen, Aliaksandr Siarohin, Willi Menapace, Ekaterina Deyneka, Hsiang-wei Chao, Byung Eun Jeon, Yuwei Fang, Hsin-Ying Lee, Jian Ren, Ming-Hsuan Yang, et al. Panda-70m: Captioning 70m videos with multiple cross-modality teachers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13320–13331, 2024. 13

[7] Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024. 2

[8] Zesen Cheng, Sicong Leng, Hang Zhang, Yifei Xin, Xin Li, Guanzheng Chen, Yongxin Zhu, Wenqi Zhang, Ziyang Luo, Deli Zhao, et al. Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms. *arXiv preprint arXiv:2406.07476*, 2024. 2

[9] Jang Hyun Cho, Andrea Madotto, Effrosyni Mavroudi, Triantafyllos Afouras, Tushar Nagarajan, Muhammad Maaz, Yale Song, Tengyu Ma, Shuming Hu, Suyog Jain, et al. Perceptionlm: Open-access data and models for detailed visual understanding. *arXiv preprint arXiv:2504.13180*, 2025. 2

[10] Jihoon Chung, Tyler Zhu, Max Gonzalez Saez-Diez, Juan Carlos Niebles, Honglu Zhou, and Olga Russakovsky. Unifying specialized visual encoders for video language models. *arXiv preprint arXiv:2501.01426*, 2025. 2

[11] DAMO-NLP-SG. Videorefer benchmark evaluation for general mllms. https://github.com/DAMO-NLP-SG/VideoRefer/blob/main/benchmark/evaluation_general_mllms.md, 2024. Accessed: 2025-06-29. 13, 14

[12] Henghui Ding, Chang Liu, Shuting He, Xudong Jiang, and Chen Change Loy. Mevis: A large-scale benchmark for video segmentation with motion expressions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 13

[13] Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 24108–24118, 2025. 6, 8

[14] Kirill Gavrilyuk, Amir Ghodrati, Zhenyang Li, and Cees G. M. Snoek. Actor and action video segmentation from a sentence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 13

[15] Yongxin Guo, Jingyu Liu, Mingda Li, Dingxin Cheng, Xiaoying Tang, Dianbo Sui, Qingbin Liu, Xi Chen, and Kevin Zhao. Vtg-llm: Integrating timestamp knowledge into video llms for enhanced video temporal grounding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3302–3310, 2025. 3

[16] Miran Heo, Min-Hung Chen, De-An Huang, Sifei Liu, Subhashree Radhakrishnan, Seon Joo Kim, Yu-Chiang Frank Wang, and Ryo Hachiuma. Omni-rgpt: Unifying image and video region-level understanding via token marks. *arXiv preprint arXiv:2501.08326*, 2025. 2, 3

[17] Wenyi Hong, Weihan Wang, Ming Ding, Wenmeng Yu, Qingsong Lv, Yan Wang, Yean Cheng, Shiyu Huang, Junhui Ji, Zhao Xue, et al. Cogvlm2: Visual language models for image and video understanding. *arXiv preprint arXiv:2408.16500*, 2024. 2

[18] Bin Huang, Xin Wang, Hong Chen, Zihan Song, and Wenwu Zhu. Vtimellm: Empower llm to grasp video moments. In *CVPR*, 2024. 3

[19] Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. Tgif-qa: Toward spatio-temporal reasoning in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2758–2766, 2017. 12

[20] Qing Jiang, Lin Wu, Zhaoyang Zeng, Tianhe Ren, Yuda Xiong, Yihao Chen, Qin Liu, and Lei Zhang. Referring to any person, 2025. 2, 4, 5

[21] Xuan Ju, Yiming Gao, Zhaoyang Zhang, Ziyang Yuan, Xintao Wang, Ailing Zeng, Yu Xiong, Qiang Xu, and Ying Shan. Miradata: A large-scale video dataset with long durations and structured captions. *Advances in Neural Information Processing Systems*, 37:48955–48970, 2024. 3, 12

[22] Evangelos Kazakos, Cordelia Schmid, and Josef Sivic. Large-scale pre-training for grounded video caption generation. *arXiv preprint arXiv:2503.10781*, 2025. 10

[23] Jie Lei, Tamara L Berg, and Mohit Bansal. Detecting moments and highlights in videos via natural language queries. *Advances in Neural Information Processing Systems*, 34: 11846–11858, 2021. 6, 8, 13, 14

[24] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024. 2

[25] KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. Videochat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355*, 2023. 2, 3

[26] Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. Mvbench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22195–22206, 2024. 2, 3

[27] Lei Li, Yuanxin Liu, Linli Yao, Peiyuan Zhang, Chenxin An, Lean Wang, Xu Sun, Lingpeng Kong, and Qi Liu. Temporal reasoning transfer from text to video. In *ICLR 2025*. OpenReview.net, 2025. 3

[28] Yanwei Li, Chengyao Wang, and Jiaya Jia. Llama-vid: An image is worth 2 tokens in large language models. 2024. 2

[29] Long Lian, Yifan Ding, Yunhao Ge, Sifei Liu, Hanzi Mao, Boyi Li, Marco Pavone, Ming-Yu Liu, Trevor Darrell, Adam Yala, et al. Describe anything: Detailed localized image and video captioning. *arXiv preprint arXiv:2504.16072*, 2025. 3

[30] Jingyang Lin, Jialian Wu, Ximeng Sun, Ze Wang, Jiang Liu, Yusheng Su, Xiaodong Yu, Hao Chen, Jiebo Luo, Zicheng Liu, et al. Unleashing hour-scale video training for long video-language understanding. *arXiv preprint arXiv:2506.05332*, 2025. 3

[31] Weifeng Lin, Xinyu Wei, Ruichuan An, Tianhe Ren, Tingwei Chen, Renrui Zhang, Ziyu Guo, Wentao Zhang, Lei Zhang, and Hongsheng Li. Perceive anything: Recognize, explain, caption, and segment anything in images and videos. *arXiv preprint arXiv:2506.05302*, 2025. 3

[32] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023. 2, 4, 5

[33] Yuanxin Liu, Shicheng Li, Yi Liu, Yuxiang Wang, Shuhuai Ren, Lei Li, Sishuo Chen, Xu Sun, and Lu Hou. Tempcompass: Do video llms really understand videos? *arXiv preprint arXiv:2403.00476*, 2024. 2, 3, 8

[34] Chuofan Ma, Yi Jiang, Jiannan Wu, Zehuan Yuan, and Xiaojuan Qi. Groma: Localized visual tokenization for grounding multimodal large language models. In *European Conference on Computer Vision*, pages 417–435. Springer, 2024. 2

[35] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. *arXiv preprint arXiv:2306.05424*, 2023. 2, 3, 12

[36] Arjun Mani, Nobline Yoo, Will Hinthorn, and Olga Russakovsky. Point and ask: Incorporating pointing into visual question answering. *arXiv preprint arXiv:2011.13681*, 2020. 2

[37] Shehan Munasinghe, Rusiru Thushara, Muhammad Maaz, Hanoona Abdul Rasheed, Salman Khan, Mubarak Shah, and Fahad Khan. Pg-video-llava: Pixel grounding large video-language models. *arXiv preprint arXiv:2311.13435*, 2023. 10

[38] Shehan Munasinghe, Hanan Gani, Wenqi Zhu, Jiale Cao, Eric Xing, Fahad Shahbaz Khan, and Salman Khan. Videoglamm: A large multimodal model for pixel-level visual grounding in videos. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 19036–19046, 2025. 2

[39] Long Qian, Juncheng Li, Yu Wu, Yaobo Ye, Hao Fei, Tat-Seng Chua, Yueting Zhuang, and Siliang Tang. Momentor: Advancing video large language model with fine-grained temporal reasoning. *arXiv preprint arXiv:2402.11435*, 2024. 3

[40] Jihao Qiu, Yuan Zhang, Xi Tang, Lingxi Xie, Tianren Ma, Pengyu Yan, David Doermann, Qixiang Ye, and Yunjie Tian. Artemis: Towards referential understanding in complex videos. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 2

[41] Hanoona Rasheed, Muhammad Maaz, Sahal Shaji, Abdelrahman Shaker, Salman Khan, Hisham Cholakkal, Rao M Anwer, Eric Xing, Ming-Hsuan Yang, and Fahad S Khan. Glamm: Pixel grounding large multimodal model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13009–13018, 2024. 2

[42] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 4, 5

[43] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 2, 9, 11

[44] Tianhe Ren, Shuo Shen, et al. Grounded sam 2: Ground and track anything in videos with grounding dino, florence-2 and sam 2. GitHub repository, 2025. https://github.com/IDEA-Research/Grounded-SAM-2. 2, 4, 7, 18, 19, 20, 21

[45] Michael S Ryoo, Honglu Zhou, Shrikant Kendre, Can Qin, Le Xue, Manli Shu, Jongwoo Park, Kanchana Ranasinghe, Silvio Savarese, Ran Xu, et al. xgen-mm-vid (blip-3-video): You only need 32 tokens to represent a video even in vlms. *arXiv preprint arXiv:2410.16267*, 2024. 2, 6, 8, 10, 11, 12, 14

[46] Eli Schwartz, Leshem Choshen, Joseph Shtok, Sivan Doveh, Leonid Karlinsky, and Assaf Arbelle. Numerologic: Number encoding for enhanced llms' numerical reasoning. *arXiv preprint arXiv:2404.00459*, 2024. 12

[47] Ye Sun, Hao Zhang, Henghui Ding, Tiehua Zhang, Xingjun Ma, and Yu-Gang Jiang. Sama: Towards multi-turn referential grounded video chat with large language models. *arXiv preprint arXiv:2505.18812*, 2025. 2, 3

[48] Qwen Team. Qwen2.5: A party of foundation models, 2024. 5

[49] Yuli Vasiliev. *Natural language processing with Python and spaCy: A practical introduction*. No Starch Press, 2020. 4

[50] Haibo Wang, Zhiyang Xu, Yu Cheng, Shizhe Diao, Yufan Zhou, Yixin Cao, Qifan Wang, Weifeng Ge, and Lifu

Huang. Grounded-videollm: Sharpening fine-grained temporal grounding in video large language models. *arXiv preprint arXiv:2410.03290*, 2024. 3, 6, 10, 12

[51] Han Wang, Yongjie Ye, Yanjie Wang, Yuxiang Nie, and Can Huang. Elysium: Exploring object-level perception in videos via mllm. In *European Conference on Computer Vision*, pages 166–185. Springer, 2024. 2

[52] Jiawei Wang, Liping Yuan, Yuchen Zhang, and Haomiao Sun. Tarsier: Recipes for training and evaluating large video description models, 2024. 2, 5

[53] Jiannan Wu, Yi Jiang, Peize Sun, Zehuan Yuan, and Ping Luo. Language as queries for referring video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. Also explores Ref-Youtube-VOS, Ref-DAVIS17, A2D-Sentences, JHMDB-Sentences. 13

[54] Rujie Wu, Xiaojian Ma, Hai Ci, Yue Fan, Yuxuan Wang, Haozhe Zhao, Qing Li, and Yizhou Wang. Longvitu: Instruction tuning for long-form video understanding. *arXiv preprint arXiv:2501.05037*, 2025. 3

[55] Yongliang Wu, Xinting Hu, Yuyang Sun, Yizhou Zhou, Wenbo Zhu, Fengyun Rao, Bernt Schiele, and Xu Yang. Number it: Temporal grounding videos like flipping manga. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 13754–13765, 2025. 3, 6, 13, 14

[56] Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. Next-qa: Next phase of question-answering to explaining temporal actions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9777–9786, 2021. 5, 12

[57] Dejing Xu, Zhou Zhao, Jun Xiao, Fei Wu, Hanwang Zhang, Xiangnan He, and Yueting Zhuang. Video question answering via gradually refined attention over appearance and motion. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1645–1653, 2017. 12

[58] Jiarui Xu, Xingyi Zhou, Shen Yan, Xiuye Gu, Anurag Arnab, Chen Sun, Xiaolong Wang, and Cordelia Schmid. Pixel-aligned language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13030–13039, 2024. 2

[59] Lin Xu, Yilin Zhao, Daquan Zhou, Zhijie Lin, See Kiong Ng, and Jiashi Feng. Pllava: Parameter-free llava extension from images to videos for video dense captioning. *arXiv preprint arXiv:2404.16994*, 2024. 2

[60] Mingze Xu, Mingfei Gao, Zhe Gan, Hong-You Chen, Zhengfeng Lai, Haiming Gang, Kai Kang, and Afshin Dehghan. Slowfast-llava: A strong training-free baseline for video large language models. *arXiv preprint arXiv:2407.15841*, 2024. 2

[61] Le Xue, Manli Shu, Anas Awadalla, Jun Wang, An Yan, Senthil Purushwalkam, Honglu Zhou, Viraj Prabhu, Yutong Dai, Michael S Ryoo, et al. xgen-mm (blip-3): A family of open large multimodal models. *arXiv preprint arXiv:2408.08872*, 2024. 12

[62] An Yan, Zhengyuan Yang, Junda Wu, Wanrong Zhu, Jianwei Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Julian McAuley, Jianfeng Gao, et al. List items one by one: A new data source and learning paradigm for multimodal llms. *arXiv preprint arXiv:2404.16375*, 2024. 2

[63] Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao. Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v. *arXiv preprint arXiv:2310.11441*, 2023. 6, 13, 14

[64] Haoxuan You, Haotian Zhang, Zhe Gan, Xianzhi Du, Bowen Zhang, Zirui Wang, Liangliang Cao, Shih-Fu Chang, and Yinfei Yang. Ferret: Refer and ground anything anywhere at any granularity. *arXiv preprint arXiv:2310.07704*, 2023. 2

[65] En Yu, Liang Zhao, Yana Wei, Jinrong Yang, Dongming Wu, Lingyu Kong, Haoran Wei, Tiancai Wang, Zheng Ge, Xiangyu Zhang, et al. Merlin: Empowering multimodal llms with foresight minds. In *European Conference on Computer Vision*, pages 425–443. Springer, 2024. 2

[66] Zhou Yu, Dejing Xu, Jun Yu, Ting Yu, Zhou Zhao, Yueting Zhuang, and Dacheng Tao. Activitynet-qa: A dataset for understanding complex web videos via question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 9127–9134, 2019. 12

[67] Yuqian Yuan, Wentong Li, Jian Liu, Dongqi Tang, Xinjie Luo, Chi Qin, Lei Zhang, and Jianke Zhu. Osprey: Pixel understanding with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28202–28211, 2024. 2

[68] Yuqian Yuan, Hang Zhang, Wentong Li, Zesen Cheng, Boqiang Zhang, Long Li, Xin Li, Deli Zhao, Wenqiao Zhang, Yueting Zhuang, et al. Videorefer suite: Advancing spatial-temporal object understanding with video llm. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 18970–18980, 2025. 3, 6, 7, 8, 10, 12, 13, 14

[69] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11975–11986, 2023. 5

[70] Boqiang Zhang, Kehan Li, Zesen Cheng, Zhiqiang Hu, Yuqian Yuan, Guanzheng Chen, Sicong Leng, Yuming Jiang, Hang Zhang, Xin Li, et al. Videollama 3: Frontier multimodal foundation models for image and video understanding. *arXiv preprint arXiv:2501.13106*, 2025. 2

[71] Hao Zhang, Hongyang Li, Feng Li, Tianhe Ren, Xueyan Zou, Shilong Liu, Shijia Huang, Jianfeng Gao, Leizhang, Chunyuan Li, et al. Llava-grounding: Grounded visual chat with large multimodal models. In *European Conference on Computer Vision*, pages 19–35. Springer, 2024. 2

[72] Haotian Zhang, Haoxuan You, Philipp Dufter, Bowen Zhang, Chen Chen, Hong-You Chen, Tsu-Jui Fu, William Yang Wang, Shih-Fu Chang, Zhe Gan, et al. Ferret-v2: An improved baseline for referring and grounding with large language models. *arXiv preprint arXiv:2404.07973*, 2024.

[73] Shilong Zhang, Peize Sun, Shoufa Chen, Min Xiao, Wenqi Shao, Wenwei Zhang, Yu Liu, Kai Chen, and Ping Luo. Gpt4roi: Instruction tuning large language model on region-of-interest. In *European Conference on Computer Vision*, pages 52–70. Springer, 2025. 2

[74] Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Video instruction tuning with synthetic data, 2024. 3, 14

[75] Henghao Zhao, Ge-Peng Ji, Rui Yan, Huan Xiong, and Zechao Li. Videoexpert: Augmented llm for temporal-sensitive video understanding. *arXiv preprint arXiv:2504.07519*, 2025. 3