UniPixel: Unified Object Referring and Segmentation for Pixel-Level Visual Reasoning

Ye Liu[1,2], **Zongyang Ma**[2,3], **Junfu Pu**[2], **Zhongang Qi**[4], **Yang Wu**[5], **Ying Shan**[2], **Chang Wen Chen**[1]*
[1] The Hong Kong Polytechnic University [2] ARC Lab, Tencent PCG
[3] Institute of Automation, Chinese Academy of Sciences [4] vivo Mobile Communication Co.
[5] MindWingman Technology (Shenzhen) Co., Ltd.
coco.ye.liu@connect.polyu.hk
https://polyu-chenlab.github.io/unipixel/

Figure 1: **UniPixel** flexibly supports a large variety of fine-grained image and video understanding tasks, including referring/reasoning/interactive segmentation, motion-grounded video reasoning, and referred video description & question answering. It can also handle a novel **PixelQA** task that jointly requires object-centric referring, segmentation, and question answering in videos.

Abstract

Recent advances in Large Multi-modal Models (LMMs) have demonstrated their remarkable success as general-purpose multi-modal assistants, with particular focuses on holistic image- and video-language understanding. Conversely, less attention has been given to scaling fine-grained pixel-level understanding capabilities, where the models are expected to realize pixel-level alignment between visual

* *Corresponding author.*

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

signals and language semantics. Some previous studies have applied LMMs to related tasks such as region-level captioning and referring expression segmentation. However, these models are limited to performing either referring or segmentation tasks independently and fail to integrate these fine-grained perception capabilities into visual reasoning. To bridge this gap, we propose **UniPixel**, a large multi-modal model capable of flexibly comprehending visual prompt inputs and generating mask-grounded responses. Our model distinguishes itself by seamlessly integrating pixel-level perception with general visual understanding capabilities. Specifically, UniPixel processes visual prompts and generates relevant masks on demand, and performs subsequent reasoning conditioning on these intermediate pointers during inference, thereby enabling fine-grained **pixel-level reasoning**. The effectiveness of our approach has been verified on 10 benchmarks across a diverse set of tasks, including pixel-level referring/segmentation and object-centric understanding in images/videos. A novel **PixelQA** task that jointly requires referring, segmentation, and question answering is also designed to verify the flexibility of our method.

1 Introduction

Large Multi-modal Models (LMMs) have been the de facto standard for developing general-purpose assistants. By effectively aligning multi-modalities with language, their significance has been demonstrated across various applications, including multi-modal analysis [59, 19, 107, 108, 48], autonomous driving (AD) [16, 79, 106, 11], and Embodied AI [111, 22, 30, 99].

In the field of visual-language understanding, efforts have been dedicated to developing *holistic understanding models*, where simple projection layers between visual encoders and LLMs are utilized to bridge vision and language modalities. Supported by large-scale alignment pre-training and visual instruction tuning, such a straightforward paradigm achieves strong performance in holistic understanding tasks such as captioning [40, 6, 114] and general question answering [36, 24, 54, 49]. However, these models exhibit two fundamental limitations in fine-grained scenarios. **First**, their interactions with users are limited to text format, lacking support for more intuitive forms of communication such as drawing points/boxes as references or grounding model responses with key regions represented by masks. **Second**, the internal reasoning process of these models predominantly operates at a coarse level, directly perceiving the entire content rather than reasoning over specific objects/regions, making them hard to understand fine-grained details. Some previous studies have explored the application of LMMs to related tasks such as region-level captioning [12, 102, 103], referring expression segmentation [29, 55, 41, 21, 71, 62], and reasoning segmentation [32, 27, 96, 4, 112]. Nevertheless, their models are limited to performing either referring or segmentation tasks independently via rigidly defined input/output templates (*e.g.*, "It's <SEG>." in LISA [32]), lacking the flexibility to comprehend user-referred concepts and generate mask-grounded responses simultaneously. More importantly, these methods cannot integrate such fine-grained perception capabilities with their original human-like [90, 89, 88, 92, 91] multi-modal reasoning abilities, resulting in degraded performance on general visual understanding benchmarks [100, 93, 28].

In this work, we seek to bridge this gap by introducing **UniPixel**, a large multi-modal model that can flexibly comprehend visual prompt inputs (*i.e.*, points, boxes, and masks) and generate mask-grounded responses. Our model significantly differentiates itself from existing ones by unifying the internal representations of referred and segmented objects via a novel **object memory bank**, which is a hashmap storing the spatial-temporal information of object-of-interests. During inference, UniPixel initializes the object memory bank and updates it on demand by adding object-centric information according to the context. The model responses are then generated conditioning on the fine-grained object memory. Benefits from such unification, UniPixel is able to perform not only basic referring/segmentation tasks, but also flexible **pixel-level reasoning** tasks that require simultaneous visual prompt comprehension and mask prediction. As illustrated in Fig. 1 (the last row), given a video[2], a question, and optionally a visual prompt (*e.g.*, a point specified by a click on an object in any frame), UniPixel can (1) infer the mask for the referred object in the corresponding frame, (2) propagate it to all video frames containing the same instance, (3) extract the mask-grounded object features, and finally (4) answer the question conditioning on both the video-level and object-centric

[2] *Images are treated as single-frame videos, thus we do not explicitly differentiate them in this work.*

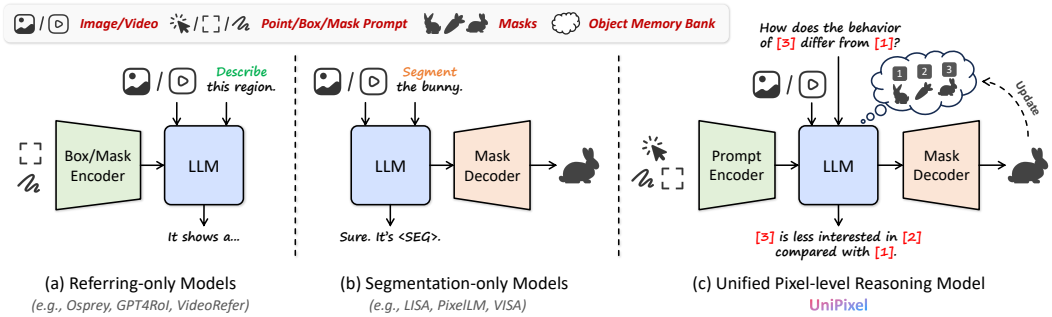

Figure 2: **Schematic comparison between UniPixel and its counterparts.** To the best of our knowledge, UniPixel is the first unified method supporting simultaneous object referring and segmentation.

information. All these operations are seamlessly conducted *within a single model*, eliminating the need for external frame samplers [96], mask generators [102, 103], or object trackers [4].

We evaluate the effectiveness of UniPixel from two aspects, *i.e.*, basic referring/segmentation capabilities and flexible pixel-level reasoning capabilities. For the first aspect, we conduct extensive experiments on 10 public benchmarks across 9 image/video referring/segmentation tasks. Our method achieves state-of-the-art performance in diverse scenarios. Notably, on the challenging video reasoning segmentation and referred video QA tasks, our 3B model obtains **62.1** $\mathcal{J}\&\mathcal{F}$ on ReVOS [96] and **72.8%** Acc on VideoRefer-Bench$^{\text{Q}}$ [103], surpassing strong counterparts with 7B $\sim$ 13B parameters. Further ablation studies also demonstrate the mutual reinforcement effect of referring and segmentation. For the second aspect, we introduce a novel **PixelQA** task that jointly requires object-centric referring, segmentation, and QA in videos, which cannot be handled by existing methods. UniPixel establishes a strong baseline for this novel setting. Our contributions are summarized below:

1. We propose **UniPixel**, a unified large multi-modal model that supports flexible object referring and segmentation in images and videos, via a novel **object memory bank** design.
2. Our model achieves state-of-the-art performance on 10 public benchmarks across 9 referring/segmentation tasks, verifying the **mutual reinforcement effect** of such unification.
3. We also introduce a novel **PixelQA** task that jointly requires object-centric referring, segmentation, and QA in videos, where UniPixel establishes a strong baseline for this setting.

2 Related Work

Large Multi-modal Models The remarkable success of large multi-modal models (LMMs) has shifted the paradigm of visual-language understanding from close-ended experts to open-ended task solvers. Early attempts [43, 42, 17, 115] involve an MLP projector or Q-Former [34] to align visual encoders to LLMs, enabling open-ended tasks such as visual question answering. With advanced designs such as dynamic resolution and data augmentation, open-source models, *e.g.*, Qwen-VL [2, 77, 3] and InternVL [14, 74, 13] series, have narrowed the gap with advanced proprietary models like the GPT [58, 59] and Gemini families [67, 18]. Recent studies [60, 25, 51, 37, 48] also explore test-time scaling on visual-language understanding. However, these methods are spatially coarse-grained. UniPixel can also be regarded as an object-centric test-time scaling approach, where key objects are first segmented then encoded to facilitate the subsequent reasoning process.

Visual Referring and Segmentation To meet the growing demand for fine-grained visual understanding [50, 46, 47, 45, 84], recent efforts have focused on enhancing LMMs with object referring and segmentation capabilities, as compared in Fig. 2. LISA [32] is a representative model that enables LMM-based segmentation by integrating SAM [31] as its decoder. They also introduced a novel reasoning segmentation task, requiring models to perform segmentation based on implicit queries. Other works in this direction [104, 69, 110, 65, 27] have explored advanced mask decoders, more flexible tasks, and larger-scale datasets. Recent studies have also extended these capabilities to videos [4, 96, 101]. Additionally, some research has examined regional understanding through boxes [12] and masks [102, 103]. While recent approaches attempt to unify these two capabilities, they either support only images [65] or rely on sub-optimal, tool-based pipelines [23]. To the best of knowledge, UniPixel is the first end-to-end method unifying object referring and mask prediction.

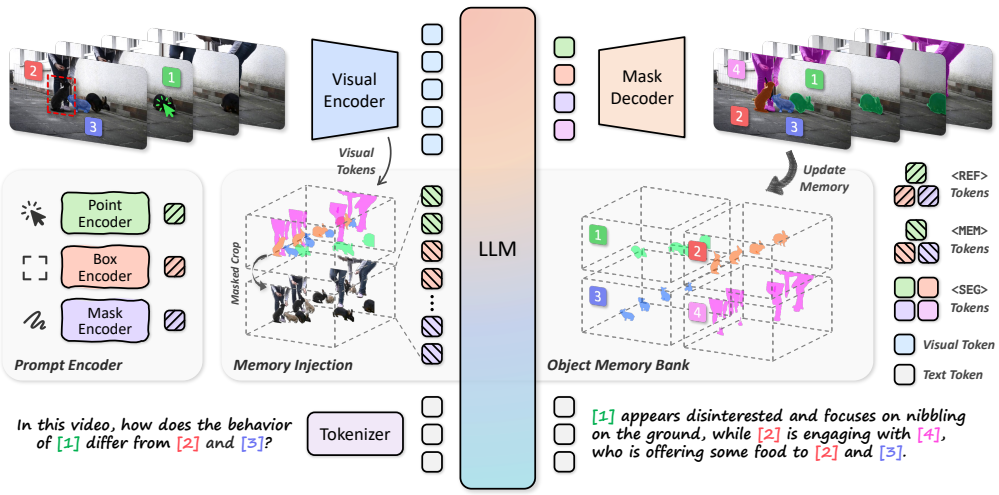

Figure 3: **The architecture of UniPixel.** Given a video, a question, and visual prompts, the model encodes them into tokens via the visual encoder, prompt encoder, and tokenizer, respectively, then predicts a spatial-temporal mask for each visual prompt via the mask decoder. The masks are updated into the object memory bank, and subsequently injected into the prompt for pixel-level reasoning.

3 Method

Problem Formulation We provide a unified definition for pixel-level reasoning tasks. Formally, the inputs are an image or a video $\mathcal{X}$, a text prompt $\mathcal{T}$, and N optional visual prompts $\{\mathcal{P}_i\}_{i=1}^{N}$ where each $\mathcal{P}_i$ could be a point, box, or mask on a specific frame. The outputs are textual responses to the prompt with K grounded spatial-temporal masks $\{\mathcal{M}_i\}_{i=1}^{K}$. Here, both N and K could be zero (degenerating to a normal visual understanding task) and K is not necessarily equal to N, as the model may segment extra objects/regions that are not specified by the visual prompts.

Overview Fig. 3 presents an overview of UniPixel. It is built upon the Qwen2.5-VL [3] framework, consisting of an LLM backbone and a ViT-based visual encoder that supports dynamic resolution inputs. Given a video and a text prompt, the model first tokenizes them via the visual encoder and text tokenizer, then sends them into the LLM for response generation. To boost this framework from holistic-level to pixel-level, we introduce (1) a *prompt encoder* (Sec.3.1) supporting three types of visual prompts, (2) an *object memory bank* (Sec.3.2) for storing object information and injecting it into the response generation process, and (3) a *mask decoder* (Sec.3.3) for generating spatial-temporal masks. We also extend the LLM's vocabulary by adding `<REF>`, `<MEM>`, and `<SEG>` tokens. The former two serve as placeholders in the input prompt that would be replaced by visual prompt and memory tokens, respectively, while the `<SEG>` token is utilized to trigger and guide the mask decoding process. Detailed designs and interactions among these components are illustrated as follows.

3.1 Prompt Encoder

This module aims to effectively encode each visual prompt into a single token that can be processed by the LLM. We denote a point prompt as a tuple (x, y, t) containing its spatial coordinates (x, y) and the corresponding frame index t. For box prompts, it is extended to (x_1, y_1, x_2, y_2, t) containing the positions of top-left and bottom-right corners. A mask prompt is densely represented by a 2D binary mask $\mathbf{m}_{ij} \in \{0, 1\}$ with the same shape as the encoded target frame.

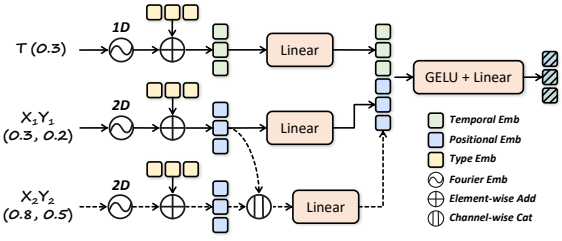

Figure 4: **Joint positional & temporal encoding** for point (X_1Y_1T) and box ($X_1Y_1X_2Y_2T$) prompts.

For sparse prompts (points and boxes), as shown in Fig. 4, we encode each position (x_i, y_i) as the sum of a 2D Fourier embedding [73] and a learnable type embedding (indicating whether it is a single point, top-left corner, or bottom-right corner). For box prompts, we merge the two positional embeddings

by concatenating them along the channel dimension and linearly projecting them back to the original size. Frame indices are also encoded similarly with 1D Fourier embeddings. The resulting positional and temporal embeddings are concatenated again, and then projected to the LLM's embedding space via a `GELU` → `Linear` block, such that the sparse coordinates in a point/box are encoded into a compact high-dimensional token. This design is inspired by [31, 66] with two key differences: (1) the spatial-only embeddings are extended to include temporal information, and (2) the negative points are discarded. For dense prompts (masks), we directly resize the binary masks and apply masked pooling on the outputs of the visual encoder. An M→L projector (`Linear` → `GELU` → `Linear`) is leveraged to project the pooled visual features to the LLM's embedding space.

3.2 Object Memory Bank

Although sparse prompts contain rich positional and temporal information indicating the objects that users are referring to, it is still hard for the model to focus on these important regions. Previous studies [12, 102, 103] also confirm that direct region cropping can generally provide better object awareness compared to positional pointers. To seamlessly integrate such a mechanism while preserving the flexibility of visual prompts (*e.g.*, allow pointing on a single frame instead of drawing complete masks on all frames), we propose an object memory bank to bridge sparse visual prompts and dense object masks. This is a hashmap where the keys are object IDs and the values are the corresponding spatial-temporal masks. It is initialized as an empty storage for every chat session, and is dynamically updated on demand. We define two operations for the object memory bank, namely *memory pre-filling* and *memory injection*. Below is an example of memory-enhanced multi-round conversation.

Prompt 1: How does the behavior of [1] <REF> differ from [2] <REF> and [3] <REF>?

<REF> detected, enhancing the prompt with object memory.

Memory Pre-filling Response:
The relevant regions for this question are [1] <SEG> [2] <SEG> [3] <SEG> [4] <SEG>. ← *4 objects saved into the memory*

Memory Injected Prompt:
Here is a video with 4 frames denoted as <1> to <4>. The highlighted regions are as follows:
[1]: <1> <MEM> <2> <MEM> <3> <MEM> ← *This object cannot be seen in the last frame*
[2]: <1> <MEM> <2> <MEM> <3> <MEM> <4> <MEM>
[3]: <1> <MEM> <2> <MEM> <3> <MEM> <4> <MEM>
[4]: <1> <MEM> <2> <MEM> <3> <MEM> <4> <MEM>
How does the behavior of [1] differ from [2] and [3]?

Response 1: [1] appears disinterested and focuses on nibbling on the ground, while [2] is engaging with [4], who is offering some food to [2] and [3].
Prompt 2: What food is [4] offering? ← *Users can directly refer to objects in the memory*
Response 2: [4] is offering carrots.

Memory Pre-filling This operation is triggered upon the detection of <REF> tokens in the input prompt, aiming to thoroughly analyze the referred objects and predict their corresponding masks. In this stage, the model responds with object IDs and <SEG> tokens for the relevant objects according to the context, and predicts their spatial-temporal masks accordingly. These object-mask pairs are then saved into the object memory bank.

Memory Injection We inject the features of the saved objects into the prompt to enhance object-awareness. Similar to the mask prompt encoder described in Sec. 3.1, each frame-level object mask is downsampled to match the resolution of visual tokens. We then apply masked pooling to aggregate object-centric features. Each frame-level mask is condensed into a single feature token, projected through the mask projector, and subsequently utilized to replace the corresponding <MEM> token in the memory-injected prompt. Through this *pre-filling and injection* mechanism, object-centric information is effectively integrated into the model inference process.

Why using object memory bank? An alternative is directly appending a <SEG> token to each <REF> token, followed by masked pooled features obtained during inference. However, we do not adopt this approach for two reasons: (1) During mask prediction, the <SEG> tokens, due to the unidirectional nature of causal self-attention, are unable to aggregate the full context of the prompt, thereby compromising the quality of predicted masks. (2) By utilizing the object memory bank, we can effectively decouple regional understanding and mask prediction, allowing each to benefit from referring and segmentation data during training, thus enhancing both capabilities.

Table 1: Comparison with state-of-the-art methods on ReVOS [96] `val` split. The best and second-best results are marked **bold** and underlined, respectively.

Method	Size	Referring			Reasoning			Overall			$\mathcal{R}$
		$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	
Non-LLM-based Specialists											
MTTR [5]	–	29.8	30.2	30.0	20.4	21.5	21.0	25.1	25.9	25.5	5.6
LMPM [21]	–	29.0	39.1	34.1	13.3	24.3	18.8	21.2	31.7	26.4	3.2
ReferFormer [85]	–	31.2	34.3	32.7	21.3	25.6	23.4	26.2	29.9	28.1	8.8
LLM-based Generalists											
LISA [32]	13B	45.2	47.9	46.6	34.3	39.1	36.7	39.8	43.5	41.6	8.6
TrackGPT [72]	13B	48.3	50.6	49.5	38.1	42.9	40.5	43.2	46.8	45.0	12.8
VISA [96]	13B	55.6	59.1	57.4	42.0	46.7	44.3	48.8	52.9	50.9	14.5
HyperSeg [81]	3B	56.0	60.9	58.5	50.2	55.8	53.0	53.1	58.4	55.7	–
InstructSeg [82]	3B	54.8	59.2	57.0	49.2	54.7	51.9	52.0	56.9	54.5	–
GLUS [39]	7B	56.0	60.7	58.3	48.8	53.9	51.4	52.4	57.3	54.9	17.9
ViLLa [112]	6B	–	–	–	–	–	–	54.9	59.1	57.0	–
Sa2VA [101]	4B	–	–	–	–	–	–	–	–	53.2	–
UniPixel (Ours)	3B	62.3	66.7	64.5	57.1	62.1	59.6	59.7	64.4	62.1	19.0
UniPixel (Ours)	7B	**63.9**	**67.8**	**65.8**	**59.4**	**63.7**	**61.5**	**61.7**	**65.7**	**63.7**	**19.4**

3.3 Mask Decoder

We adopt SAM 2.1 [66] as the mask decoder to disentangle the discrete language modeling and continuous mask prediction capabilities. For each `<SEG>` token, we extract its last-layer hidden states, downsample them via an L→M projector (architecturally identical to the M→L projector), and reshape them into two tokens. Using two tokens ensures better preservation of object information when downsampling from high- to low-dimensional channel space. These tokens prompt the mask decoder to predict the mask on the first frame, which is then propagated to the other frames.

3.4 Model Training

The training loss for UniPixel is a linear combination of language modeling loss and mask decoding losses [66], including a focal loss and dice loss for mask prediction, a mean-absolute-error (MAE) loss for IoU prediction, and a cross-entropy loss for objectness prediction. The loss weights are set to 1, 100, 5, 5, and 5, respectively. We train the model through a three-stage progressive alignment recipe. The datasets are listed in Tab. 12. In the first stage, we pre-train the sparse prompt encoder using 851K regional captioning data. Then, we align the LLM and mask decoder by training the L→M projector on 87K referring segmentation data. In the last stage, we further unfreeze the M→L projector and mask decoder, and apply LoRA [26] on the visual encoder and LLM. The model is jointly trained on a large-scale corpus with around 1M samples for diverse tasks.

4 Experiments

We evaluate the effectiveness of UniPixel by conducting extensive experiments across a diverse set of benchmarks. Specifically, we study the following research questions.

- **Q1.** Whether UniPixel is flexible and effective on basic image/video referring and segmentation tasks compared to the corresponding representative methods?
- **Q2.** How does it perform on the more challenging PixelQA task, which requires joint referring, segmentation, and question answering in videos?
- **Q3.** What effects does each architectural design contribute? More importantly, does the unified modeling of referring and segmentation lead to a mutual reinforcement effect?

Detailed information about the benchmarks, evaluation metrics, implementation details, and more experimental results can be found in the appendix.

4.1 Q1: Comparison with State-of-the-Arts on Referring and Segmentation Tasks

Reasoning Video Object Segmentation We begin with the most challenging ReVOS [96] dataset, which requires models to predict masks based on implicit text queries demanding complex reasoning abilities based on world knowledge. The results are shown in Tab. 1. Our 3B variant outperforms all

Table 2: Comparison with state-of-the-art methods on referring video object segmentation (RVOS) and motion-grounded video reasoning datasets, including MeViS [21] (`val`), Ref-YouTube-VOS [71] (`val`), Ref-DAVIS17 [62] (`val`), and GroundMoRe [20] (`test`). The best and second-best results are marked **bold** and underlined, respectively.

Method	Size	MeViS			Ref-YouTube-VOS			Ref-DAVIS17			GroundMoRe		
		$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$
Non-LLM-based Specialists													
ReferFormer [85]	–	29.8	32.2	31.0	61.3	64.6	62.9	58.1	64.1	61.1	11.2	14.3	12.7
LMPM [21]	–	34.2	40.2	37.2	–	–	–	–	–	–	12.7	14.0	13.3
OnlineRefer [83]	–	–	–	–	61.6	65.5	63.5	61.6	67.7	64.8	–	–	–
LLM-based Generalists													
PixelLM [69]	7B	36.3	41.1	38.7	54.3	55.7	55.0	63.4	70.0	66.7	9.9	10.0	10.0
LISA [32]	13B	35.8	40.0	37.9	54.0	54.8	54.4	63.2	68.8	66.0	6.3	6.7	6.5
VISA [96]	13B	41.8	47.1	44.5	61.4	64.7	63.0	67.0	73.8	70.4	5.3	4.7	5.9
VideoLISA [4]	3.8B	41.3	47.6	44.4	61.7	65.7	63.7	64.9	72.7	68.8	–	–	–
VideoGLaMM [56]	3.8B	42.1	48.2	45.2	65.4	68.2	66.8	73.3	65.6	69.5	–	–	–
ViLLa [112]	6B	46.5	52.3	49.4	64.6	70.4	67.5	70.6	78.0	74.3	–	–	–
GLUS [39]	7B	48.5	54.2	51.3	65.5	69.0	67.3	–	–	–	–	–	–
Sa2VA [101]	4B	–	–	46.2	–	–	70.0	–	–	73.8	–	–	–
MoRA [20]	7B	–	–	–	–	–	–	–	–	–	27.4	26.9	27.2
UniPixel (Ours)	3B	50.4	55.7	53.1	68.6	72.3	70.5	70.7	77.8	74.2	36.0	38.7	37.4
UniPixel (Ours)	7B	**53.2**	**58.3**	**55.8**	**69.5**	**72.4**	**71.0**	**72.7**	**80.1**	**76.4**	**36.5**	**39.1**	**37.8**

Table 3: Comparison with state-of-the-art methods on image referring expression segmentation (RES) and reasoning segmentation datasets, including RefCOCO/+/g [29, 55] and ReasonSeg [32] (`val`). The best and second-best results are marked **bold** and underlined, respectively.

Method	Size	RefCOCO			RefCOCO+			RefCOCOg		ReasonSeg	
		val	testA	testB	val	testA	testB	val(U)	test(U)	gIoU	cIoU
Non-LLM-based Specialists											
ReLA [41]	–	73.8	76.5	70.2	66.0	71.0	57.7	65.0	66.0	–	–
X-Decoder [116]	–	–	–	–	–	–	–	64.6	–	22.6	17.9
SEEM [117]	–	–	–	–	–	–	–	65.7	–	25.5	21.2
LLM-based Image Generalists											
NExT-Chat [104]	7B	74.7	78.9	69.5	65.1	71.9	56.7	67.0	67.0	–	–
PixelLM [69]	7B	73.0	76.5	68.2	66.3	71.7	58.3	69.3	70.5	–	–
LISA [32]	7B	74.9	79.1	72.3	65.1	70.8	58.1	67.9	70.6	61.3	62.9
Groundhog [110]	7B	78.5	79.9	75.7	70.5	75.0	64.9	74.1	74.6	56.2	–
LaSagnA [80]	7B	76.8	78.7	73.8	66.4	70.6	60.1	70.6	71.9	48.8	47.2
M^2SA [27]	13B	74.6	77.6	71.0	64.0	68.1	57.6	69.0	69.3	–	–
LLM-based Video Generalists											
VideoLISA [4]	3.8B	73.8	76.6	68.8	63.4	68.8	56.2	68.3	68.8	61.4	**67.1**
VISA [96]	7B	72.4	75.5	68.1	59.8	64.8	53.1	65.5	66.4	52.7	57.8
Vitron [23]	7B	75.5	79.5	72.2	66.7	72.5	58.0	67.9	68.9	–	–
Sa2VA [101]	4B	78.9	–	–	71.7	–	–	74.1	–	–	–
UniPixel (Ours)	3B	80.5	82.6	76.9	74.3	78.9	68.4	76.3	77.0	**64.0**	56.2
UniPixel (Ours)	7B	**80.8**	**83.0**	**77.4**	**75.3**	**80.1**	**70.0**	**76.4**	**77.1**	60.5	58.7

existing methods with larger LLMs (including Sa2VA-4B [101] also with SAM 2 decoder), achieving 62.1 overall $\mathcal{J}\&\mathcal{F}$. The 7B model further boosts the performance to 64.0 $\mathcal{J}\&\mathcal{F}$ – an improvement of 12% over the previous state-of-the-art – demonstrating that UniPixel can effectively understand implicit queries based on its world knowledge, and accurately generate masks as responses.

Referring Video Object Segmentation The performance comparisons on MeViS [21], Ref-YouTube-VOS [71], and Ref-DAVIS17 [62] datasets are presented in Tab. 2. UniPixel consistently achieves the best performance among its counterparts. Its advantage is particularly evident on the more challenging MeViS dataset, where our 3B model outperforms GLUS-7B [39] by around 3.5%, as well as the similarly sized VideoGLaMM-3.8B [56] by 17%. More experimental results on MeViS [21] `val`$^{\text{u}}$ set and Ref-SAV [101] `val` set are provided in Tab. 4 and Tab. 5, respectively. Ref-SAV features long referring descriptions, large object motion, large camera motion, and heavy occlusion compared with existing datasets. Given these complex descriptions and video content, our method consistently performs better than counterparts, including those fine-tuned on the target dataset.

Motion-Grounded Video Reasoning We also evaluate our method on GroundMoRe [20] dataset (results shown in Tab. 2), which highlights visual answer generation that requires joint spatial and

Table 4: Experimental results on MeViS [21] `val`$^{\text{u}}$ set. Post means applying post optimization.

Method	Size	FT	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$
LMPM [21]	–	✗	36.5	43.9	40.2
LISA [32]	7B	✗	39.9	46.5	43.2
LISA [32] + XMem [15]	7B	✗	41.9	49.3	45.6
VideoLISA [4]	7B	✗	48.4	54.9	51.7
VideoLISA [4] + Post	7B	✗	50.9	58.1	54.5
Sa2VA [101]	4B	✗	–	–	52.1
Sa2VA [101]	8B	✗	–	–	57.0
UniPixel (Ours)	3B	✗	56.1	63.2	59.7
UniPixel (Ours)	7B	✗	**58.4**	**65.0**	**61.7**

Table 5: Comparison on Ref-SAV [101] `val` set. FT means fine-tuning after pre-/co-training.

Method	Size	FT	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$
UniRef++ [86]	–	✗	11.6	9.5	10.5
UNINEXT [95]	–	✗	8.8	6.4	7.6
LMPM [21]	–	✗	12.2	9.8	10.3
VISA [96]	7B	✗	13.2	11.3	11.8
Sa2VA [101]	8B	✗	39.6	43.0	41.3
UniRef++ [86]	–	✓	15.8	13.4	14.6
Sa2VA [101]	8B	✓	48.3	51.7	50.0
UniPixel (Ours)	3B	✗	66.9	67.6	67.2
UniPixel (Ours)	7B	✗	**68.5**	**69.6**	**69.0**

Table 6: Fine-tuned performance on referring expression segmentation (RES) datasets, including RefCOCO/+/g [29, 55]. The best and second-best results are marked **bold** and underlined, respectively.

Method	Size	RefCOCO			RefCOCO+			RefCOCOg	
		val	testA	testB	val	testA	testB	val(U)	test(U)
LISA [32]	7B	74.9	79.1	72.3	65.1	70.8	58.1	67.9	70.6
GSVA [87]	7B	77.2	78.9	73.5	65.9	69.6	59.8	72.7	73.3
OMG-LLaVA [109]	7B	78.0	80.3	74.1	69.1	73.1	63.0	72.9	72.9
GLaMM [65]	7B	79.5	83.2	76.9	72.6	78.7	64.6	74.2	74.9
Sa2VA [101]	4B	80.4	–	–	74.3	–	–	75.7	–
UniPixel (Ours)	3B	81.9	83.5	78.6	75.3	80.3	70.6	77.2	78.5
UniPixel (Ours)	7B	**83.0**	**84.9**	**80.4**	**77.8**	**82.3**	**72.7**	**78.7**	**79.7**

Table 7: Experimental results on referring expression comprehension (REC) datasets, including RefCOCO/+/g [29, 55]. The best and second-best results are marked **bold** and underlined, respectively.

Method	Size	RefCOCO			RefCOCO+			RefCOCOg	
		val	testA	testB	val	testA	testB	val(U)	test(U)
OFA [78]	–	80.0	83.7	76.4	68.3	76.0	61.8	67.6	67.6
Shikra [9]	7B	87.0	90.6	80.2	81.6	87.4	72.1	82.3	82.2
MiniGPT-v2 [8]	7B	88.7	91.6	85.3	79.9	85.1	74.4	84.4	84.6
Vitron [23]	7B	90.9	93.2	**89.3**	83.7	89.1	76.9	86.4	87.0
UniPixel (Ours)	3B	91.8	93.8	87.5	86.3	90.8	80.3	88.0	88.2
UniPixel (Ours)	7B	**92.0**	**94.4**	88.1	**87.2**	**91.9**	**82.1**	**88.6**	**88.7**

temporal grounding. Note that we mainly compare the results with MoRA [20], which is fine-tuned on GroundMoRe while other methods are evaluated under the zero-shot setting. Benefit from the strong pixel-level reasoning capability, UniPixel significantly performs better than the baseline.

Referring Expression Segmentation and Reasoning Segmentation Tab. 3 compares the image segmentation capabilities using explicit and implicit queries. We evaluate our co-trained model on RefCOCO/+/g [29, 55] and ReasonSeg [32]. While state-of-the-art performance has been achieved on RES datasets, we observe that the reasoning segmentation data (239 samples) can be easily overwhelmed by the other samples during training due to its limited size. Tab. 6 presents the RES performance after fine-tuning. We follow the common practice that jointly fine-tunes the model on RefCOCO/+/g datasets [29, 55], and then evaluate on them separately. These results demonstrate the generalizability of UniPixel when facing both explicit and implicit queries.

Referring Expression Comprehension Our method also supports referring expression comprehension by inferring the bounding boxes from predicted masks. Its performance (accuracy with IoU $\geqslant 0.5$) is compared with representative methods in Tab. 7. Benefiting from the high-quality mask prediction, UniPixel can also achieve very competitive performance on this simpler task.

Referred Video Description and Question Answering We study UniPixel's regional understanding capabilities on VideoRefer-Bench [103], which contains two subsets for description and question answering tasks. The comparisons are in Tab. 8 and Tab. 9. BQ, SQ, RQ, CQ, and FP denote basic questions, sequential questions, relational questions, reasoning questions, and future predictions, respectively. Both tasks leverage mask prompts as inputs, where single-frame and multi-frame modes denote applying the masks only on a specific frame and on all frames, respectively. UniPixel can

Table 8: Comparison with state-of-the-art methods on VideoRefer-Bench^{D} [103]. The best and second-best results are marked **bold** and underlined, respectively.

Method	Size	Single-Frame					Multi-Frame				
		SC	AD	TD	HD	Avg.	SC	AD	TD	HD	Avg.
General LMMs											
LLaVA-OV [33]	7B	2.62	1.58	2.19	2.07	2.12	3.09	1.94	2.50	2.41	2.48
Qwen2-VL [77]	7B	2.97	2.24	2.03	2.31	2.39	3.30	2.54	2.22	2.12	2.55
InternVL2 [74]	26B	3.55	2.99	2.57	2.25	2.84	4.08	**3.35**	3.08	2.28	3.20
GPT-4o-mini [59]	–	3.56	2.85	2.87	2.38	2.92	3.89	3.18	2.62	2.50	3.05
GPT-4o [59]	–	3.34	2.96	3.01	2.50	2.95	4.15	3.31	3.11	2.43	3.25
Image Referring LMMs											
Ferret [98]	7B	3.08	2.01	1.54	2.14	2.19	3.20	2.38	1.97	1.38	2.23
Osprey [102]	7B	3.19	2.16	1.54	2.45	2.34	3.30	2.66	2.10	1.58	2.41
Video Referring LMMs											
Elysium [75]	7B	2.35	0.30	0.02	3.59	1.57	–	–	–	–	–
Artemis [63]	7B	–	–	–	–	–	3.42	1.34	1.39	2.90	2.26
VideoRefer [103]	7B	**4.41**	**3.27**	3.03	2.97	**3.42**	**4.44**	3.27	3.10	3.04	**3.46**
UniPixel (Ours)	3B	4.04	3.15	**3.10**	3.37	**3.42**	4.08	3.13	**3.13**	3.42	3.44
UniPixel (Ours)	7B	3.83	3.07	2.96	**3.62**	3.37	3.82	3.05	3.01	**3.57**	3.36

Table 9: Comparison with state-of-the-art methods on VideoRefer-Bench^{Q} [103] (*mask prompts*). MF denotes multi-frame mode. Full question types are in Sec. 4.1.

Method	Size	MF	BQ	SQ	RQ	CQ	FP	Avg.
General LMMs								
LLaVA-OV [33]	7B	✗	58.7	62.9	64.7	87.4	76.3	67.4
Qwen2-VL [77]	7B	✗	62.0	69.6	54.9	87.3	74.6	66.0
InternVL2 [74]	26B	✗	58.5	63.5	53.4	88.0	78.9	65.0
GPT-4o-mini [59]	–	✗	57.6	67.1	56.5	85.9	75.4	65.8
GPT-4o [59]	–	✗	62.3	**74.5**	**66.0**	88.0	73.7	71.3
Image Referring LMMs								
Ferret [98]	7B	✗	35.2	44.7	41.9	70.4	74.6	48.8
Osprey [102]	7B	✗	45.9	47.1	30.0	48.6	23.7	39.9
Video Referring LMMs								
VideoRefer [103]	7B	✗	**75.4**	68.6	59.3	**89.4**	78.1	71.9
UniPixel (Ours)	3B	✗	73.6	70.3	60.7	88.8	78.0	72.2
UniPixel (Ours)	7B	✗	68.9	73.1	64.7	88.8	**83.3**	**73.4**
VideoRefer [103]	7B	✓	–	70.6	60.5	–	–	72.1
UniPixel (Ours)	3B	✓	**75.3**	70.7	62.3	87.4	77.2	72.8
UniPixel (Ours)	7B	✓	70.6	**74.6**	**64.7**	**88.8**	**82.5**	**74.1**

Table 10: Evaluation results on our newly introduced PixelQA task. All the visual prompts are applied in a single frame. See Sec. 4.2 for detailed settings.

Method	Size	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	Acc
Point Prompts					
InternVL2 [74]	26B	–	–	–	60.8
Qwen2-VL [77]	72B	–	–	–	69.3
UniPixel (Ours)	3B	**57.3**	**64.4**	**60.9**	71.1
UniPixel (Ours)	7B	42.1	47.1	44.6	**71.4**
Box Prompts					
InternVL2 [74]	26B	–	–	–	61.3
Qwen2-VL [77]	72B	–	–	–	69.0
UniPixel (Ours)	3B	**57.8**	**64.7**	**61.3**	70.3
UniPixel (Ours)	7B	41.1	46.4	43.8	**71.4**
Mixed (50% Points + 50% Boxes)					
InternVL2 [74]	26B	–	–	–	60.9
Qwen2-VL [77]	72B	–	–	–	69.1
UniPixel (Ours)	3B	**57.2**	**64.1**	**60.6**	70.8
UniPixel (Ours)	7B	42.3	47.5	44.9	**71.4**

effectively comprehend both types of prompts, and accurately respond with object-centric descriptions or answers, surpassing strong models including GPT-4o [59] and VideoRefer [103].

4.2 Q2: Pixel-Level Video Question Answering (PixelQA)

We design the new PixelQA task based on VideoRefer-Bench^{Q} [103], where the original mask prompts are replaced with more challenging point or box prompts. Given these ambiguous visual cues, models are expected to correctly identify the target object according to the question and the visual prompt, then respond with **both the textual answer and the corresponding object masks**. We report the mask prediction $\mathcal{J}\&\mathcal{F}$ and MCQ accuracy in Tab. 10. Note that none of the existing methods supports this scenario. Thus, we apply set-of-mark prompts [97] directly on video frames, and evaluate the QA accuracies of two strong LMMs [77, 74] as our baselines. Aside from point- or box-only prompts, we also explore a more flexible setting that randomly chooses different prompts for different objects. The results verify that our ***memory pre-filling & injection*** paradigm effectively enhances the model's reasoning capabilities. Visualizations of this task are shown in Fig. 5.

4.3 Q3: Key Ablation Studies

Effect of Task Unification We study the effect of task unification in Tab. 11 (a). Unifying referring and segmentation capabilities into a single model and training them jointly leads to better results

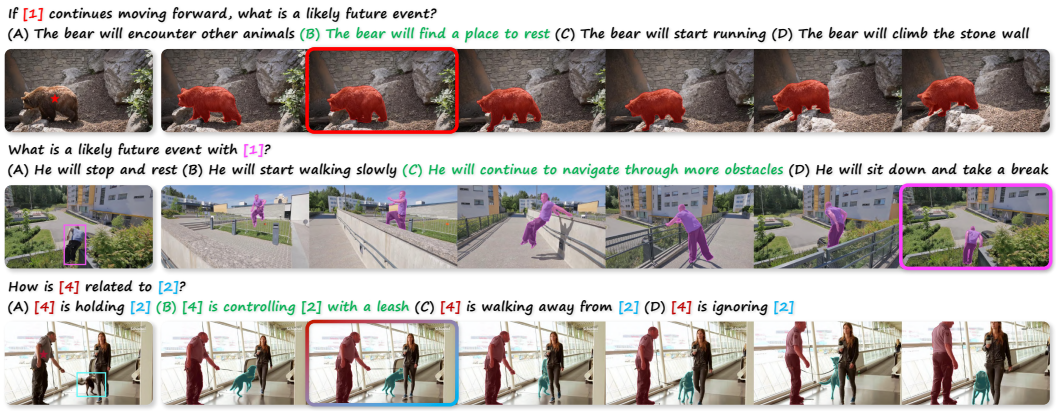

Figure 5: **Visualization of the outputs from UniPixel on PixelQA task**. Star marks and boxes refer to point and box prompts, respectively. The boxed frames denote where the visual prompts are applied. Given different types of visual prompts on a single frame, our method can flexibly infer the relevant object, track it across the entire video, and involve its features in reasoning.

Table 11: Key ablation studies with UniPixel-3B on PixelQA (*mixed*). See Sec. 4.3 for explanations.

(a) Task Unification

Refer	Segment	Memory	$\mathcal{J}\&\mathcal{F}$	Acc
✓			–	64.6
	✓		47.5	–
✓	✓		48.2	67.4
✓	✓	✓	**49.0**	**68.5**

(b) Object Memory Bank

Referring Method	$\mathcal{J}\&\mathcal{F}$	Acc
① <REF>	46.8	64.5
② <REF><SEG>	47.8	64.9
③ <REF><SEG> + Pooling	47.5	66.3
④ **Object Memory Bank**	**49.0**	**68.5**

(c) Prompt Encoder & Mask Decoder

Encoder	Decoder	$\mathcal{J}\&\mathcal{F}$	Acc
w/o Time	–	44.3	63.7
w/ Time	–	**49.0**	**68.5**
–	Independent	46.1	66.2
–	Propagation	**49.0**	**68.5**

on both tasks (first three rows), demonstrating the **mutual reinforcement effect** of such unification. Incorporating memory pre-filling as an auxiliary task (last row) brings extra improvements.

Effect of Object Memory Bank Tab. 11 (b) verifies the effectiveness of object memory bank. ① means using a single token for each referred object. ② means adding an extra segmentation token to segment it as an auxiliary task. ③ further appends masked-pooled visual tokens after it. The results show that (1) both adding auxiliary segmentation task and masked-pooled features help regional understanding, and (2) decoupling them via object memory bank can further boost the performance.

Design Space of Prompt Encoder & Mask Decoder We compare different prompt encoder and mask decoder designs in Tab. 11 (c). The performance significantly drops when the temporal encoding in the prompt encoder is removed (first two rows). For the mask decoder (last two rows), we explore an alternative strategy that treats video frames independently (as batched images), which could largely accelerate inference but lead to sub-optimal accuracies. We hypothesize that this is because the LLM-generated <SEG> token cannot well-capture the object information in all frames, thus disentangling the segmentation and tracking capabilities to an external module is reasonable.

5 Conclusion

In this work, we proposed **UniPixel**, a large multi-modal model that supports flexible pixel-level visual reasoning. It unifies the internal representations of referred and segmented objects through a novel **object memory bank**. We observe that by such unification, the performance of object referring and segmentation can be jointly enhanced. Extensive experiments on diverse pixel-level understanding tasks, including the **PixelQA** task, demonstrate the significance of the proposed method. We hope this work inspires future advancements in pixel-level visual understanding.

Acknowledgements

This study was supported by The Hong Kong RGC Grant (15229423) and a financial support from ARC Lab, Tencent PCG (ZGG9). We also acknowledge The University Research Facility in Big Data Analytics (UBDA) at The Hong Kong Polytechnic University for providing computing resources that have contributed to the research results reported within this paper.

References

[1] Ali Athar, Xueqing Deng, and Liang-Chieh Chen. Vicas: A dataset for combining holistic and pixel-level video understanding using captions with grounded segmentation. *arXiv:2412.09754*, 2024.

[2] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. 2023.

[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:2502.13923*, 2025.

[4] Zechen Bai, Tong He, Haiyang Mei, Pichao Wang, Ziteng Gao, Joya Chen, Zheng Zhang, and Mike Zheng Shou. One token to seg them all: Language instructed reasoning segmentation in videos. In *NeurIPS*, pages 6833–6859, 2024.

[5] Adam Botach, Evgenii Zheltonozhskii, and Chaim Baskin. End-to-end referring video object segmentation with multimodal transformers. In *CVPR*, pages 4985–4995, 2022.

[6] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, pages 961–970, 2015.

[7] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *CVPR*, pages 1209–1218, 2018.

[8] Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechun Liu, Pengchuan Zhang, Raghuraman Krishnamoorthi, Vikas Chandra, Yunyang Xiong, and Mohamed Elhoseiny. Minigpt-v2: large language model as a unified interface for vision-language multi-task learning. *arXiv:2310.09478*, 2023.

[9] Keqin Chen, Zhao Zhang, Weili Zeng, Richong Zhang, Feng Zhu, and Rui Zhao. Shikra: Unleashing multimodal llm's referential dialogue magic. *arXiv:2306.15195*, 2023.

[10] Xianjie Chen, Roozbeh Mottaghi, Xiaobai Liu, Sanja Fidler, Raquel Urtasun, and Alan Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts. In *CVPR*, pages 1971–1978, 2014.

[11] Yuan Chen, Zi-han Ding, Ziqin Wang, Yan Wang, Lijun Zhang, and Si Liu. Asynchronous large language model enhanced planner for autonomous driving. In *ECCV*, pages 22–38, 2024.

[12] Zewen Chen, Juan Wang, Wen Wang, Sunhan Xu, Hang Xiong, Yun Zeng, Jian Guo, Shuxun Wang, Chunfeng Yuan, Bing Li, othersShilong 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. *arXiv:2307.03601*, 2023.

[13] 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:2412.05271*, 2024.

[14] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Zhong Muyan, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. *arXiv:2312.14238*, 2023.

[15] Ho Kei Cheng and Alexander G Schwing. Xmem: Long-term video object segmentation with an atkinson-shiffrin memory model. In *ECCV*, pages 640–658, 2022.

[16] Can Cui, Yunsheng Ma, Xu Cao, Wenqian Ye, Yang Zhou, Kaizhao Liang, Jintai Chen, Juanwu Lu, Zichong Yang, Kuei-Da Liao, et al. A survey on multimodal large language models for autonomous driving. In *WACV*, pages 958–979, 2024.

[17] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. In *NeurIPS*, 2024.

[18] Google DeepMind. Introducing gemini 2.0: our new ai model for the agentic era, 2024.

[19] Google DeepMind. Gemini 2.5: Our most intelligent ai model, 2025.

[20] Andong Deng, Tongjia Chen, Shoubin Yu, Taojiannan Yang, Lincoln Spencer, Yapeng Tian, Ajmal Saeed Mian, Mohit Bansal, and Chen Chen. Motion-grounded video reasoning: Understanding and perceiving motion at pixel level. *arXiv:2411.09921*, 2024.

[21] 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*, pages 2694–2703, 2023.

[22] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, et al. Palm-e: An embodied multimodal language model. *arXiv:2303.03378*, 2023.

[23] Hao Fei, Shengqiong Wu, Hanwang Zhang, Tat-Seng Chua, and Shuicheng Yan. Vitron: A unified pixel-level vision llm for understanding, generating, segmenting, editing. *arXiv:2412.19806*, 2024.

[24] 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. *arXiv:2405.21075*, 2024.

[25] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv:2501.12948*, 2025.

[26] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 2022.

[27] Donggon Jang, Yucheol Cho, Suin Lee, Taehyeon Kim, and Dae-Shik Kim. Mmr: A large-scale benchmark dataset for multi-target and multi-granularity reasoning segmentation. *arXiv:2503.13881*, 2025.

[28] Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. Tgif-qa: Toward spatio-temporal reasoning in visual question answering. In *CVPR*, pages 2758–2766, 2017.

[29] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. Referitgame: Referring to objects in photographs of natural scenes. In *EMNLP*, pages 787–798, 2014.

[30] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv:2406.09246*, 2024.

[31] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *ICCV*, pages 4015–4026, 2023.

[32] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. In *CVPR*, pages 9579–9589, 2024.

[33] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv:2408.03326*, 2024.

[34] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, pages 19730–19742, 2023.

[35] 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:2305.06355*, 2023.

[36] 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 *CVPR*, pages 22195–22206, 2024.

[37] Zongzhao Li, Zongyang Ma, Mingze Li, Songyou Li, Yu Rong, Tingyang Xu, Ziqi Zhang, Deli Zhao, and Wenbing Huang. Star-r1: Spacial transformation reasoning by reinforcing multimodal llms. *arXiv:2505.15804*, 2025.

[38] Bin Lin, Bin Zhu, Yang Ye, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. *arXiv:2311.10122*, 2023.

[39] Lang Lin, Xueyang Yu, Ziqi Pang, and Yu-Xiong Wang. Glus: Global-local reasoning unified into a single large language model for video segmentation. *arXiv:2504.07962*, 2025.

[40] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755, 2014.

[41] Chang Liu, Henghui Ding, and Xudong Jiang. Gres: Generalized referring expression segmentation. In *CVPR*, pages 23592–23601, 2023.

[42] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *CVPR*, pages 26296–26306, 2024.

[43] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, pages 34892–34916, 2023.

[44] Ruyang Liu, Chen Li, Haoran Tang, Yixiao Ge, Ying Shan, and Ge Li. St-llm: Large language models are effective temporal learners. In *ECCV*, pages 1–18, 2024.

[45] Ye Liu, Jixuan He, Wanhua Li, Junsik Kim, Donglai Wei, Hanspeter Pfister, and Chang Wen Chen. r^2-tuning: Efficient image-to-video transfer learning for video temporal grounding. In *ECCV*, 2024.

[46] Ye Liu, Huifang Li, Chao Hu, Shuang Luo, Yan Luo, and Chang Wen Chen. Learning to aggregate multi-scale context for instance segmentation in remote sensing images. *IEEE Transactions on Neural Networks and Learning Systems*, 36(1):595–609, 2024.

[47] Ye Liu, Siyuan Li, Yang Wu, Chang Wen Chen, Ying Shan, and Xiaohu Qie. Umt: Unified multi-modal transformers for joint video moment retrieval and highlight detection. In *CVPR*, pages 3042–3051, 2022.

[48] Ye Liu, Kevin Qinghong Lin, Chang Wen Chen, and Mike Zheng Shou. Videomind: A chain-of-lora agent for long video reasoning. *arXiv:2503.13444*, 2025.

[49] Ye Liu, Zongyang Ma, Zhongang Qi, Yang Wu, Ying Shan, and Chang W Chen. E.t. bench: Towards open-ended event-level video-language understanding. In *NeurIPS*, pages 32076–32110, 2024.

[50] Ye Liu, Junsong Yuan, and Chang Wen Chen. Consnet: Learning consistency graph for zero-shot human-object interaction detection. In *ACM MM*, pages 4235–4243, 2020.

[51] Zongyang Ma, Yuxin Chen, Ziqi Zhang, Zhongang Qi, Chunfeng Yuan, Shaojie Zhu, Chengxiang Zhuo, Bing Li, Ye Liu, Zang Li, Ying Shan, and Weiming Hu. Visionmath: Vision-form mathematical problem-solving. In *ICCV*, 2025.

[52] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Khan. Videogpt+: Integrating image and video encoders for enhanced video understanding. *arXiv:2406.09418*, 2024.

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

[54] Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. Egoschema: A diagnostic benchmark for very long-form video language understanding. In *NeurIPS*, 2024.

[55] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *CVPR*, pages 11–20, 2016.

[56] 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:2411.04923*, 2024.

[57] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *ICCV*, pages 4990–4999, 2017.

[58] OpenAI. Gpt-4v(ision) system card, 2023.

[59] OpenAI. Gpt-4o system card, 2024.

[60] OpenAI. Openai o1 system card, 2024.

[61] Wujian Peng, Lingchen Meng, Yitong Chen, Yiweng Xie, Yang Liu, Tao Gui, Hang Xu, Xipeng Qiu, Zuxuan Wu, and Yu-Gang Jiang. Inst-it: Boosting multimodal instance understanding via explicit visual prompt instruction tuning. *arXiv:2412.03565*, 2024.

[62] 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:1704.00675*, 2017.

[63] 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 *NeurIPS*, 2024.

[64] Vignesh Ramanathan, Anmol Kalia, Vladan Petrovic, Yi Wen, Baixue Zheng, Baishan Guo, Rui Wang, Aaron Marquez, Rama Kovvuri, Abhishek Kadian, et al. Paco: Parts and attributes of common objects. In *CVPR*, pages 7141–7151, 2023.

[65] 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*, pages 13009–13018, 2024.

[66] 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:2408.00714*, 2024.

[67] Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv:2403.05530*, 2024.

[68] Shuhuai Ren, Linli Yao, Shicheng Li, Xu Sun, and Lu Hou. Timechat: A time-sensitive multimodal large language model for long video understanding. In *CVPR*, pages 14313–14323, 2024.

[69] Zhongwei Ren, Zhicheng Huang, Yunchao Wei, Yao Zhao, Dongmei Fu, Jiashi Feng, and Xiaojie Jin. Pixellm: Pixel reasoning with large multimodal model. In *CVPR*, pages 26374–26383, 2024.

[70] Chaitanya Ryali, Yuan-Ting Hu, Daniel Bolya, Chen Wei, Haoqi Fan, Po-Yao Huang, Vaibhav Aggarwal, Arkabandhu Chowdhury, Omid Poursaeed, Judy Hoffman, et al. Hiera: A hierarchical vision transformer without the bells-and-whistles. In *ICML*, pages 29441–29454, 2023.

[71] Seonguk Seo, Joon-Young Lee, and Bohyung Han. Urvos: Unified referring video object segmentation network with a large-scale benchmark. In *ECCV*, pages 208–223, 2020.

[72] Nicholas Stroh. Trackgpt–a generative pre-trained transformer for cross-domain entity trajectory forecasting. *arXiv:2402.00066*, 2024.

[73] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In *NeurIPS*, pages 7537–7547, 2020.

[74] OpenGVLab Team. Internvl2: Better than the best—expanding performance boundaries of open-source multimodal models with the progressive scaling strategy, 2024.

[75] Han Wang, Yongjie Ye, Yanjie Wang, Yuxiang Nie, and Can Huang. Elysium: Exploring object-level perception in videos via mllm. In *ECCV*, pages 166–185, 2024.

[76] 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*, pages 4057–4066, 2023.

[77] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv:2409.12191*, 2024.

[78] Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In *ICML*, pages 23318–23340, 2022.

[79] Shihao Wang, Zhiding Yu, Xiaohui Jiang, Shiyi Lan, Min Shi, Nadine Chang, Jan Kautz, Ying Li, and Jose M Alvarez. Omnidrive: A holistic llm-agent framework for autonomous driving with 3d perception, reasoning and planning. *arXiv:2405.01533*, 2024.

[80] Cong Wei, Haoxian Tan, Yujie Zhong, Yujiu Yang, and Lin Ma. Lasagna: Language-based segmentation assistant for complex queries. *arXiv:2404.08506*, 2024.

[81] Cong Wei, Yujie Zhong, Haoxian Tan, Yong Liu, Zheng Zhao, Jie Hu, and Yujiu Yang. Hyperseg: Towards universal visual segmentation with large language model. *arXiv:2411.17606*, 2024.

[82] Cong Wei, Yujie Zhong, Haoxian Tan, Yingsen Zeng, Yong Liu, Zheng Zhao, and Yujiu Yang. Instructseg: Unifying instructed visual segmentation with multi-modal large language models. *arXiv:2412.14006*, 2024.

[83] Dongming Wu, Tiancai Wang, Yuang Zhang, Xiangyu Zhang, and Jianbing Shen. Onlinerefer: A simple online baseline for referring video object segmentation. In *ICCV*, pages 2761–2770, 2023.

[84] Jianlong Wu, Wei Liu, Ye Liu, Meng Liu, Liqiang Nie, Zhouchen Lin, and Chang Wen Chen. A survey on video temporal grounding with multimodal large language model. *arXiv:2508.10922*, 2025.

[85] Jiannan Wu, Yi Jiang, Peize Sun, Zehuan Yuan, and Ping Luo. Language as queries for referring video object segmentation. In *CVPR*, pages 4974–4984, 2022.

[86] Jiannan Wu, Yi Jiang, Bin Yan, Huchuan Lu, Zehuan Yuan, and Ping Luo. Segment every reference object in spatial and temporal spaces. In *ICCV*, pages 2538–2550, 2023.

[87] Zhuofan Xia, Dongchen Han, Yizeng Han, Xuran Pan, Shiji Song, and Gao Huang. Gsva: Generalized segmentation via multimodal large language models. In *CVPR*, pages 3858–3869, 2024.

[88] Linshan Xie, Xuehong Lin, Xunda Wang, Junjian Wen, Teng Ma, Jiahao Hu, Peng Cao, Alex TL Leong, and Ed X Wu. Simultaneous eeg-fmri reveals spontaneous neural oscillatory activity in cingulate cortex underlying transient rsfmri network dynamics. In *ISMRM*, 2024.

[89] Linshan Xie, Xunda Wang, Xuehong Lin, Teng Ma, Junjian Wen, Peng Cao, Alex TL Leong, and Ed X Wu. Single-pulse optogenetic perturbation of thalamo-cortical networks reveals functional architecture of rsfmri networks. In *ISMRM*, 2024.

[90] Linshan Xie, Xunda Wang, Xuehong Lin, Junjian Wen, Teng Ma, Alex TL Leong, and Ed X Wu. Brain-wide resting-state fmri network dynamics elicited by activation of single thalamic input. *Nature Communications*, 2025.

[91] Linshan Xie, Xunda Wang, Teng Ma, Pit Shan Chong, Lee Wei Lim, Peng Cao, Pek-Lan Khong, Ed X Wu, and Alex TL Leong. Are topographically segregated excitatory neurons in visual thalamus functionally diverse? an optogenetic fmri study. In *ISMRM*, 2022.

[92] Linshan Xie, Xunda Wang, Teng Ma, Hang Zeng, Junjian Wen, Peng Cao, Ed X Wu, and Alex TL Leong. Short single pulse optogenetic fmri mapping of downstream targets in thalamo-cortical pathways. In *ISMRM*, 2023.

[93] 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 *ACM MM*, pages 1645–1653, 2017.

[94] 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:2404.16994*, 2024.

[95] Bin Yan, Yi Jiang, Jiannan Wu, Dong Wang, Ping Luo, Zehuan Yuan, and Huchuan Lu. Universal instance perception as object discovery and retrieval. In *CVPR*, pages 15325–15336, 2023.

[96] 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. In *ECCV*, pages 98–115, 2024.

[97] 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:2310.11441*, 2023.

[98] 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:2310.07704*, 2023.

[99] Samson Yu, Kelvin Lin, Anxing Xiao, Jiafei Duan, and Harold Soh. Octopi: Object property reasoning with large tactile-language models. *arXiv:2405.02794*, 2024.

[100] 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 *AAAI*, pages 9127–9134, 2019.

[101] Haobo Yuan, Xiangtai Li, Tao Zhang, Zilong Huang, Shilin Xu, Shunping Ji, Yunhai Tong, Lu Qi, Jiashi Feng, and Ming-Hsuan Yang. Sa2va: Marrying sam2 with llava for dense grounded understanding of images and videos. *arXiv:2501.04001*, 2025.

[102] 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 *CVPR*, pages 28202–28211, 2024.

[103] 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. *arXiv:2501.00599*, 2024.

[104] Ao Zhang, Yuan Yao, Wei Ji, Zhiyuan Liu, and Tat-Seng Chua. Next-chat: An lmm for chat, detection and segmentation. *arXiv:2311.04498*, 2023.

[105] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv:2306.02858*, 2023.

[106] Jiawei Zhang, Chejian Xu, and Bo Li. Chatscene: Knowledge-enabled safety-critical scenario generation for autonomous vehicles. In *CVPR*, pages 15459–15469, 2024.

[107] Peirong Zhang, Haowei Xu, Jiaxin Zhang, Guitao Xu, Xuhan Zheng, Zhenhua Yang, Junle Liu, Yuyi Zhang, and Lianwen Jin. Aesthetics is cheap, show me the text: An empirical evaluation of state-of-the-art generative models for ocr. *arXiv:2507.15085*, 2025.

[108] Peirong Zhang, Jiaxin Zhang, Jiahuan Cao, Hongliang Li, and Lianwen Jin. Smaller but better: Unifying layout generation with smaller large language models. *International Journal of Computer Vision*, 133:3891–3917, 2025.

[109] Tao Zhang, Xiangtai Li, Hao Fei, Haobo Yuan, Shengqiong Wu, Shunping Ji, Chen Change Loy, and Shuicheng Yan. Omg-llava: Bridging image-level, object-level, pixel-level reasoning and understanding. In *NeurIPS*, pages 71737–71767, 2024.

[110] Yichi Zhang, Ziqiao Ma, Xiaofeng Gao, Suhaila Shakiah, Qiaozi Gao, and Joyce Chai. Groundhog: Grounding large language models to holistic segmentation. In *CVPR*, pages 14227–14238, 2024.

[111] Xufeng Zhao, Mengdi Li, Cornelius Weber, Muhammad Burhan Hafez, and Stefan Wermter. Chat with the environment: Interactive multimodal perception using large language models. In *IROS*, pages 3590–3596, 2023.

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

[113] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, pages 633–641, 2017.

[114] Luowei Zhou, Chenliang Xu, and Jason Corso. Towards automatic learning of procedures from web instructional videos. In *AAAI*, 2018.

[115] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv:2304.10592*, 2023.

[116] Xueyan Zou, Zi-Yi Dou, Jianwei Yang, Zhe Gan, Linjie Li, Chunyuan Li, Xiyang Dai, Harkirat Behl, Jianfeng Wang, Lu Yuan, et al. Generalized decoding for pixel, image, and language. In *CVPR*, pages 15116–15127, 2023.

[117] Xueyan Zou, Jianwei Yang, Hao Zhang, Feng Li, Linjie Li, Jianfeng Wang, Lijuan Wang, Jianfeng Gao, and Yong Jae Lee. Segment everything everywhere all at once. In *NeurIPS*, pages 19769–19782, 2023.

Appendix

In this appendix, we provide more details about the training data, model implementation, and experimental settings to complement the main paper. Additional analysis, ablation studies, visualizations, and discussions are also incorporated. Below is the table of contents.

A. Model

  1. Implementation Details

  2. Training Recipe

B. Experiments

  1. Tasks and Benchmarks

  2. Evaluation Metrics

  3. More Experimental Results

  4. Ablation Studies

  5. Qualitative Results

C. Discussions

  1. Limitations & Future Work

  2. Potential Societal Impacts

D. Licenses

A Model

A.1 Implementation Details

We instantiate our base models with 3B and 7B versions of Qwen2.5-VL [3]. Both variants employ pre-trained SAM 2.1 [66] with Hiera Base+ [70] backbone as the mask decoder. The M→L projector is initialized with the weights from the V→L projector of Qwen2.5-VL. The hidden size inside the prompt encoder is 256. To reduce GPU memory and accelerate training, we randomly sample 8 frames per video, with each frame resized to $316^2 \sim 448^2$ pixels ($128 \sim 256$ tokens per frame). The frame sampling strategies follow the specifications of each benchmark during inference. The mask decoder has a fixed resolution of 768×768. For each segmentation sample, up to 5 objects are randomly selected to compute the mask prediction losses. During training, LoRA adapters [26] with `rank=128` and `alpha=256` are applied to all `QKVO` layers in the visual encoder and LLM. The input sequences are restricted to 4K tokens. We train the model with 8 RTX A6000 Ada (48G) GPUs, with a global batch size of 256 for stages 1 and 2, and 32 for stage 3. In the first two stages, the learning rates are set to 1e-3. In the last stage, it is set to 5e-6 for the mask decoder and 2e-5 for all the other parameters, respectively. A linear warmup in the first 3% steps followed by cosine decay is adopted in all stages. The configurations of datasets are introduced in the following section.

A.2 Training Recipe

The detailed distribution of training datasets for UniPixel is shown in Tab. 12. Within the three-stage training recipe, we first pre-train the sparse prompt encoder using short caption samples from Inst-IT [61] and VideoRefer [103]. For each sample, we randomly select a point inside the ground truth mask (50%) or generate an augmented box from it (50%). This stage aims to enable the model with simple visual prompt comprehension and regional captioning capabilities on images and videos. In the second stage, we align the LLM and mask decoder using referring object segmentation datasets [29, 55, 71]. We use short caption/query samples for the first two stages to focus on alignment rather than knowledge learning. For the last stage, we collect a large-scale, high-quality corpus called `UniPixel-SFT-1M`[3] to jointly train the model on diverse pixel-level tasks. The original annotations have been rewritten using task-specific templates to incorporate instructions. All the repurposed datasets and pre-processing pipelines will be publicly available to facilitate future research.

[3] `https://huggingface.co/datasets/PolyU-ChenLab/UniPixel-SFT-1M`

Table 12: The distribution of training datasets for UniPixel. We use different background colors to denote object referring, object segmentation, regional understanding, memory pre-filling, and general video understanding data, respectively.

Stage	Dataset	Inputs						Outputs		#Samples	#Repeat	Ratio
		Text	Image	Video	Point	Box	Mask	Text	Mask			
1	Inst-IT-Image-Short-Caption [61]	✓	✓		✓	✓		✓		351K	1	41.2%
	VideoRefer-Short-Caption [103]	✓		✓	✓	✓		✓		500K	1	58.8%
2	RefCOCO [29]	✓	✓					✓	✓	17K	5	20.8%
	RefCOCO+ [29]	✓	✓					✓	✓	17K	5	20.8%
	RefCOCOg [55]	✓	✓					✓	✓	22K	5	26.8%
	RefClef [29]	✓	✓					✓	✓	18K	5	22.0%
	Ref-YouTube-VOS [71]	✓		✓				✓	✓	13K	3	9.5%
3	Osprey-Conversation [102]	✓	✓				✓	✓		1.4K	5	0.1%
	Osprey-Detail-Description [102]	✓	✓				✓	✓		29K	5	2.5%
	Osprey-Pos-Neg [102]	✓	✓				✓	✓		20K	5	1.7%
	VideoRefer-Detailed-Caption [103]	✓		✓			✓	✓		120K	5	10.1%
	VideoRefer-QA [103]	✓		✓			✓	✓		69K	5	5.8%
	Inst-IT-Video-QA [61]	✓		✓			✓	✓		159K	5	13.4%
	VideoRefer-QA-Memory [103]	✓		✓	✓	✓		✓	✓	69K	3	3.5%
	Inst-IT-QA-Memory [61]	✓		✓	✓	✓		✓	✓	158K	3	8.0%
	RefCOCO [29]	✓	✓					✓	✓	17K	10	2.9%
	RefCOCO+ [29]	✓	✓					✓	✓	17K	10	2.9%
	RefCOCOg [55]	✓	✓					✓	✓	22K	10	3.7%
	RefClef [29]	✓	✓					✓	✓	18K	10	3.0%
	ReasonSeg [32]	✓	✓					✓	✓	1.6K	10	0.3%
	ADE20K [113]	✓	✓					✓	✓	20K	3	1.0%
	COCOStuff [7]	✓	✓					✓	✓	118K	3	6.0%
	Mapillary Vistas [57]	✓	✓					✓	✓	18K	3	0.9%
	PACO-LVIS [64]	✓	✓					✓	✓	46K	3	2.3%
	PASCAL-Part [10]	✓	✓					✓	✓	4.4K	3	0.2%
	Ref-YouTube-VOS [71]	✓		✓				✓	✓	13K	5	1.1%
	Ref-DAVIS17 [62]	✓		✓				✓	✓	0.6K	10	0.1%
	Ref-SAV [101]	✓		✓				✓	✓	56K	3	2.8%
	MeViS [21]	✓		✓				✓	✓	23K	5	1.9%
	LV-VIS [76]	✓		✓				✓	✓	11K	3	0.6%
	ViCaS [1]	✓		✓				✓	✓	41K	3	2.1%
	ReVOS [96]	✓		✓				✓	✓	29K	5	2.5%
	GroundMoRe [20]	✓		✓				✓	✓	5.6K	3	0.3%
	LLaVA-1.5-Mix-665K [42]	✓	✓					✓		647K	1	10.9%
	VideoGPT+ Instruct [52]	✓		✓				✓		573K	1	9.7%

B Experiments

B.1 Tasks and Benchmarks

Our method is extensively evaluated across 9 fine-grained image/video understanding tasks. The benchmark(s) used for each task are listed as follows:

1. **Reasoning Video Object Segmentation:** ReVOS [96]
2. **Referring Video Object Segmentation:** MeViS [21], Ref-YouTube-VOS [71], Ref-DAVIS17 [62], Ref-SAV [101]
3. **Motion-Grounded Video Reasoning:** GroundMoRe [20]
4. **Referring Expression Segmentation:** RefCOCO [29], RefCOCO+ [29], RefCOCOg [55]
5. **Reasoning Segmentation:** ReasonSeg [32]
6. **Referring Expression Comprehension:** RefCOCO [29], RefCOCO+ [29], RefCOCOg [55]
7. **Referred Video Description:** VideoRefer-Bench$^{\text{D}}$ [103]
8. **Referred Video Question Answering:** VideoRefer-Bench$^{\text{Q}}$ [103]
9. **Flexible Pixel-Level Understanding:** PixelQA (Ours)

B.2 Evaluation Metrics

For video segmentation tasks, we adopt $\mathcal{J}\&\mathcal{F}$ as the main metric to jointly consider region similarity $\mathcal{J}$ and contour accuracy $\mathcal{F}$. Image segmentation is evaluated using cIoU (the cumulative intersection over the cumulative union) and gIoU (the average of all per-image IoUs) following existing work. For referred video description and question answering tasks, we follow the official evaluation protocols to report GPT-4o [59] scores and MCQ accuracy, respectively. For referring expression comprehension,

Table 13: Performance comparison on general video question answering (VideoQA) on MVBench [36]. Note that UniPixel is the only model supporting pixel-level referring & segmentation.

Model	Size	AS	AP	AA	FA	UA	OE	OI	OS	MD	AL	ST	AC	MC	MA	SC	FP	CO	EN	ER	CI	Avg.
GPT-4V [58]	–	55.5	63.5	72.0	46.5	73.5	18.5	59.0	29.5	12.0	40.5	83.5	39.0	12.0	22.5	45.0	47.5	52.0	31.0	**59.0**	11.0	43.5
Video-ChatGPT [53]	7B	23.5	26.0	62.0	22.5	26.5	54.0	28.0	40.0	23.0	20.0	31.0	30.5	25.5	39.5	48.5	29.0	33.0	29.5	26.0	35.5	32.7
Video-LLaMA [105]	7B	27.5	25.5	51.0	29.0	39.0	48.0	40.5	38.0	22.5	22.5	43.0	34.0	22.5	32.5	45.5	32.5	40.0	30.0	21.0	37.0	34.1
VideoChat [35]	7B	33.5	26.5	56.0	33.5	40.5	53.0	40.5	30.0	25.5	27.0	48.5	35.0	20.5	42.5	46.0	26.5	41.0	23.5	23.5	36.0	35.5
Video-LLaVA [38]	7B	46.0	42.5	56.5	39.0	53.5	53.0	48.0	41.0	29.0	31.5	82.5	45.0	26.0	53.0	41.5	33.5	41.5	27.5	38.5	31.5	43.0
TimeChat [68]	7B	40.5	36.0	61.0	32.5	53.0	53.5	41.5	29.0	19.5	26.5	66.5	34.0	20.0	43.5	42.0	36.5	36.0	29.0	35.0	35.0	38.5
PLLaVA [94]	7B	58.0	49.0	55.5	41.0	61.0	56.0	61.0	36.0	23.5	26.0	82.0	39.5	42.0	52.0	45.0	42.0	53.5	30.5	48.0	31.0	46.6
ST-LLM [44]	7B	66.0	53.5	**84.0**	44.0	58.5	80.5	73.5	38.5	42.5	31.0	86.5	36.5	56.5	78.5	43.0	44.5	46.5	34.5	41.5	58.5	54.9
VideoGPT+ [52]	4B	69.0	60.0	83.0	48.5	66.5	85.5	**75.5**	36.0	44.0	34.0	89.5	39.5	71.0	90.5	45.0	53.0	50.0	29.5	44.0	60.0	58.7
VideoChat2 [36]	7B	**75.5**	58.0	83.5	**50.5**	60.5	87.5	74.5	**45.0**	47.5	**44.0**	82.5	37.0	64.5	87.5	51.0	**66.5**	47.0	**35.0**	37.0	72.5	60.4
UniPixel (Ours)	3B	69.5	62.5	83.0	48.5	76.5	86.5	66.5	38.0	49.0	40.5	87.0	**49.0**	**74.0**	**95.0**	49.0	45.0	63.5	34.5	58.0	**73.5**	62.5
UniPixel (Ours)	7B	71.0	**68.0**	**84.0**	45.0	**78.0**	**91.5**	66.5	35.5	**57.5**	43.0	**91.5**	47.0	73.5	92.5	**58.0**	53.0	**74.0**	**37.5**	49.0	69.0	**64.3**

Table 14: Effectiveness justification of multi-stage training. The best and second-best results are marked **bold** and underlined, respectively. The three-stage recipe leads to optimal performance.

Stage 1	Stage 2	Stage 3	ReVOS			MeViS (val^u)			VideoRefer-Bench^Q	
			$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	Single-Frame	Multi-Frame
		✓	58.3	63.6	61.0	54.8	61.9	58.4	71.1	71.5
✓		✓	59.0	63.4	61.2	55.2	62.1	58.7	71.8	72.3
	✓	✓	59.6	63.5	61.6	55.7	62.5	59.1	71.2	71.6
✓	✓	✓	**59.7**	**64.4**	**62.1**	**56.1**	**63.2**	**59.7**	**72.2**	**72.8**

we leverage mean accuracies, where a predicted bounding box is considered correct when it has the intersection over union (IoU) with the ground truth no less than 0.5.

B.3 More Experimental Results

General Video Question Answering We also evaluate UniPixel on MVBench [36] to compare its general video understanding capabilities with existing methods. The results are illustrated in Tab. 13. Note that our method is the only one in the table that supports referring and segmentation. By jointly training on holistic-level and pixel-level data, UniPixel can effectively balance the capabilities under both scenarios, demonstrated by the strong performance compared with holistic-level models.

B.4 Ablation Studies

Effect of Multi-stage Training We investigate the effectiveness of multi-stage training in Tab. 14. As shown in the first line, directly training the model using large-scale data only leads to sub-optimal performance, due to the unaligned representations among prompt encoder, LLM, and mask decoder. We observe that pre-training either the sparse prompt encoder or the L→M projector (the second and third lines) brings performance gains on both tasks (referring and segmentation). We hypothesize that this is because pre-aligning either of them can alleviate the burden of joint-task learning in stage 3. The last row verifies that the performance can be further boosted by pre-aligning both of them.

Number of Hidden Tokens for Mask Decoder As mentioned in the main paper, there is a huge gap between the feature dimensions of the LLM and the mask decoder, thus splitting the `<SEG>` token into more hidden tokens can better preserve the object information from the LLM. We ablate this mechanism in Tab. 15. According to the results, using only 1 hidden token cannot fully preserve the object information, as the mask prediction performance is sub-optimal. However, we also observe that using more than 2 hidden tokens (*e.g.*, 4 or 8) only brings negligible performance gain. Therefore, we choose 2 hidden tokens per object in our final model.

Training Strategy for the M→L projector The M→L projector aims to project the masked-pooled object-centric features to the LLM's embedding space. Since the object features originate from the visual encoder, it is possible to re-use the pre-trained weights of the original V→L projector in Qwen2.5-VL. Its effects are studied in Tab. 16. We investigated two strategies: 1) re-using the weights and 2) adding an extra pre-training stage for better alignment. The comparison shows that directly re-using weights without extra pre-training can achieve the best results.

Table 15: Ablation study on the number of hidden tokens for each <SEG>. Performance gains are negligible with more than 2 tokens/object.

#Tokens	ReVOS			MeViS (val^u)		
	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$
1	59.6	63.5	61.6	55.8	62.5	59.2
2	**59.7**	**64.4**	**62.1**	56.1	**63.2**	59.7
4	59.8	63.9	61.9	**56.8**	63.1	**59.9**
8	59.5	64.0	61.8	56.4	62.8	59.6

Table 16: Ablation study on M→L projector. Init and PT denote weight initialization from V→L projector and extra pre-training, respectively.

Init	PT	VideoRefer-Bench^Q		PixelQA
		Single-Frame	Multi-Frame	Mixed Acc
	✓	71.4	71.9	67.7
✓		71.5	71.7	67.4
✓	✓	**72.4**	72.6	68.2
		72.2	**72.8**	**68.5**

Table 17: Ablation study on training data used in stage 3. The best and second-best results are marked **bold** and underlined, respectively. Gradually adding more pixel-level data brings performance gains.

Regional	Segmentation	Memory	General	ReVOS			MeViS (val^u)			VideoRefer-Bench^Q	
				$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	Single-Frame	Multi-Frame
✓				–	–	–	–	–	–	72.1	72.0
	✓			58.9	63.8	61.4	56.0	63.2	59.6	–	–
✓	✓			59.2	63.7	61.5	55.8	63.1	59.5	72.3	72.6
✓	✓	✓		59.6	**64.5**	**62.1**	**56.3**	**63.5**	**59.9**	**72.4**	72.5
✓	✓	✓	✓	**59.7**	64.4	**62.1**	56.1	63.2	59.7	72.2	**72.8**

Combination of Training Data Tab. 17 studies the effect of the combination of multi-task co-training data in stage 3. Compared with training only on the regional or segmentation data, leveraging both of them leads to considerable performance on both tasks. Incorporating memory pre-filling data (requiring both referring and segmentation) can further boost the performance. We also mix some general holistic-level video understanding data to preserve the original capabilities of the pre-trained model, while it slightly affects the performance on pixel-level tasks.

B.5 Qualitative Results

Fig. 6 ∼ 11 present more visualizations of outputs from UniPixel on different pixel-level understanding tasks. Our method can effectively handle flexible visual prompts [103], implicit queries [32, 96], long queries [101], and motion-grounded questions [20].

C Discussion

C.1 Limitations & Future Work

Due to the limited computing resources, we did not further scale up the training data to incorporate more pixel-level tasks such as grounded caption generation (GCG) on images [65] or videos [56], which are interesting scenarios and their data may bring more performance gains. Besides, the mask decoder currently predicts the first mask on the first frame and propagates it to the following frames, while it potentially supports predicting on the best frame (defined as the frame with the best view of the target) and propagates it to both sides of the video. We will focus in our future work to explore more pixel-level understanding tasks and more flexible mechanisms for the mask decoder.

C.2 Potential Societal Impacts

This work introduces a new framework for pixel-level visual-language understanding, which could potentially be used in education, surveillance, and healthcare industries, where flexible interactions with the users and fine-grained understanding of images & videos are required. In other scenarios requiring multi-modal assistants, our method can also serve as a more advanced alternative. To the best of our knowledge, there are no potential negative societal impacts to declare.

D Licenses

Our model is built based on the pre-trained Qwen2.5-VL [3] and SAM 2.1 [66] models. They are both licensed under the Apache License 2.0 (https://www.apache.org/licenses/LICENSE-2.0).

Figure 6: Visualization of the predictions from UniPixel on PixelQA.

Figure 7: Visualization of the predictions from UniPixel on ReVOS [96].

Figure 8: Visualization of the predictions from UniPixel on Ref-DAVIS17 [62].

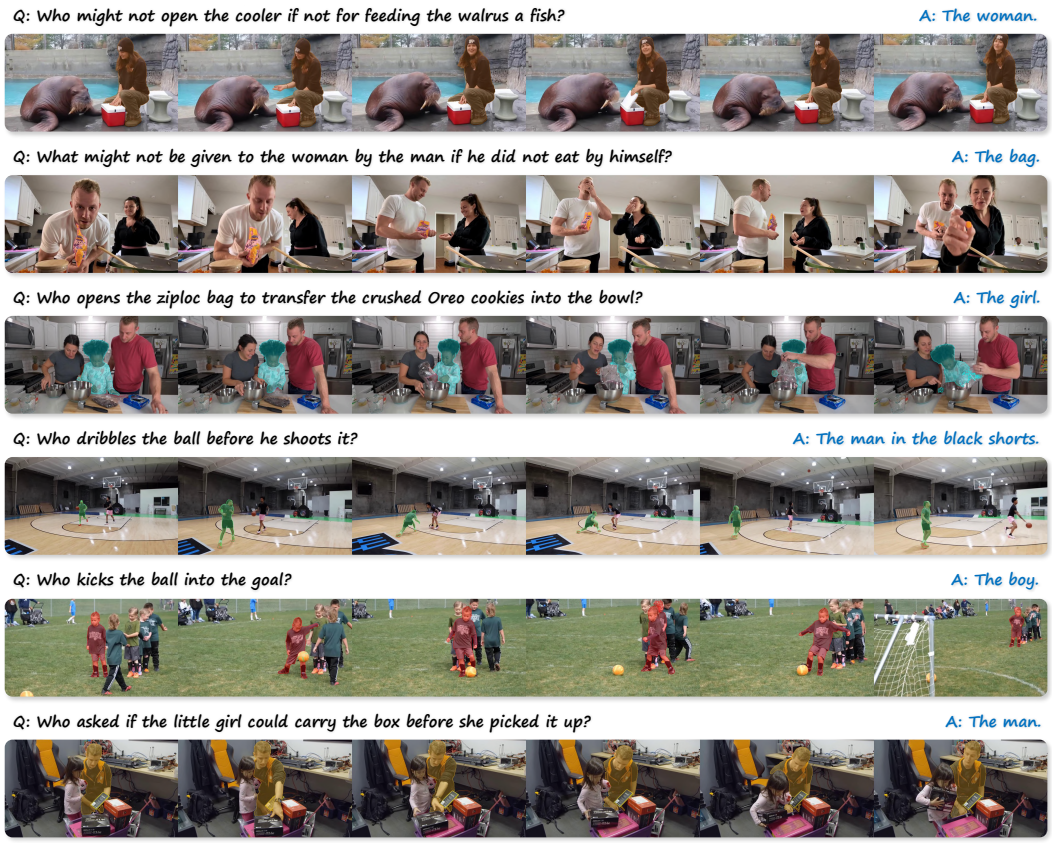

Figure 9: Visualization of the predictions from UniPixel on GroundMoRe [20].

Figure 10: Visualization of the predictions from UniPixel on Ref-SAV [101].

Figure 11: Visualization of the predictions from UniPixel on ReasonSeg [32].

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: To the best of our knowledge, all the claims made in the abstract and introduction are verified either by referring to previous studies or by conducting experiments.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations of this work are carefully discussed in the appendix.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: The paper does not include theoretical results. All the claims shall be verified through ablation studies.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We clearly present the list of datasets used for training, the implementation details, and the evaluation metrics we used to produce the experimental results in the appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We open-source all the code, checkpoints, data, and training logs to ensure full reproducibility.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: All the detailed settings, hyperparameters, and other necessary information are provided in the appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: We are not able to fully report the error bars given the limited computing resources.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: This information is included in the implementation details in the appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We confirm that the research conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The potential societal impacts are discussed in the appendix.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We provided citations to the original papers for all the used datasets and base models.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.