GLUS: Global-Local Reasoning Unified into
A Single Large Language Model for Video Segmentation

Lang Lin*   Xueyang Yu*   Ziqi Pang*   Yu-Xiong Wang
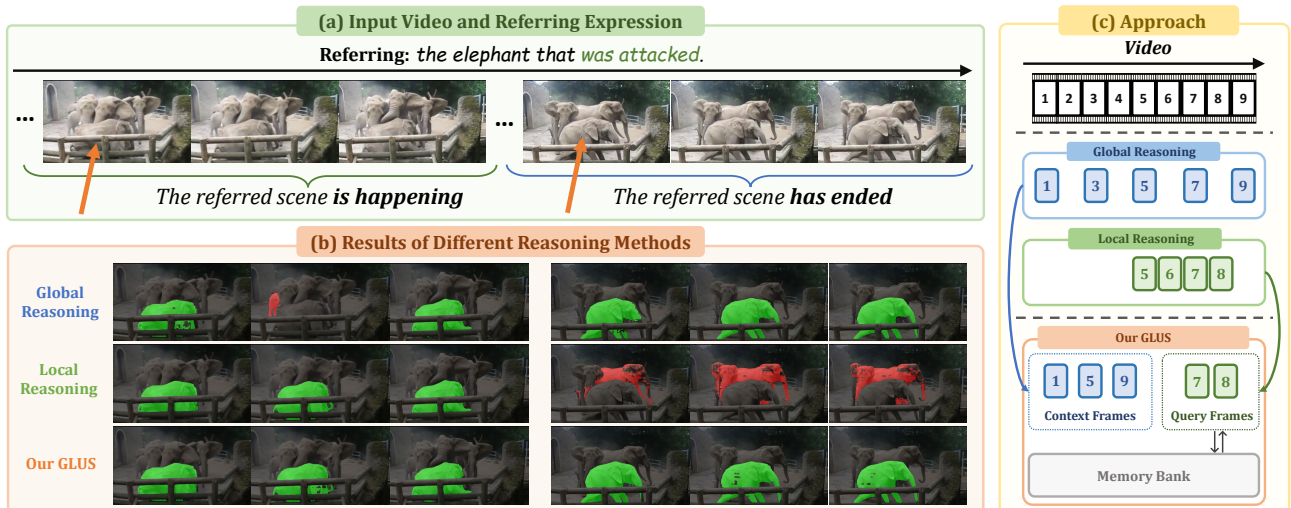University of Illinois Urbana-Champaign

Figure 1. Overview of **GLUS**. **(a)** We present a representative RefVOS example where the referring expression only covers part of the frames (the orange arrows point to the ground truth). Therefore, RefVOS requires both *global reasoning* for finding the target objects in frames without the referred motions and *local reasoning* for predicting temporally consistent masks. **(b)** We show green and red masks for correct and wrong segmentation. Existing multimodal large language models (MLLMs) focus on either global or local reasoning. However, global-only shows fluctuation in local "VOS," while local-only "Ref" to wrong objects without overall video contexts when the referring is not happening. Instead, our unified global-local reasoning shows consistent and correct predictions. **(c)** GLUS provides a simple yet effective baseline that unifies global and local reasoning through both global and local frame sampling and end-to-end memory modules.

Abstract

*This paper proposes a novel framework utilizing multimodal large language models (MLLMs) for referring video object segmentation (RefVOS). Previous MLLM-based methods commonly struggle with the dilemma between "Ref" and "VOS": they either specialize in understanding a few key frames (global reasoning) or tracking objects on continuous frames (local reasoning), and rely on external VOS or frame selectors to mitigate the other end of the challenge. However, our framework **GLUS** shows that **G**lobal and **L**ocal consistency can be **U**nified into a single video **S**egmentation MLLM: a set of sparse "context frames" provides global information, while a stream of continuous "query frames" conducts local object tracking. This is further supported by jointly training the MLLM with a pre-trained VOS memory bank to simultaneously digest short-range and long-range temporal information. To improve the information efficiency within the limited context window of MLLMs, we introduce object contrastive learning to distinguish hard false-positive objects and a self-refined framework to identify crucial frames and perform propagation. By collectively integrating these insights, our **GLUS** delivers a simple yet effective baseline, achieving new state-of-the-art for MLLMs on the MeViS and Ref-Youtube-VOS benchmark. Our project page is at https://glus-video.github.io/.*

1. Introduction

Using language expressions to specify target objects requires joint reasoning of visual contents and language intentions. Such capabilities in videos are recently studied via the task of referring video object segmentation (RefVOS),

---

*Equal Contribution.

which requires the models to localize and consistently track target object(s) according to language descriptions, such as "*the elephant that was attacked*" in Fig. 1(a). The unique challenge of RefVOS is that the described action may only be observable in a subset of frames. Therefore, the models must understand the global characteristics of each object to reliably propagate such reasoning across the whole video.

Motivated by the reasoning capabilities of multi-modal large language models (MLLMs) in referring image segmentation (RIS), *e.g.*, LISA [19], RefVOS studies [3, 45, 52] adapt these MLLMs to videos, hoping to reason the variety of language expressions effectively. These approaches commonly consider MLLM as a *multi-image* framework and reason a limited number of N frames. Consequently, they struggle to handle the entire video, especially with their context window N significantly fewer than the video frames to process. These MLLMs then face the shared "Ref" and "VOS" dilemma in RefVOS: using the N frames to either understand the *global* context or address *local* temporal continuity. Focusing solely on either global or local aspects results in inadequate performance for complex RefVOS scenarios (as in Fig. 1(b)). To satisfy both aspects, they resort to *external* modules like off-the-shelf video object segmentation (VOS) models, which unavoidably increases the system's complexity. Instead, we aim to demonstrate a framework **GLUS** where *a single MLLM alone has the ability of Global-Local Unified reasoning for video Segmentation*.

Our GLUS begins from a simple yet principled adaptation to MLLM by *providing* and *only providing* the necessary information for *global* and *local* reasoning. For global reasoning, the MLLM must have a set of frames covering the whole video to maximize the probability of observing the referred property. For local reasoning, we realize that MLLMs are natively compatible with the VOS formulation, because the current predictions are always based on the precedent frames. Therefore, our GLUS divides the N frames for an MLLM into two groups (as in Fig. 1(c)): **(1) Context frames**: several frames uniformly sampled from the video to cover the global reasoning context; **(2) Query frames**: the frames to produce segmentation results, which are continuously sampled to support temporal continuity naturally. Such a procedure also intuitively mimics the behaviors of a human to address video segmentation: first, check a few sparsely sampled frames (*i.e.*, context frames) to understand the video; then, drawing the masks of an object on frames iteratively (*i.e.*, query frames).

Compared with global-only and local-only strategies, GLUS divides total N frames into two types and inevitably reduces both global and local video information obtained from the video. To tackle the issue, we seek to break the limits of an MLLM's context window size N by introducing a memory bank that can store information from previous predictions, which enhances both local and global reasoning. Since our query frames are continuous, we discover that GLUS can be trained and inferred jointly with the memory module from a pre-trained foundational VOS model, *e.g.*, SAM2 [32]. By creatively unifying such an "*online*" VOS module and optimizing it end-to-end with an MLLM, our GLUS offers a more straightforward system without calling external VOS models.

Enabled by our framework capturing global and local contexts, which provides more comprehensive temporal contexts than conventional approaches, we can better explore *distinguishing the target information in a fine-grained way*. From the frame level, the prediction quality of GLUS can reflect the relevance between a frame and referring expression, and create useful pseudo-labels for video MLLMs to understand the notion of "key frames." Contrasting the previous works[45] using *off-the-shelf* key frame selectors, our fine-tuning enables better contextual information for our global-local reasoning. The selector trained from GLUS is further used for internal propagation which leads to a self-refinement framework. From the object level, GLUS enhances the fine-grained matching between objects and language instructions via contrastive losses, maximizing the distance of tokens referring to different objects.

In conclusion, our contributions in GLUS are:

1. We demonstrate that unifying global and local reasoning into a *single* MLLM for RefVOS through the design of *context* and *query* frames constitutes a simple yet effective baseline method for MLLM-based RefVOS models.
2. We illustrate *end-to-end training of a MLLM with pre-trained VOS memory modules* and decouple the needs for calling external VOS models.
3. We introduce plug-and-play self-refinement with key frame selectors and object contrastive loss distinguishing hard false-positive objects.

Our simple yet effective GLUS serve as a strong MLLM-based RefVOS baseline, demonstrating advantages on the major RefVOS benchmarks MeViS [9] and Ref-Youtube-VOS [34] over previous MLLMs.

2. Related Work

Referring Video Object Segmentation. Referring video object segmentation (RefVOS) aims to segment the target object within a video based on a given language expression. A recent benchmark, MeViS [9], introduces complex multi-object scenarios with extensive motion dynamics, posing greater challenges to the community. Common practices [4, 25, 35, 42, 43, 50] utilize language queries to attend to the referred object. Some recent works [9, 13] propose motion aggregation to capture motion information. Besides, recent works [3, 27, 45, 52, 54] use multimodal large language models (MLLMs) to reason over complex expressions. In this work, we propose a novel MLLM-based framework that unifies both global and local reasoning and

| | Training | | Inference | | $\langle\text{SEG}\rangle$ | External VOS |
	Sampled Frames	Information	Sampled Frames	Information		
VideoLISA [3]	Uniform	Global	Uniform	Global	1	✓
VISA [45]	Random	Random	Uniform + Continuous	Global + Local	1	✓
ViLLa [52]	Continuous	Local	Continuous	Local	N	✗
GLUS (Ours)	Uniform + Continuous	Global + Local	Uniform + Continuous	Global + Local	N	✗

Table 1. Comparison of *frame* and *information* utilized in existing RefVOS MLLMs. Despite the distinct challenges of "Ref" (global reasoning) and "VOS" (local reasoning) in RefVOS, previous methods fail to unify them and offer inconsistent strategies for training and inference. Compared with previous methods, our GLUS proposes the "context + query frames" strategy (Sec. 4) that effectively unifies global and local reasoning for both training and inference stages.

decouples the need of motion aggregation modules.

Large Language Models in Segmentation and Grounding. Inspired by the impressive reasoning capabilities of large language models (LLMs), researchers are seeking to equip LLMs with fine-grained understanding in visual tasks [19, 26, 28, 33, 51]. LISA [19] pioneered such approach by connecting a multi-modal LLM (MLLM) with the Segment Anything Model [18] via a special token to produce accurate segmentation masks. Some recent works extend LISA [19] to the video domain. TrackGPT [54] makes a straightforward adaptation by iteratively updating the special token as the video progresses. VISA [45] further provides global context for producing a special token, while ViLLa [52] designs a context-aggregation module to derive more meaningful visual features. VideoLISA [3] uses a single token for tracking throughout the video. In this work, we introduce an effective MLLM baseline to endow the model with better motion understanding abilities without additional VOS modules needed.

Video Object Segmentation. Different from RefVOS, VOS is an *online* task: the target object is marked with a ground truth mask in the first frame, and the VOS models function in a streaming way to track this object. The memory bank, which stores historical information, is the core module that enables the temporal reasoning abilities of VOS. Existing methods commonly leverage pixel-level [6, 16, 37, 47–49] or object-level [1, 2, 8, 21] matching via attention mechanisms [36] when using memory banks. In addition to interacting with memory banks, VOS studies [7, 32, 46, 53] also discover the importance of organizing the memory banks properly. In the context of RefVOS, our framework GLUS supports the benefits of training a pre-trained memory bank end-to-end with a RefVOS MLLM, including both its memory reading attention and organization. This enables the MLLM to reason with information beyond its own context window and acquire the ability of consistent object tracking.

3. Preliminaries

3.1. Referring Video Object Segmentation

Given an input video consisting of T frames $I_{1:T} \in \mathbb{R}^{T \times H \times W \times 3}$ and a referring language expression R, Re-

fVOS aims to build a model ϕ_θ predicting binary segmentation masks $\mathcal{M}_{1:T}$ of the referred object:

$$M_{1:T} = \phi_\theta(I_{1:T}, R) \tag{1}$$

RefVOS differs from both VOS and referring image segmentation (RIS) tasks. Compared with VOS, RefVOS is an offline task. It exhibits the critical challenge of localizing the target object and action from the whole video, where only a short key video clip might correspond to the language expression (as Fig. 2). Compared with RIS, RefVOS requires global video reasoning and temporal coherence with consistent objects across numerous frames.

3.2. LISA: MLLM for Segmentation

As matching target objects with language expressions demands reasoning capabilities, recent MLLM-based RefVOS methods mainly follow the successful image-based segmentation models represented by LISA [19]. LISA designs a dedicated $\langle\text{SEG}\rangle$ token to represent the target object and call a segmentation decoder Dec [18] to predict the object masks M:

$$\langle\text{SEG}\rangle = \text{MLLM}(I, R), M = \text{Dec}(I, \langle\text{SEG}\rangle), \tag{2}$$

where we slightly simplify the MLLM to output the segmentation token only. Besides LISA, more studies [30, 51] also enable MLLMs for RIS and object grounding tasks, but we mainly discuss LISA since the majority of MLLMs in RefVOS [3, 45, 52] follow its design.

3.3. Adapting Image MLLMs for RefVOS

To adapt an image-based framework like LISA to videos, the natural intuition is to extend it into a *multi-image* framework. Concretely, the MLLM has to take multiple frames $I_{1:T}$ as input, and the segmentation token(s) is used to decode masks for each frame, as adapted from Eqn. 2:

$$\langle\text{SEG}\rangle_{1:N} = \text{MLLM}(I_{1:N}, R),$$
$$M_{1:N} = \text{Dec}(I_{1:N}, \langle\text{SEG}\rangle_{1:N}), \tag{3}$$

where N is the maximum number of frames an MLLM can take (*e.g.*, 16) for training. As N is commonly smaller than the total number of video frames T that require segmentation, we observe different strategies for bridging this gap from the aspects listed in Table 1. These methods mostly sample N key frames and utilize an external VOS model [7] to propagate the masks. As discussed in Sec. 1, this not only
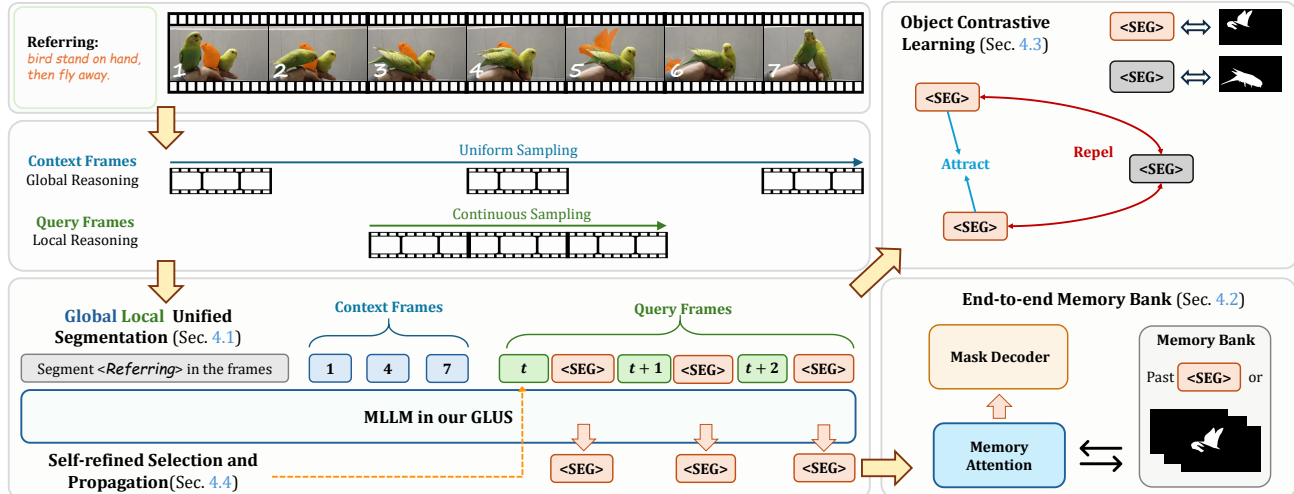
Figure 2. Overview of our GLUS. The yellow arrows mark the flow. **(1)** Sec. 4.1. Beginning from a video and referring, we design *context frames* and *query frames* to unify the distinct global and local reasoning into a single MLLM. The ⟨SEG⟩ tokens represent the target object. **(2)** Sec. 4.2. GLUS end-to-end integrates pre-trained VOS memory modules to enhance temporal reasoning and decouple the reliance of existing models on VOS post-processing. The memory-enhanced decoder decodes ⟨SEG⟩ tokens into masks. **(3)** Sec. 4.3. The ⟨SEG⟩ tokens are further supervised to better distinguish the different objects. **(4)** Sec. 4.4. The accuracy of ⟨SEG⟩ can be used for finetuning a plug-and-play key frame selector to further enhance GLUS's inference-time ability.

makes the system more complicated but also restricts the propagation from using the language instructions as guidance. Moreover, the performance of the MLLM is also bounded by the key frame selector, which is yet another multi-modal understanding task.

4. Methods

Our GLUS addresses these challenges via unifying global and local reasoning into a single MLLM (Sec. 4.1). Then we demonstrate an end-to-end framework involving both MLLM and a pre-trained VOS memory bank, which significantly enhances the temporal continuity of our framework and decouples the reliance on calling an off-the-shelf VOS model (Sec. 4.2). To further improve the fine-grained matching between language expressions and objects in global reasoning, we introduce an object-level contrastive loss (Sec. 4.3). Finally, we discuss the potential of our GLUS in enhancing the key frame selectors, which are commonly used as off-the-shelf oracles in prior arts [3, 45], via producing the pseudo-labels for finetuning (Sec. 4.4). An overview of GLUS is in Fig. 2.

4.1. Global-Local Unified MLLM

Dilemma of Global and Local Reasoning for Conventional MLLMs. The task of "RefVOS" involves two distinct abilities of global and local reasoning as described in Sec. 3.3. However, unifying them is contradicting for a single MLLM. **(1)** *Global reasoning*. As a most straightforward adaptation from LISA [19], the input to the MLLM is N uniformly sampled frames of the video to capture global

contexts. The output is then N or 1 corresponding ⟨SEG⟩ tokens for these frames. **(2)** *Local reasoning*. Although previous methods [3, 45] utilize external VOS for local mask propagation, we realize that the autoregressive formulation of MLLMs is already sufficient to address temporal consistency, where the token ⟨SEG⟩$_t$ is generated by using previous predictions ⟨SEG⟩$_{1:t-1}$ as contexts. However, the limitation of local reasoning is the adjacency of the N frames.

Previous approaches cannot unify both aspects because their training and inference styles are fundamentally different: models emphasizing global reasoning have to divide the video into separate folds to cover long-range contexts. In contrast, local reasoning models mainly rely on a sliding window covering a short range of neighboring frames. As in Table 1, previous methods deal with either global or local reasoning when training the models.

Context and Query Frames. Different from previous methods, our GLUS bypasses the above dilemma by explicitly unifying both global and local contexts for a single MLLM as in Fig. 2. Regarding global reasoning, we propose a set of sparse "*context frames*" $I_{1:N_C}^C$ covering the overall context of the videos and supporting the matching between language instructions and objects. Then GLUS introduces a set of continuous "*query frames*" $I_{1:N_Q}^Q$ for local reasoning and decoding the segmentation masks. As in Fig. 2, our GLUS places the query frames after the context frames because MLLMs function in a causal direction and such an order enables the decoding of segmentation results to receive video contexts. Intuitively, we imitate how a human would address the RefVOS task: first checking the

4

video contents roughly to localize an object and then marking the object locations iteratively on every frame.

Formally, we delve deep into the LLM part of our MLLM, and it generates the t-th segmentation token as:

$$\langle\texttt{SEG}\rangle_t = \texttt{LLM}([R, I_{1:N_c}^C, \\ I_1^Q, \langle\texttt{SEG}\rangle_1, I_2^Q, \langle\texttt{SEG}\rangle_2, ..., I_t^Q]), \quad (4)$$

where we autoregressively apply the referring, context frames, precedent query frames and segmentation tokens as the context for prediction. With such a design, our method GLUS unifies both global and local understanding in a simple and training-inference consistent way, which overcomes the limitations of previous MLLMs (Table 1).

Training. We adopt the most straightforward way of preparing context frames and query frames. Regarding context frames $I_{1:N_C}^C$, we first split the entire video into N_C equally spaced clips and then randomly sample one frame from each video clip to obtain N_c context frames, which is similar to [39]. For the query frames, we randomly sample short clips of N_Q frames to imitate the procedure of iteratively decoding masks for every frame of a video.

Our loss function is similar to the original LISA [19] involving both text and mask supervision. The objective of text supervision is enforcing the underlying LLM to generate the special $\langle\texttt{SEG}\rangle$ token and our GLUS adopts the standard cross entropy loss for this. Regarding mask supervision, we follow SAM2 [32] in combining the per-pixel binary cross-entropy (BCE) loss and DICE loss. More details are in the supplementary materials Sec B.2 .

Inference. Our GLUS exhibits a fully aligned inference procedure with training, which is a significant advantage. For context frames, we first evenly divide the video into N_C clips then select their center frame as context frames. Such a set of uniformly sampled frames cover the overall video contexts for global reasoning. Please note that this set of context frame remains identical during the whole inference procedure on this video. To obtain the masks for all of the T frames in the video, we use sliding windows with size N_q and stride 1 to generate a batch of query frames groups. With the frames being adjacent to each other, our strategy maximizes the temporal continuity for local reasoning.

4.2. End-to-end Memory Banks for MLLMs

Motivation: Rethinking VOS Models for RefVOS. To address the local continuity problem, existing MLLMs [3, 45] commonly treat VOS as an external module. One of the advantages of VOS methods compared with MLLMs is their using a memory bank to store long-term historical information, which is usually larger than the context window N of an MLLM. In addition, the VOS models also involve specialized memory reading and updating operations [7, 32] empowered by pre-trained transformers.

Therefore, our GLUS aims to end-to-end unify memory banks into MLLMs to enhance the ability of MLLMs for temporal reasoning and simplify the RefVOS framework. This significantly enhances GLUS in maintaining long-range temporal information for global and local reasoning.

Design. The key principle of our unified memory bank is the joint optimization of the memory bank modules and MLLM end-to-end as in Fig. 2. Concretely, the decoding of the t-th query frame further involves the memory bank:

$$M_t = \texttt{Dec}(I_t^Q, \langle\texttt{SEG}\rangle_t, \texttt{MemBank}). \quad (5)$$

Therefore, the gradient can be back-propagated from both the features stored in the memory banks and the pre-trained VOS attention used to read the memory bank features. In this way, our MLLM can cooperate with pre-trained memory banks from foundational VOS models and enjoy their enhancement in attending to historical information.

Our joint optimization enables aligned training and inference distributions, different from calling an external VOS model during inference time. In our case, the memory bank iterates through all the video frames with our query frames I_t^Q predicting the masks frame by frame. Accordingly, our training simulates such a streaming behavior with the N_Q query frames. As later verified in Sec. 5, our memory bank is a convenient *plug-and-play* component enhancing the "VOS" ability of RefVOS MLLMs. Its effectiveness further supports our design of dividing the frames into global context and local reasoning ones, which can seamlessly enjoy the progress in VOS models.

4.3. Object Contrastive Loss

Motivation. In addition to unifying global and local reasoning from an architectural perspective, we advance our investigation to enhance the correct matching between language instructions and target objects, which is the critical challenge of referring segmentation tasks.

As shown in the example of Fig. 1 and Fig. 2, a video might contain multiple objects with similar appearances to the ground truth. In this case, MLLM may easily confuse these objects and generate similar $\langle\texttt{SEG}\rangle$ tokens. Therefore, we aim to enhance the fine-grained perception of MLLMs by distinguishing such confusing object pairs.

Design. We introduce the object contrastive loss that maximizes the distance between the $\langle\texttt{SEG}\rangle$ tokens referring to different objects. When constructing positive pairs, GLUS notices the uniqueness of MeViS in that multiple different language expressions might refer to the same object. Therefore, the $\langle\texttt{SEG}\rangle$ tokens generated from different referring expressions of the same object are naturally positive samples. For negative sample pairs, GLUS aims to construct a sufficient number of negative examples motivated by [5]. This is achieved by maintaining a segmentation token bank [11, 12, 14, 44] of different objects.

5

Formally, the object contrastive loss follows the Sim-CLR [5], where the feature $\mathbf{v}$ of a $\langle\texttt{SEG}\rangle$ token is supervised by both positive sample $\mathbf{k}^+$ and negative samples $\mathbf{k}^-$:

$$-\sum_{\mathbf{v},\mathbf{k}^+}\log\frac{\exp\left(\mathrm{sim}(\mathbf{v},\mathbf{k}^+)/\tau\right)}{\exp\left(\mathrm{sim}(\mathbf{v},\mathbf{k}^+)/\tau\right)+\sum_{\mathbf{k}^-}\exp(\mathrm{sim}\left(\mathbf{v},\mathbf{k}^-\right)/\tau)}.\tag{6}$$

However, the natural challenge of contrastive loss is that not all objects have multiple distinct referring expressions and, accordingly, positive sample pairs. Thus, the contrastive loss is computed only when positive samples are presented. According to our statistics, the likelihood of sampling positive pairs within a data batch is approximately 91.5% for MeViS. As Ref-Youtube-VOS relatively lacks multiple complex referring expressions for the same objects, our contrastive loss is only applied to MeViS samples, but surprisingly benefits Ref-Youtube-VOS as well.

4.4. Self-refined Selection and Propagation

Guiding Key Frame Selection with GLUS. As the language instructions might refer to a motion occurring on parts of the videos, RefVOS methods commonly select the key frames to assist the localization of target objects. Although GLUS has already shown effective global-local unified reasoning, we are limited by the context window of MLLMs, and the sparsely sampled context frames could miss crucial information. Existing methods [45] adopt off-the-shelf video QA models for key frame selection, but such models are not aligned with the "key frame selection" objective for RefVOS models. In this sense, we suggest that the prediction quality from our GLUS is a natural indicator of the relevance between language instructions and video frames and could supervise such QA models. Therefore, the key frame selector, an optional plug-and-play module for RefVOS, can be enhanced by the guidance of our GLUS.

Concretely, we let GLUS annotate the Intersection over Union (IoU) scores on the training set, which functions as pseudo-labels for fine-tuning the video QA-based key frame selector to predict the IoU score on each frame. Intuitively, the key frame is simply selected as the frame with the highest score. As we noticed in Sec. 5.4.4, these pseudo-labels effectively improve the benefits of key frame selection.

GLUS Propagation without External VOS. With the key frame selected as the anchor, existing methods [3, 45] utilize external VOS models to propagate the results to other frames. However, these *online* modules cannot access global video contexts and comprehend the language instructions compared with our GLUS. Therefore, we distinguish ourselves by utilizing GLUS itself as the propagation module. Specifically, GLUS initiates the RefVOS procedural by treating the key frame as the first query frame, then conducts RefVOS on both forward and backward directions of

the video. More details are in the supplementary Sec. B.4.

5. Experiments

5.1. Datasets and Evaluation Metrics

MeViS. Our study primarily focuses on the MeViS [9], which presents complex video scenarios with multiple similar objects and intricate motion patterns. It is regarded as the most challenging RefVOS benchmark at present.

Ref-Youtube-VOS. We additionally evaluate on Ref-Youtube-VOS [34], which is an earlier and widely adopted dataset. Compared with MeViS, this dataset generally exhibits simpler scenarios and language expressions focusing less on the object motions. So its challenging level is lower.

ReVOS & ReasonVOS. GLUS could also tackle various scenarios that require reasoning with the LLMs. Therefore, we also evaluate our GLUS on Reasoning VOS [3, 45, 52] datasets concerning reasoning abilities, *i.e.*, reasoning with world knowledge.

Evaluation Metrics. Unless specified otherwise, the evaluation metrics used are $\mathcal{J}$ (average IoU), $\mathcal{F}$ (boundary F measure), and $\mathcal{J}\&\mathcal{F}$ (average of $\mathcal{J}$ and $\mathcal{F}$).

5.2. Baselines and Implementation Details

Model Architecture. Unless specified otherwise, we adopt LISA-7B-v1 [19] as the base MLLM to provide initial image segmentation abilities. We adopt the mask decoder from SAM-2 [32] as our segmentation decoder and incorporate the SAM-2 memory attention for end-to-end optimization with the memory bank. During training, only the MLLM and the SAM-2 decoder are trainable and the MLLM are fine-tuned with LoRA [15] for efficiency. The key frame selector is fine-tuned from Chat-UniVi-7B [17].

Multi-dataset Supervised Finetuning (SFT). Existing MLLM-based RefVOS methods commonly integrate multiple datasets for training to overcome the contradiction between the scarcity of video segmentation data and the large capacities of language models. Their training sets include RefVOS, image-based questions answering and segmentation, and VOS datasets. In comparison, we initialized from LISA and conducted SFT on RefVOS. We provide two SFT options: (1) Standard-SFT ($\mathbf{GLUS}^S$): We utilize MeViS and Ref-Youtube-VOS for SFT, and (2) Additional-SFT ($\mathbf{GLUS}^A$): we further include ReVOS [45], LV-VIS [38] and DAVIS-17 [29] to alleviate the data scarcity issue and tackle reasoning VOS. Due to limited computational resources, our Additional-SFT does not leverage object contrastive loss and keyframe selector. Notably, GLUS utilizes fewer datasets for SFT yet achieves better performance than prior arts.

Training Setup. For inputs, we use a context window of $N = 8$ frames, with 4 input context frames and 4 query frames. Due to our limited computation, we downsample

Method	MeViS $\mathcal{J}\&\mathcal{F}$	MeViS $\mathcal{J}$	MeViS $\mathcal{F}$	Ref-Youtube-VOS $\mathcal{J}\&\mathcal{F}$	Ref-Youtube-VOS $\mathcal{J}$	Ref-Youtube-VOS $\mathcal{F}$
Methods without LLMs						
URVOS [34]	27.8	25.7	29.9	47.2	45.2	49.1
LBDT [10]	29.3	27.8	30.8	49.4	48.2	50.6
MTTR [4]	30.0	28.8	31.2	55.3	54.0	56.6
ReferFormer [43]	31.0	29.8	32.2	62.9	61.3	64.6
OnlineRefer [42]	-	-	-	63.5	61.6	65.5
SOC [25]	-	-	-	**67.3**	65.3	**69.3**
TempCD [35]	-	-	-	65.8	63.6	68.0
LoSh [50]	-	-	-	64.2	62.5	66.0
LMPM [9]	37.2	34.2	40.2	-	-	-
DsHmp [13]	46.4	43.0	49.8	67.1	65.0	69.1
Methods with LLMs						
LISA-7B [19]	37.2	35.1	39.4	53.9	53.4	54.3
LISA-13B [19]	37.9	35.8	40.0	54.4	54.0	54.8
TrackGPT-7B [54]	40.1	37.6	42.6	56.4	55.3	57.4
TrackGPT-13B [54]	41.2	39.2	43.1	59.5	58.1	60.8
VideoGLaMM [27]	45.2	42.1	48.2	-	-	-
VideoLISA-3.8B [3]	44.4	41.3	47.6	63.7	61.7	65.7
VISA-7B [45]	43.5	40.7	46.3	61.5	59.8	63.2
VISA-13B [45]	44.5	41.8	47.1	63.0	61.4	64.7
ViLLa [52]	-	-	-	66.5	64.6	68.6
GLUS^S (ours)	50.3	47.5	53.2	66.6	65.0	68.3
GLUS^A (ours)	<u>51.3</u>	<u>48.5</u>	<u>54.2</u>	<u>67.3</u>	<u>65.5</u>	<u>69.0</u>

Table 2. The quantitative evaluation results on MeViS and Refer-Youtube-VOS. Our GLUS performs significantly better on the most challenging MeViS benchmark, which emphasizes understanding the complex motions of objects. Meanwhile, GLUS performs comparatively with other MLLM approaches. These support the effectiveness of our design. "-" means the performance not reported by a method; **bold** denotes the best scores; <u>underline</u> denotes the best scores among MLLM-based methods. "**GLUS**^{S}" and "**GLUS**^{A}" denote the dataset options of standard-SFT and additional-SFT (Sec. 5.2), respectively.

the features of every frame by 4x, resulting in 64 visual tokens per frame. The complete training process requires ∼25 hours on 4 NVIDIA 40G A100 GPUs, with 3000 default optimization steps. Each step corresponds to a batch size of 2 per device and 10 gradient accumulation steps. More implementation details are in the supplementary Sec. B.2.

5.3. Referring VOS Comparison

MeViS and Ref-Youtube-VOS. In Table 2, we compare GLUS with previous methods on two major RefVOS datasets: MeViS and Ref-Youtube-VOS. Our approach sets a new state-of-the-art on the challenging MeViS, with a substantial improvement: GLUS handles complex video scenarios with a $\mathcal{J}\&\mathcal{F}$ boost of over 5% compared to other MLLM-based RefVOS models. Additionally, GLUS surpasses the previous SOTA model, DsHmp [13], with an approximately 4% $\mathcal{J}\&\mathcal{F}$ improvement.

On Ref-Youtube-VOS, GLUS outperforms most

Method	ReVOS Reasoning $\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	ReVOS Referring $\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	Robustness	ReasonVOS $\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$
VISA-7B [45]	43.0	40.6	45.4	50.9	49.2	52.6	15.5	-	-	-
VISA-13B [45]	44.3	42.0	46.7	57.4	55.6	59.1	14.5	-	-	-
VideoLISA [3]	-	-	-	-	-	-	-	47.5	45.1	49.9
GLUS^A (ours)	**51.4**	**48.8**	**53.9**	**58.3**	**56.0**	**60.7**	**17.9**	**49.9**	**47.5**	**52.4**

Table 3. The quantitative evaluation results on ReVOS and ReasonVOS. Our GLUS performs significantly better on both datasets, among both reasoning data and referring data, which demonstrates the effect of GLUS in complex reasoning-require scenarios.

Method	MeViS (valid_u)	MeViS (valid)	RefYTB (valid)
Baselines			
Global Reasoning	55.1	47.2	62.8
Local Reasoning	56.9	46.2	61.6
GLUS (Ours)			
+ GLU	58.3	47.4	63.6
+ GLU + MB	59.7	49.5	65.2
+ GLU + MB + OC	60.9	**50.3**	65.5
+ GLU + MB + OC + KFS	**61.6**	**50.3**	**66.6**

Table 4. Every design from GLUS effectively improves the performance. "GLU": Global-local unification (Sec. 4.1), "MB": End-to-end memory bank (Sec. 4.2), "OC": Object contrastive loss (Sec. 4.3), "KFS": key frame selection (Sec. 4.4).

MLLM-based RefVOS models, only slightly lagging behind ViLLa [52] on $\mathcal{F}$ metric when using the subset of SFT datasets. Notably, previous works [3, 45, 52], including ViLLa, leverage more datasets spanning different tasks for SFT. Under a cleaner comparison, where our SFT similarly utilizes additional training sets (additional-SFT), our GLUS shows better performance on Ref-Youtube-VOS.

ReVOS & ReasonVOS. We additionally evaluate our GLUS in RefVOS scenes that require multi-modal reasoning capabilities: ReVOS and ReasonVOS. Since only our additional-SFT includes the training set of Reasoning VOS, we use GLUS^F for comparison, as in Table 3. GLUS demonstrates significant improvements compared with previous MLLM VISA [45]. In this way, our model shows consistent improvement on various reasoning tasks, *e.g.*, reasoning with object motion and world knowledge, which also demonstrates the necessity of utilizing MLLMs in RefVOS.

5.4. Ablation Studies

5.4.1. Global-local Unified Reasoning

We analyze the effect of our global-local unified reasoning (Sec. 4.1) in Table 4, where either global-only or local-only reasoning performs worse than our unified strategy ("GLU"). Our qualitative results in Fig. 1 also suggest the strength of our GLUS.

5.4.2. End-to-end Memory Bank

As discussed in Sec. 4.2, GLUS is inherently compatible with a VOS memory bank and optimize the MLLM end-
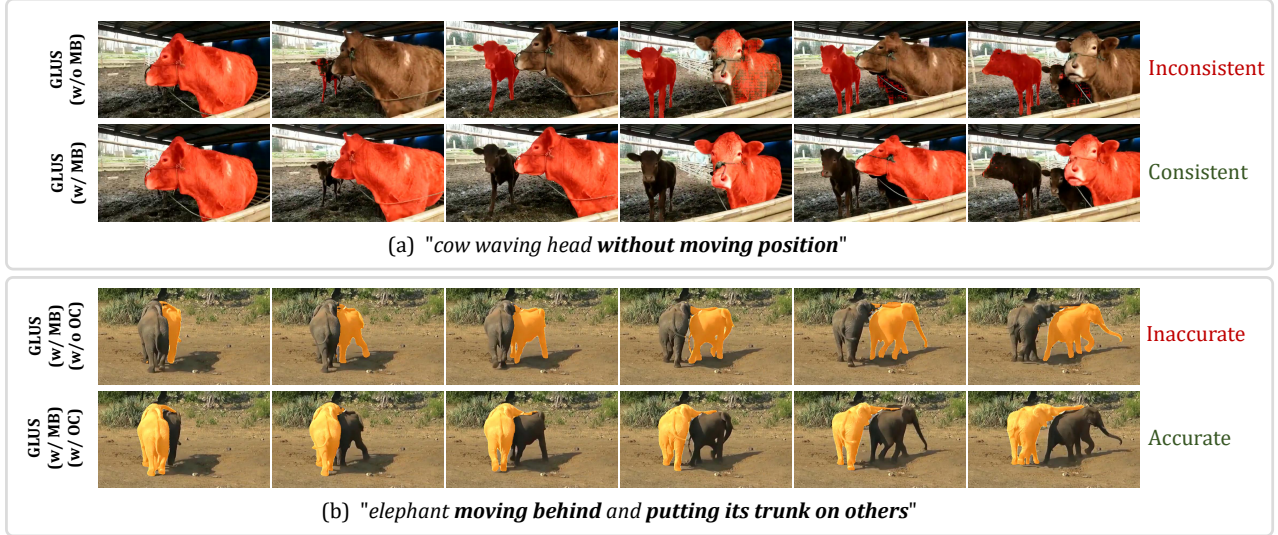
Figure 3. Qualitative comparisons on our key designs. "MB": End-to-end memory bank, "OC": Object contrastive loss. Our memory bank improves the temporal consistency of segmentation, and our object contrastive loss encourages more precise localization of objects.

Method	MeViS (valid_u)	MeViS (valid)	RefYTB (valid)
GLUS No Selector	60.9	**50.3**	65.5
LLaMA-VID 13B [23]	**61.6**	49.8	65.5
Chat-Univi 7B [17]	61.5	49.9	65.4
Fine-tuned Chat-Univi 7B (ours)	**61.6**	**50.3**	**66.6**

Table 5. Ablation study on choosing key frames. We compare our selector with no selection and two baselines that utilize an off-the-shelf model for selection. We show the benefits of key frame selection and the necessity of our pseudo-label fine-tuning.

to-end with the memory bank to better comprehend historical information. As in Table 4 ("GLU + MB"), GLUS improves on both RefVOS datasets after end-to-end training and inference with SAM-2's memory attention and memory banks. Qualitative observations also suggest that our unifying MLLM with the memory bank greatly enhances the consistency of the generated masks, as in Fig. 3(a).

5.4.3. Object Contrastive Loss

In Sec. 4.3, we introduced an object contrastive learning pipeline to enhance the fine-grained representations of ⟨SEG⟩ tokens. As in Table. 4 ("GLU + MU + OC"), GLUS enjoys improvement from the object contrastive loss, even though such contrastive learning only occurs on the MeViS data. This demonstrates the effectiveness of our design and the generalizability of learning from the challenging motion data (MeViS) for video understanding. As in Fig. 3(b), the contrastive loss corrects the misidentified referred objects.

5.4.4. Self-refined Key Frame Selection & Propagation

In Sec. 4.4, we propose a self-refinement framework to select key frame. To illustrate the necessity of fine-tuning selector with the pseudo-labels generated from GLUS, We

compare our selector choice with other selection methods and present the results in Table 5. As in Table 5, our fine-tuned selector outperforms off-the-shelf methods by a significant margin. These results underscore the benefits of utilizing a grounding model (e.g., GLUS) to provide fine-grained supervision to video question-answering models.

Additionally, unlike previous methods, which rely on an external VOS module, we propose utilizing the internal MLLM in GLUS for propagation. We compare propagating choices between GLUS and two external state-of-the-art VOS models: Cutie [8] and SAM-2 [32]. As presented in Table 6, GLUS outperforms both VOS methods by a significant margin, highlighting the necessity of utilizing video information and referring expressions for propagation.

To further illustrate the necessity of fine-tuning selector with the pseudo-labels generated from GLUS, We compare our selector choice with other selection methods and present the results in Table 5. "No selector" refers to regular GLUS without key frame selection, and we also adopt the off-the-shelf selectors of LLaMA-VID [23] and Chat-Univ [17] used in VISA [45]. As in Table 5, our fine-tuned selector outperforms off-the-shelf methods by a significant margin. These results underscore the benefits of utilizing a grounding model (e.g., GLUS) to provide fine-grained supervision to video question-answering models.

5.4.5. Propagation without External VOS

Unlike previous methods, which rely on an external VOS module for propagation after key frame selection, we propose utilizing the internal MLLM in GLUS for this task. We compare propagating the masks from the key frame to the whole video between GLUS and two external state-of-the-art VOS models: Cutie [8] and SAM-2 [32]. As pre-

Method	MeViS (valid_u)	MeViS (valid)	RefYTB (valid)
Cutie [8]	58.3	45.0	64.9
SAM-2 [32]	56.9	47.1	65.5
GLUS (ours)	**61.6**	**50.3**	**66.6**

Table 6. Ablation study on propagation modules. We compare our model with two state-of-the-art VOS methods Cutie [8] and SAM-2 [32]. The better performance of our GLUS indicates the effectiveness of our unified design in decoupling the need to use external VOS models for propagation.

MeViS : RefYTB	MeViS (valid_u)	MeViS (valid)	RefYTB (valid)	Best Step
2:1	**60.8**	49.0	64.1	1500
1:1	<u>59.7</u>	<u>49.5</u>	65.2	1500
1:2	59.6	49.3	**65.6**	2500
4:15	59.6	**49.9**	<u>65.5</u>	3000

Table 7. Ablation studies on sampling ratio of MeViS:Ref-Youtube-VOS for training. We report the performance ($\mathcal{J}\&\mathcal{F}$) and the training steps needed for convergence. <u>underline</u> denotes the second best. We select 1:1 as the standard ratio for GLUS to balance performance across datasets and training efficiency. (The 4:15 ratio is adopted from [45].)

sented in Table 6, GLUS outperforms both VOS methods by a significant margin, highlighting the necessity of utilizing global video information and referring expressions for propagation.

5.4.6. Sampling Ratio of Training Datasets

As noticed in previous works [41], balancing the training data is critical for vision language models. We observe the same when training GLUS with Ref-Youtube-VOS and MeViS. For this ablation, we use the GLUS with memory bank and global-local unified reasoning enabled, and train it across different sampling ratios of the two datasets. The performance and optimization steps needed for convergence are in Table 7. For balanced performance and training efficiency, we select 1:1 as the standard sampling rate for our models.

5.4.7. Data Scarcity of MLLM in Video Segmentation

Fine-tuning LLMs requires large amounts of data, especially for video MLLMs [20, 40, 41]. However, video data is scarce, especially when requiring fine-grained annotations like RefVOS. With the default training steps 3000, the training of GLUS without extended datasets averagely spans ~11.6 epochs over the whole frames set, which contrasts the common 1 or 2 epochs SFT schedule for vision-language models fine-tuned with sufficient data [17, 20, 22, 24, 40, 41].

This led to noticeable overfitting with more training steps, according to the change of validation set performance (MeViS valid_u) in Fig. 4. Although the object contrastive loss alleviates the overfitting issue, they all suffer from a
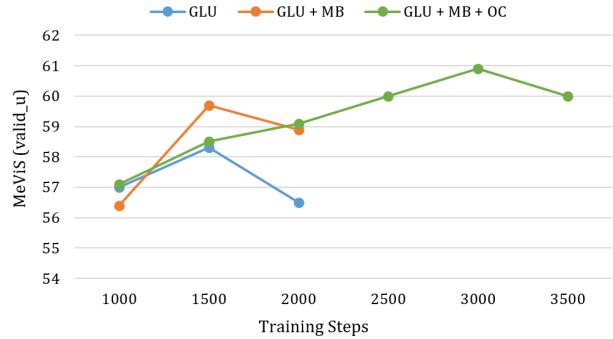

Figure 4. Curves of MeViS valid_u performance ($\mathcal{J}\&\mathcal{F}$) with distinct training steps. The figure clearly demonstrates noticeable overfitting in the model. "GLU": Global-local unification, "MB": End-to-end memory bank, "OC": Object contrastive loss.

significant drop at the final steps. We hypothesize that such a data scarcity problem constrains the performance of video MLLMs, especially when they don't have tailored designs such as hierarchical perception [13]. We hope our observation can encourage more explorations on scaling up the video segmentation data.

6. Conclusion

We introduce a simple yet effective framework based on MLLMs for referring video object segmentation (RefVOS). Named "GLUS," our method establishes unified global and local reasoning in a single LLM, addressing both "Ref" and "VOS" challenges. The central design is to provide MLLM with both global (*context frames*) and local (*query frames*) contexts. Such unified reasoning is further enhanced with end-to-end optimized VOS memory modules to improve the consistency of GLUS. Finally, GLUS introduces plug-and-play *object contrastive loss* and *pseudo-labeling* for key frame selection, enabling the MLLM to distinguish the correct object and frame with its limited context window. Our GLUS establishes the new state of the arts on RefVOS benchmarks. We hope our baseline can inspire more systematic studies enabling MLLMs to fine-grained video understanding.

References

[1] Ali Athar, Jonathon Luiten, Alexander Hermans, Deva Ramanan, and Bastian Leibe. HODOR: High-level object descriptors for object re-segmentation in video learned from static images. In *CVPR*, 2022. 3

[2] Ali Athar, Alexander Hermans, Jonathon Luiten, Deva Ramanan, and Bastian Leibe. TarVis: A unified approach for target-based video segmentation. In *CVPR*, 2023. 3

[3] Zechen Bai, Tong He, Haiyang Mei, Pichao Wang, Ziteng Gao, Joya Chen, Lei Liu, Zheng Zhang, and Mike Zheng Shou. One token to seg them all: Language instructed reasoning segmentation in videos. *NeurIPS*, 2024. 2, 3, 4, 5, 6, 7

[4] Adam Botach, Evgenii Zheltonozhskii, and Chaim Baskin. End-to-end referring video object segmentation with multi-modal transformers. In *CVPR*, 2022. 2, 7

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*. PMLR, 2020. 5, 6

[6] Yuhua Chen, Jordi Pont-Tuset, Alberto Montes, and Luc Van Gool. Blazingly fast video object segmentation with pixel-wise metric learning. In *CVPR*, 2018. 3

[7] Ho Kei Cheng and Alexander G Schwing. Xmem: Long-term video object segmentation with an atkinson-shiffrin memory model. In *ECCV*, 2022. 3, 5

[8] Ho Kei Cheng, Seoung Wug Oh, Brian Price, Joon-Young Lee, and Alexander Schwing. Putting the object back into video object segmentation. In *CVPR*, 2024. 3, 8, 9

[9] Henghui Ding, Chang Liu, Shuting He, Xudong Jiang, and Chen Change Loy. Mevis: A large-scale benchmark for video segmentation with motion expressions. In *ICCV*, 2023. 2, 6, 7, 1

[10] Zihan Ding, Tianrui Hui, Junshi Huang, Xiaoming Wei, Jizhong Han, and Si Liu. Language-bridged spatial-temporal interaction for referring video object segmentation. In *CVPR*, 2022. 7

[11] Mohamed El Banani, Karan Desai, and Justin Johnson. Learning visual representations via language-guided sampling. In *CVPR*, 2023. 5

[12] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. 5

[13] Shuting He and Henghui Ding. Decoupling static and hierarchical motion perception for referring video segmentation. In *CVPR*, 2024. 2, 7, 9, 1

[14] Shuting He and Henghui Ding. Decoupling static and hierarchical motion perception for referring video segmentation. In *CVPR*, 2024. 5

[15] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *ICLR*, 2021. 6, 1, 2

[16] Yuan-Ting Hu, Jia-Bin Huang, and Alexander G Schwing. Videomatch: Matching based video object segmentation. In *ECCV*, 2018. 3

[17] Peng Jin, Ryuichi Takanobu, Wancai Zhang, Xiaochun Cao, and Li Yuan. Chat-univi: Unified visual representation empowers large language models with image and video understanding. In *CVPR*, 2024. 6, 8, 9, 2

[18] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment anything. In *ICCV*, 2023. 3

[19] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. In *CVPR*, 2024. 2, 3, 4, 5, 6, 7, 1

[20] 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. 9

[21] Minghan Li, Shuai Li, Xindong Zhang, and Lei Zhang. Univs: Unified and universal video segmentation with prompts as queries. In *CVPR*, 2024. 3

[22] Yanwei Li, Yuechen Zhang, Chengyao Wang, Zhisheng Zhong, Yixin Chen, Ruihang Chu, Shaoteng Liu, and Jiaya Jia. Mini-gemini: Mining the potential of multi-modality vision language models. *arXiv preprint arXiv:2403.18814*, 2024. 9

[23] Yanwei Li, Chengyao Wang, and Jiaya Jia. Llama-vid: An image is worth 2 tokens in large language models. In *ECCV*, 2025. 8

[24] Ji Lin, Hongxu Yin, Wei Ping, Pavlo Molchanov, Mohammad Shoeybi, and Song Han. Vila: On pre-training for visual language models. In *CVPR*, 2024. 9

[25] Zhuoyan Luo, Yicheng Xiao, Yong Liu, Shuyan Li, Yitong Wang, Yansong Tang, Xiu Li, and Yujiu Yang. SOC: semantic-assisted object cluster for referring video object segmentation. In *NeurIPS*, 2023. 2, 7

[26] 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. 3

[27] 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. *arXiv preprint arXiv:2411.04923*, 2024. 2, 7

[28] Renjie Pi, Jiahui Gao, Shizhe Diao, Rui Pan, Hanze Dong, Jipeng Zhang, Lewei Yao, Jianhua Han, Hang Xu, Lingpeng Kong, and Tong Zhang. Detgpt: Detect what you need via reasoning. In *ACL*, 2023. 3

[29] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017. 6

[30] 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 *CVPR*, 2024. 3

[31] Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *KDD*, 2020. 1

[32] 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. 2, 3, 5, 6, 8, 9, 1

[33] Zhongwei Ren, Zhicheng Huang, Yunchao Wei, Yao Zhao, Dongmei Fu, Jiashi Feng, and Xiaojie Jin. Pixellm: Pixel reasoning with large multimodal model. In *CVPR*, 2024. 3

[34] Seonguk Seo, Joon-Young Lee, and Bohyung Han. Urvos: Unified referring video object segmentation network with a large-scale benchmark. In *ECCV*, 2020. 2, 6, 7

[35] Jiajin Tang, Ge Zheng, and Sibei Yang. Temporal collection and distribution for referring video object segmentation. In *ICCV*, 2023. 2, 7

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 3

[37] Paul Voigtlaender, Yuning Chai, Florian Schroff, Hartwig Adam, Bastian Leibe, and Liang-Chieh Chen. FeelVOS: Fast end-to-end embedding learning for video object segmentation. In *CVPR*, 2019. 3

[38] Haochen Wang, Cilin Yan, Shuai Wang, Xiaolong Jiang, XU Tang, Yao Hu, Weidi Xie, and Efstratios Gavves. Towards open-vocabulary video instance segmentation. In *ICCV*, 2023. 6

[39] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, 2016. 5

[40] Yi Wang, Kunchang Li, Yizhuo Li, Yinan He, Bingkun Huang, Zhiyu Zhao, Hongjie Zhang, Jilan Xu, Yi Liu, Zun Wang, et al. Internvideo: General video foundation models via generative and discriminative learning. *ECCV*, 2022. 9

[41] Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Guo Chen, Baoqi Pei, Rongkun Zheng, Jilan Xu, Zun Wang, et al. Internvideo2: Scaling video foundation models for multimodal video understanding. In *ECCV*, 2024. 9

[42] Dongming Wu, Tiancai Wang, Yuang Zhang, Xiangyu Zhang, and Jianbing Shen. Onlinerefer: A simple online baseline for referring video object segmentation. In *ICCV*, 2023. 2, 7

[43] Jiannan Wu, Yi Jiang, Peize Sun, Zehuan Yuan, and Ping Luo. Language as queries for referring video object segmentation. In *CVPR*, 2022. 2, 7

[44] Junfeng Wu, Qihao Liu, Yi Jiang, Song Bai, Alan Yuille, and Xiang Bai. In defense of online models for video instance segmentation. In *ECCV*, 2022. 5

[45] Cilin Yan, Haochen Wang, Shilin Yan, Xiaolong Jiang, Yao Hu, Guoliang Kang, Weidi Xie, and Efstratios Gavves. Visa: Reasoning video object segmentation via large language models. *ECCV*, 2024. 2, 3, 4, 5, 6, 7, 8, 9

[46] Cheng-Yen Yang, Hsiang-Wei Huang, Wenhao Chai, Zhongyu Jiang, and Jenq-Neng Hwang. Samurai: Adapting segment anything model for zero-shot visual tracking with motion-aware memory. *arXiv preprint arXiv:2411.11922*, 2024. 3

[47] Linjie Yang, Yanran Wang, Xuehan Xiong, Jianchao Yang, and Aggelos K Katsaggelos. Efficient video object segmentation via network modulation. In *CVPR*, 2018. 3

[48] Zongxin Yang, Yunchao Wei, and Yi Yang. Collaborative video object segmentation by foreground-background integration. In *ECCV*, 2020.

[49] Zongxin Yang, Yunchao Wei, and Yi Yang. Collaborative video object segmentation by multi-scale foreground-background integration. *TPAMI*, 44(9), 2021. 3

[50] Linfeng Yuan, Miaojing Shi, Zijie Yue, and Qijun Chen. Losh: Long-short text joint prediction network for referring video object segmentation. In *CVPR*, 2024. 2, 7

[51] Yichi Zhang, Ziqiao Ma, Xiaofeng Gao, Suhaila Shakiah, Qiaozi Gao, and Joyce Chai. Groundhog: Grounding large language models to holistic segmentation. In *CVPR*, 2024. 3

[52] Rongkun Zheng, Lu Qi, Xi Chen, Yi Wang, Kun Wang, Yu Qiao, and Hengshuang Zhao. Villa: Video reasoning segmentation with large language model. *arXiv preprint arXiv:2407.14500*, 2024. 2, 3, 6, 7

[53] Junbao Zhou, Ziqi Pang, and Yu-Xiong Wang. Rmem: Restricted memory banks improve video object segmentation. In *CVPR*, 2024. 3

[54] Jiawen Zhu, Zhi-Qi Cheng, Jun-Yan He, Chenyang Li, Bin Luo, Huchuan Lu, Yifeng Geng, and Xuansong Xie. Tracking with human-intent reasoning. *arXiv preprint arXiv:2312.17448*, 2023. 2, 3, 7

11

GLUS: Global-Local Reasoning Unified into
A Single Large Language Model for Video Segmentation

Supplementary Material

A. Demo Video

In Demo, we provide six qualitative comparisons between the previous state-of-the-art (DsHmp [13]) and our GLUS with the videos in MeViS [9]. Notably, these examples illustrate three challenging aspects of RefVOS: (1) **Motion Understanding**: RefVOS models have to distinguish similar objects with their motions; (2) **Global Reasoning**: RefVOS models should be capable of realizing global reasoning to segment the objects presented only in a short video clip; (3) **Vision-Language Reasoning**: RefVOS models should perform vision-language unified reasoning in complex scenarios. The six examples demonstrate that our GLUS effectively tackles RefVOS in challenging language-guided segmentation cases.

B. Implementation Details

This section provides a detailed explanation of the specific model architectures and workflow of GLUS.

B.1. Model Architectures

Multimodal LLM. The input embeddings for the MLLM are generated by processing each context and query frame individually through the vision backbone, VB. Subsequently, a vision-to-language projection layer, $\phi_{V \to L}$, is applied to the outputs:

$$F_t^C = \phi_{V \to L}(\text{VB}(I_t^C)), F_t^Q = \phi_{V \to L}(\text{VB}(I_t^Q)), \quad \text{(A)}$$

where F_t^C and F_t^Q are the features for the context and query frames. Then MLLM generates the t-th segmentation token as:

$$\langle \text{SEG} \rangle_t = \text{LLM}([R, F_{1:N_c}^C, \\ F_1^Q, \langle \text{SEG} \rangle_1, F_2^Q, \langle \text{SEG} \rangle_2, ..., F_t^Q]). \quad \text{(B)}$$

This process follows our global-local unified design, and we adopt LISA-7B-v1 [19] for the initialization of LLM, projector $\phi_{V \to L}$, and backbone VB.

Mask Decoder. Our utilization of the mask decoder follows the style of LISA [19] and SAM-2 [32]. After obtaining $\langle \text{SEG} \rangle_t$, GLUS first extracts the hidden embedding $\hat{h}_t$ from the penultimate layer of the MLLM. A language-to-vision projection layer, $\phi_{L \to V}$, is then applied to $\hat{h}_t$ to generate a prompt for the mask decoder, h_t. Next, a vision encoder, Enc, processes the query frames to produce encoded features. Using the prompt and the encoded features, the mask decoder, Dec, is applied to the query image I_t^Q,

generating its corresponding mask M_t:

$$h_t = \phi_{L \to V}(\hat{h}_t), M_t = \text{Dec}(\text{Enc}(I_t^Q), h_t) \quad \text{(C)}$$

In our experiments, we initialize the weights of $\phi_{L \to V}$ projection layer with LISA-7B-v1 and utilize SAM-2 to initialize image encoder Enc and mask decoder Dec..

Memory Bank. Each time a mask M_t is generated, GLUS is able to encode it using a memory encoder, Enc_M, and stores the resulting feature F_t^M in MemBank. For memory attention, we adopt the design of SAM-2 [32], selecting features from up to m masks in MemBank. Attention is then applied to these features along with the decoded image to produce the input for the mask decoder:

$$F_t^M = \text{Enc}_M(M_t), \ \text{MemBank.Push}(F_t^M)$$
$$\hat{F}_{t+1}^M = \text{Concat}(F_{i_1}^M, F_{i_2}^M, ..., F_{i_m}^M)$$
$$\hat{F}_{t+1}^Q = \text{MemAttn}(\text{Enc}(I_{t+1}^Q), \hat{F}_{t+1}^M) \quad \text{(D)}$$
$$M_{t+1} = \text{Dec}(\hat{F}_{t+1}^Q, h_{t+1})$$

where $\{i_p\}_{p=1}^m$ is the selected masks from memory bank following SAM-2. We adopt SAM-2's memory attention module and memory encoder in our experiments.

B.2. GLUS Training Details

This section provides detailed training configurations for GLUS (Sec. 4), as summarized in Table A. During training, only the MLLM (fine-tuned with LoRA [15]), mask decoder, and projection layers are trainable. DeepSpeed [31] is employed to improve training efficiency. The sampling frequency in the memory bank is set to 1 during training to maximize its utilization. The training process takes approximately 25 hours on 4 NVIDIA A100 GPUs (40 GB each), with 3000 steps, 10 gradient accumulation steps and a batch size of 2 per device.

The training objective incorporates cross entropy (CE) loss, mask loss (comprising mask DICE loss and mask BCE loss), and contrastive loss, as described in Sec. 4.3. The corresponding weights, λ_{ce}, λ_{dice}, λ_{bce}, and λ_{ct}, are used to compute their respective averages.

B.3. GLUS Inference Details

During inference, GLUS employs a sliding window approach with a size of 4 and a stride of 1 for the query frames. The mask of the last query frame is used as the context of the next group of query frames. The sampling frequency for the memory bank is set to sample once per 3 frames, and a maximum of 7 masks are used in mask attention. Addi-

Config	Value
context frame num	4
question frame num	4
input resolution	224
features downsampling rate	4
optimizer	Adam
optimizer momentum	$\beta_1, \beta_2 = 0.9, 0.95$
optimizer weight decay	0.0
learning rate	3e-4
LoRA rank	8
λ_{ce}	1.0
λ_{dice}	0.5
λ_{bce}	2.0
λ_{ct}	0.1
batch size	80
gradient accumulation steps	10
warmup steps	100

Table A. Implementation details of GLUS training process.

tional ablation studies on sampling frequency are provided in Sec. C.

B.4. Selector training and inference

Data Annotation   To generate the pseudo-labels for fine-tuning the selector model, we use GLUS to generate the masks on the training set and compute the IoU of the masks. To mitigate the risk of overfitting, we adopt an early-stop model (trained for 500 steps) rather than the final model (trained for 3000 steps). For faster training of the selector, we label only half of the training set as the training data for selector fine-tuning.

Implementation Details   We use Chat-Univi [17] as the base Video-QA model. Similar to the design of recent grounding LLMs [3, 19, 45, 54], we introduce a special token, $\langle$SCORE$\rangle$, into the LLM vocabulary and employ an MLP to project the corresponding embeddings. During training, we randomly sample 8 frames to represent video context and produce the score for each query frame. The hidden embedding of the score token, $\hat{h}_s$, is generated as:

$$\hat{h}_s = \texttt{Selector}([P, F_{1:8}^C, F^Q, \langle\texttt{SCORE}\rangle]) \qquad (E)$$

where P represents the language prompt. The hidden embedding of $\langle$SCORE$\rangle$ is then projected to score s through an MLP layer. The selector fine-tuning objective consists of two components: $\mathcal{L}_s$, an L_1 loss that supervises the frame score s using the IoU pseudo-labels y of the query frame, and $\mathcal{L}_{\text{txt}}$, a cross-entropy loss that supervises the text outputs of the LLM:

$$
\begin{aligned}
s &= \phi_{\text{proj}}(\hat{h}_s), \\
\mathcal{L}_s &= |y - s|, \qquad\qquad (F)\\
\mathcal{L}_{\text{all}} &= \mathcal{L}_{\text{txt}} + \lambda_s \cdot \mathcal{L}_s
\end{aligned}
$$

For efficient training, the selector LLM is fine-tuned with LoRA [15], while the MLP layer is fully trainable. Further

details on selector training are provided in Table B.

Config	Value
context frame num	8
query frame num	1
optimizer	Adam
optimizer momentum	$\beta_1, \beta_2 = 0.9, 0.95$
optimizer weight decay	0.0
learning rate	3e-4
LoRA rank	8
λ_s	1.0
batch size	80
gradient accumulation steps	10
MLP layer num	3

Table B. Implementation details of selector training process.

Inference and Propagation   The selector is trained to predict a confidence score for each frame in a test-time video, reflecting the importance of a frame with respect to the given expression. During inference, we first select the frame with the highest score as the key frame for each video-expression pair. We then use GLUS to initiate tracking from the selected frame in both forward and backward propagation directions and iteratively generate the predictions for the entire video.

C. Additional Studies

Memory Bank Sampling Frequency   The VOS memory bank is integrated into our framework and optimized end-to-end to enhance global-local reasoning capabilities in complex scenarios (Sec. 4.2). We evaluate the impact of memory stride in Table C, where a longer stride prioritizes global reasoning, while a shorter stride emphasizes local consistency. We show that GLUS performs stably with varied memory bank strides, because of its design unifying both global and local reasoning.

Sampling Frequency	MeViS (valid_u)
w/o MB	58.3
1	59.3
3	59.7
5	59.7
7	59.7
9	59.7

Table C. Ablation studies on the sampling frequency of memory bank. We select 3 as the default stride of the sampling frequency, following SAM2. "MB": Memory Bank.

D. Limitations and Future Works

Our work mainly focuses on the *fine-tuning* phase of a multimodal large language model for referring video object segmentation. Therefore, the visual backbone and LLM are

limited in understanding the video. From this perspective, meaningful future work would start from an MLLM *pretrained* for video understanding to further enhance the motion understanding.

In addition, our computational resources heavily constrain our *context lengths* for an MLLM and limit the capability for video understanding. Concretely, we have to downsample the visual features and can only sample 4 context frames to summarize the video content, which might not cover the critical contexts if motions are happening fast. We hope combining our GLUS design with longer context windows can further unleash its potential.

Finally, we notice that the amount of data has become a bottleneck for video reasoning (Fig. 4). Therefore, future work can focus on improving the data scale and quality, where we hope the benefit of pseudo-labeling from GLUS can also be of use.