

Copyright Undertaking

This thesis is protected by copyright, with all rights reserved.

By reading and using the thesis, the reader understands and agrees to the following terms:

1. The reader will abide by the rules and legal ordinances governing copyright regarding the use of the thesis.
2. The reader will use the thesis for the purpose of research or private study only and not for distribution or further reproduction or any other purpose.
3. The reader agrees to indemnify and hold the University harmless from and against any loss, damage, cost, liability or expenses arising from copyright infringement or unauthorized usage.

IMPORTANT

If you have reasons to believe that any materials in this thesis are deemed not suitable to be distributed in this form, or a copyright owner having difficulty with the material being included in our database, please contact lbsys@polyu.edu.hk providing details. The Library will look into your claim and consider taking remedial action upon receipt of the written requests.

TOWARDS INTERACTIVE MULTI-MODAL CONTENT UNDERSTANDING

YE LIU

PhD

The Hong Kong Polytechnic University

2026

The Hong Kong Polytechnic University
Department of Computing

Towards Interactive Multi-modal
Content Understanding

Ye Liu

A thesis submitted in partial fulfillment of the requirements for
the degree of Doctor of Philosophy
August 2025

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgment has been made in the text.

Signature: _____

Name of Student: Ye Liu

Abstract

Human cognition process is inherently interactive, involving a dynamic interplay between perception and reasoning. In contrast, contemporary multi-modal AI systems typically operate under a single-pass processing paradigm, where perception and reasoning are performed sequentially and passively. While this approach has proven effective for constrained, well-defined tasks, it exhibits significant limitations in both efficiency and generalizability when applied to complex, open-ended scenarios. In such cases, this content-as-input paradigm struggles due to fundamental constraints in the single-pass pipeline, which hampers the comprehensive encoding of information-dense content for diverse downstream objectives.

To address this challenge, we propose a new paradigm called interactive multi-modal content understanding, in which multi-modal content is treated not as static input but as an interactive environment. Given a task instruction (*e.g.*, a question about the content), the AI agent actively explores this environment, iteratively engaging in perception and reasoning to derive the final answer. This paradigm draws inspiration from human cognition, facilitating deeper exploration and more nuanced analysis during the reasoning process.

This thesis begins with a systematic discussion on the proposed paradigm, followed by in-depth analysis of the technical challenges it presents and our strategies to address them. Specifically, **Part I** investigates the essential capabilities of video temporal perception, including visual-audio-text integration, effective video temporal

grounding, and techniques for transferring these capabilities from images to videos. **Part II** focuses on combining fine-grained perception with open-ended reasoning through large language models, with particular attention to temporal and spatio-temporal reasoning. Finally, **Part III** synthesizes the previously discussed technologies, culminating in the development of an agentic, interactive multi-modal content understanding system capable of planning, grounding, verification, and question answering to tackle complex scenarios. Concluding remarks and future research directions are summarized at the end.

Publications Arising from the Thesis

Conference Papers

1. **Ye Liu**, Siyuan Li, Yang Wu[†], Chang Wen Chen, Ying Shan, Xiaohu Qie. “UMT: Unified Multi-modal Transformers for Joint Video Moment Retrieval and Highlight Detection”. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.
2. **Ye Liu**, Jixuan He, Wanhua Li[†], Junsik Kim, Donglai Wei, Hanspeter Pfister, Chang Wen Chen[†]. “R²-Tuning: Efficient Image-to-Video Transfer Learning for Video Temporal Grounding”. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024.
3. **Ye Liu**, Zongyang Ma, Zhongang Qi[†], Yang Wu, Ying Shan, Chang Wen Chen[†]. “E.T. Bench: Towards Open-Ended Event-Level Video-Language Understanding”. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.
4. **Ye Liu**, Zongyang Ma, Junfu Pu, Zhongang Qi, Yang Wu, Ying Shan, Chang Wen Chen[†]. “UniPixel: Unified Object Referring and Segmentation for Pixel-Level Visual Reasoning”. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.
5. **Ye Liu**^{*}, Kevin Qinghong Lin^{*}, Chang Wen Chen[†], Mike Zheng Shou[†]. “Video-Mind: A Chain-of-LoRA Agent for Temporal-Grounded Video Reasoning”. In *LAW: Bridging Language, Agent, and World Models @ NeurIPS*, 2025.

Journal Articles

1. **Ye Liu**, Huifang Li[†], Chao Hu, Shuang Luo, Yan Luo, Chang Wen Chen. “Learning to Aggregate Multi-Scale Context for Instance Segmentation in Remote Sensing Images”. *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, 2024.
2. Jianlong Wu[†], Wei Liu, **Ye Liu**, Meng Liu, Liqiang Nie, Zhouchen Lin, Chang Wen Chen. “A Survey on Video Temporal Grounding with Multimodal Large Language Model”. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2025.

Patents

1. Chang Wen Chen, **Ye Liu**, Ying Shan, Siyuan Li, Yang Wu, Xiaohu Qie. “A Video Recognition Method, System, Storage Medium, and Server”. *CN Patent (CN114581821A)*, 2022.
2. Chang Wen Chen, **Ye Liu**, Zongyang Ma, Zhongang Qi, Yang Wu, Ying Shan. “Video-Language Processing Method, Model Training Method, and Device”. *CN Patent (202410866001.8)*, 2024.
3. Chang Wen Chen, **Ye Liu**, Zongyang Ma, Junfu Pu, Zhongang Qi, Yang Wu, Ying Shan. “Multimedia Processing Method, Apparatus, Device, Storage Medium, and Program Product”. *CN Patent (202510942903.X)*, 2025.

Acknowledgments

None of this would have been possible without the work of so many.

First and foremost, I would like to express my deepest gratitude to my supervisor, Prof. Chang Wen Chen, for accepting me as his first PhD student in Hong Kong. I'm indebted to his intellectual guidance, continuous support, and generous patience throughout this remarkable journey. He has consistently encouraged me to think deep, aim high, and focus on the fundamental problems. His big vision and strong heart have molded me into the researcher I am today.

I am also sincerely grateful to my thesis committee members, Prof. Jianfei Cai, Prof. Shiqi Wang, and Prof. Jing Li, for their valuable time, insightful comments, and encouragement, which have greatly strengthened this work.

My heartfelt appreciation goes to Dr. Yang Wu and Dr. Zhongang Qi, who are my mentors during my internship at Tencent ARC Lab. Most of my works would not have been possible without their insightful suggestions and generous support. Their continued guidance throughout my study life has been invaluable – not only in research, but also in shaping me as a person through their own humility.

I was fortunate to have the opportunity to visit and collaborate with research groups around the world. I'm profoundly grateful to Prof. Hanspeter Pfister for hosting me at Harvard University in 2023, and to Prof. Mike Zheng Shou for hosting me at the National University of Singapore in 2024. Both visits were memorable and enriching. I will never forget the moments I spent in Boston and Singapore.

I also have special thanks to my teachers and mentors who have been instrumental in inspiring my research journey. I'm especially thankful to Prof. Junsong Yuan for co-supervising me with Prof. Chen during my internship in Buffalo and for guiding me in publishing my first research paper. I'm also grateful to Prof. Ye Mang for his generous advice and support on my career. My gratitude goes to Prof. Chi Guo for sparking my interest in scientific research during my undergraduate years. Finally, I would like to thank Prof. Gui-Song Xia and Prof. Huifang Li for supervising my undergraduate theses and for their continuous guidance beyond research.

Being part of the Chen Lab was a wonderful and unforgettable experience. To all my labmates, thank you for making the past four years so delightful: Lingbo Liu, Fengbo Lan, Zuyao Chen, Xiaotong Yu, Fan Wang, Bruce X.B. Yu, Dingbang Liu, Zongyan Han, Guanchen Ding, Tianyu Huang, Tao Yang, Yinhi Kwok, Junfan Lin, Juwei Shen, Yang Mi, Hu Zhu, Yaosi Hu, Ruixiang Jiang, Bing Wu, Chenwu Zhang, Yuang Chen, Nannan Wu, Yuntao Wei, Tianyu Zhang, Muyao Niu, Yanshuo Wang, and Junyan Lin. Huge thanks also to my collaborators and peers: Zongyang Ma, Wanhua Li, Donglai Wei, Jixuan He, Ruxun Xiang, Qinyue Zheng, Kevin Qinghong Lin, and more. These memories and friendships will forever stay in my heart.

I am also grateful to my dear friends: Xingguang Ji, Qihao Chen, Yuanyuan Tang, Xi He, Lu Yu, Yunwei Zhu, Yao Yao, Shang Wu, Xu Xu, and many more. You have been with me through all the ups and downs. Thank you for being my comfort, my safe place, and a constant source of joy. I'm so lucky to have you in my life.

Finally, I owe my warmest gratitude to my family. To Dr. Linshan Xie, thanks for always being by my side. You have been my source of strength and comfort, reminding me to keep going even when the road seemed long. This journey would not have been possible without you. To my mom and dad, thank you for your unwavering love, faith, and patience in me. Your guidance and sacrifices have laid the foundation for every achievement I have made. From the values you taught me to the countless ways you have supported my dreams, I am forever grateful.

To my family, whose unwavering love carried me through this journey.

Table of Contents

Abstract	i
Publications Arising from the Thesis	iii
Acknowledgments	v
List of Figures	xv
List of Tables	xxi
1 Introduction	1
1.1 What Do We Mean by “Interactive”?	3
1.2 Perception: Basic Information Acquisition	4
1.3 Reasoning: Open-Ended Task-Solving	5
1.4 Interaction: Proactive Sensing & Thinking	6
Part I Perception	8
2 Unified Multi-modal Transformers for Video Content Analytics	9

2.1 Introduction	10
2.2 Related Work	12
2.2.1 Video Moment Retrieval & Highlight Detection	12
2.2.2 Text Query Based Models	13
2.2.3 Multi-modal Learning	13
2.3 Method	14
2.3.1 Overview	14
2.3.2 Uni-modal Encoder	15
2.3.3 Cross-modal Encoder	16
2.3.4 Query Generator	18
2.3.5 Query Decoder and Prediction Heads	19
2.4 Experiments	20
2.4.1 Datasets and Experimental Settings	20
2.4.2 Results on Joint Moment Retrieval & Highlight Detection	24
2.4.3 Results on Video Moment Retrieval	24
2.4.4 Results on Video Highlight Detection	25
2.4.5 Ablation Studies	25
2.4.6 Visualization	28
2.5 Summary	31
2.6 Limitations & Future Work	32
3 Efficient Image-to-Video Transfer Learning for Temporal Grounding	34

3.1 Introduction	35
3.2 Related Work	38
3.2.1 CLIP for Video Understanding	38
3.2.2 Video Temporal Grounding	39
3.2.3 Parameter-Efficient Transfer Learning	40
3.3 Method	40
3.3.1 Problem Formulation	40
3.3.2 Preliminary: The CLIP Model	41
3.3.3 Overview	41
3.3.4 Reversed Recurrent Tuning	42
3.3.5 Prediction Heads	46
3.3.6 Training and Inference	47
3.3.7 Discussion: Detailed Comparison with Counterparts	48
3.4 Experiments	50
3.4.1 Datasets & Evaluation Metrics	50
3.4.2 Implementation Details	53
3.4.3 Comparison with State-of-the-Arts	54
3.4.4 Ablation Studies	56
3.4.5 Visualization	60
3.5 Summary	60
3.6 Limitations & Future Work	64

Part II Reasoning	65
4 Event-Level & Time-Sensitive Video-Language Understanding	66
4.1 Introduction	67
4.2 Related Work	69
4.2.1 Video Large Language Models.	69
4.2.2 Benchmarks for Video-LLMs	70
4.3 E.T. Bench	70
4.3.1 Hierarchical Task Taxonomy	71
4.3.2 Data Collection and Annotation	73
4.3.3 Benchmark Analysis	80
4.4 Our Method	82
4.4.1 Model	83
4.4.2 Instruction-Tuning Dataset	87
4.5 Experiments	90
4.5.1 Evaluation Settings	90
4.5.2 Implementation Details	92
4.5.3 Baselines	93
4.5.4 Main Results	94
4.5.5 Ablation Studies	97
4.5.6 Visualizations	100
4.6 Summary	101

4.7	Limitations & Future Work	102
4.8	Licenses	102
5	Unified Referring and Segmentation for Pixel-Level Visual Reasoning	107
5.1	Introduction	108
5.2	Related Work	112
5.2.1	Large Multi-modal Models	112
5.2.2	Visual Referring and Segmentation	112
5.3	Method	113
5.3.1	Problem Formulation	113
5.3.2	Overview	113
5.3.3	Prompt Encoder	114
5.3.4	Object Memory Bank	115
5.3.5	Mask Decoder	117
5.3.6	Model Training	118
5.4	Experiments	118
5.4.1	Implementation Details	119
5.4.2	Training Datasets	119
5.4.3	Evaluation Metrics	121
5.4.4	Q1: Comparison with State-of-the-Arts	122
5.4.5	Q2: Pixel-Level Video Question Answering (PixelQA)	126
5.4.6	Q3: Ablation Studies	127

5.4.7 Visualizations	131
5.5 Summary	132
5.6 Limitations & Future Work	133
Part III Interaction	137
6 A Chain-of-LoRA Agent for Interactive Video Understanding	138
6.1 Introduction	139
6.2 Related Work	141
6.2.1 Temporal-grounded Video Understanding	141
6.2.2 Multi-modal Reasoning	142
6.2.3 Inference-time Searching	143
6.3 Method	143
6.3.1 Overview	143
6.3.2 Planner	144
6.3.3 Grounder	146
6.3.4 Verifier	149
6.3.5 Answerer	150
6.3.6 Chain-of-LoRA	151
6.4 Experiments	151
6.4.1 Benchmarks and Settings	152
6.4.2 Implementation Details	153
6.4.3 Q1: Comparison with State-of-the-Arts	154

6.4.4 Q2: The Advantages of Chain-of-LoRA	158
6.4.5 Q3: Ablation Studies	158
6.4.6 Visualization	161
6.5 Summary	162
6.6 Limitations & Future Work	163
7 Conclusion	165
7.1 Potential Future Impact	166
7.1.1 Impact on Research Directions	166
7.1.2 Impact on Real-World Applications	167
7.2 Future Work	168
References	169

List of Figures

1.1	A roadmap towards interactive multi-modal content understanding.	
	In this thesis, we comprehensively discuss and introduce a step-by-	
	step path towards developing an interactive content understanding	
	system.	4
2.1	The proposed UMT is a unified and flexible framework which can	
	handle different input modality combinations, and output video mo-	
	ment retrieval and/or highlight detection results (marked by red	
	rectangles and golden stars, respectively). Note that different text	
	queries lead to different outcomes from the same video. Arrows in	
	different colors denote different input-output combinations.	10
2.2	Overall architecture of our framework. When either the video or au-	
	dio is unavailable, the corresponding uni-modal encoder and cross-	
	modal encoder are deactivated. If text queries are not provided, the	
	model would simply use learnable moment queries instead.	14
2.3	The architecture of bottleneck transformer module. We introduce	
	bottleneck tokens for cross-modal feature compression and expan-	
	sion, largely reducing the computational cost.	17

2.4	Visualizations on QVHighlights. The predicted moments and saliency scores are shown by brackets and lines. (a) All the highlight clips are presented, indicating that UMT can learn implicit correlations between video content and query semantics. (b) Different modality combinations guide the model pay attention to different moments. (c) Our model can handle multiple queries in a single video.	29
2.5	More visualization results on QVHighlights val split.	30
2.6	Failure cases on QVHighlights val split.	31
3.1	Video temporal grounding (VTG) contains three video-language understanding problems, <i>i.e.</i> , moment retrieval (MR), highlight detection (HD), and video summarization (VS).	36
3.2	Different architectural designs for CLIP-based image-to-video transfer learning. The gray rectangle in (d) denotes the progressively refined spatial-temporal features.	37
3.3	Moment retrieval mAP with different backbones on QVHighlights val split. CLIP’s potential of temporal modeling was not fully exploited. .	38
3.4	Overall architecture of our framework. The input video and query are first encoded by frozen CLIP encoders. Their multi-layer outputs are then recurrently fused and refined by a learnable R^2 Block to construct spatial-temporal representations h , which would be scaled up/down to construct a temporal feature pyramid, followed by three heads for MR, HD, and VS, respectively.	42
3.5	Detailed architecture of the R^2 Block. It can be split into two parts: (a) <i>query-modulated spatial pooling</i> , and (b) <i>recurrent temporal refinement</i> . Note that the [CLS] tokens of visual features are omitted for clarity.	43

3.6	Different feature refinement strategies for MR on QVHighlights val split. Single means using the K -th layer only. Mean denotes averaging the features from last K layers. Forward and Reversed indicate different refinement directions.	56
3.7	Visualization of the attention maps for query-modulated spatial pooling. Diverse regions are learned for different queries.	56
3.8	Cosine similarities of video (x-axis) and text (y-axis) among 7~12 CLIP layers.	56
3.9	Visualization of joint moment retrieval and highlight detection results on QVHighlights val split. We compare our method with UniVTG, which is a strong baseline leveraging both additional temporal backbone and large-scale post pre-training. R^2 -Tuning can accurately regress the boundaries of moments and predict highlight saliency scores through its novel feature refinement design.	61
3.10	More visualizations of joint moment retrieval and highlight detection results on QVHighlights val split. R^2 -Tuning can accurately regress the boundaries of moments and predict highlight saliency scores.	62
3.11	Failure cases of predictions from R^2 -Tuning on QVHighlights val split.	63
4.1	Task definitions in E.T. Bench. The 12 tasks derives from 4 essential capabilities for time-sensitive video understanding: <i>referring</i> , <i>grounding</i> , <i>dense captioning</i> , and <i>complex understanding</i> .	68
4.2	Left: Task taxonomy and sample distribution. Right: Generation pipeline for E.T. Bench. We conduct a thorough process of pre-filtering, annotation repurposing, instruction writing, manual check, and sampling to obtain high-quality fine-grained annotations. Details are discussed in Section 4.3.	74

4.3	Left: Word cloud of text queries shows a considerable degree of diversity. Right: Distribution of averaged video durations (in seconds) across 12 tasks.	80
4.4	Frequency distribution of verbs (top 25 / 461) in E.T. Bench. The x- and y-axes denote the verbs and their frequencies, respectively. . . .	81
4.5	Frequency distribution of nouns (top 25 / 3,090) in E.T. Bench. The x- and y-axes denote the nouns and their frequencies, respectively. .	82
4.6	Overall architecture of E.T. Chat. We reformulate timestamp prediction as an embedding matching problem. Detailed are provided in Section 4.4.1.	83
4.7	Detailed illustration of the frame compressor. It accepts video patch embeddings P_t and the text prompt T as inputs, and compress video frame features into a single token.	84
4.8	Design space for timestamp processing. Existing MLLMs handle timestamps in videos or coordinates in images via (a) numerical expressions, (b) special tokens, or (c) external modules.	85
4.9	Left: Performance comparison between E.T. Chat and representative models. Right: Ranking of MLLMs on E.T. Bench, where red means higher ranks and blue represents lower ranks.	96
4.10	Qualitative comparison on [RAR] (left) and [ECA] (right).	104
4.11	Qualitative comparison on [RVQ] (left) and [TVG] (right).	104
4.12	Qualitative comparison on [EPM] (left) and [TAL] (right).	105
4.13	Qualitative comparison on [EVS] (left) and [VHD] (right).	105
4.14	Qualitative comparison on [DVC] (left) and [SLC] (right).	106
4.15	Qualitative comparison on [TEM] (left) and [GVQ] (right).	106

5.1	UniPixel flexibly supports a variety of fine-grained image and video understanding tasks, including referring/reasoning/interactive segmentation and referred video description & question answering. It can also handle a novel PixelQA task that jointly requires object-centric referring, segmentation, and QA in videos.	109
5.2	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.	111
5.3	The overall 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.	114
5.4	Illustration of joint positional & temporal encoding. The joint encoding approach supports both point (X_1Y_1T) and box ($X_1Y_1X_2Y_2T$) prompts.	115
5.5	Visualization of UniPixel-2B’s outputs on PixelQA task. In the left-most frame, star markers and boxes represent point and box prompts, respectively. The frames with highlighted borders are those where visual prompts are applied.	131
5.6	Visualization of the outputs from UniPixel on PixelQA.	134
5.7	Visualization of the outputs from UniPixel on ReVOS.	134
5.8	Visualization of the outputs from UniPixel on Ref-DAVIS17.	135
5.9	Visualization of the outputs from UniPixel on GroundMoRe.	135
5.10	Visualization of the outputs from UniPixel on Ref-SAV.	136

5.11 Visualization of the outputs from UniPixel on ReasonSeg.	136
6.1 Illustration of VideoMind’s Chain-of-LoRA reasoning mechanism.	
The problem is decomposed by the planner and distributed to	
grounder, verifier, and answerer to systematically localize, verify,	
and interpret the relevant video moments.	140
6.2 The overall workflow of VideoMind. Given a video and a query, it	
adaptively activates different roles (e.g., Planner → Grounder → Ver-	
ifier → Answerer in this case) and performs step-by-step reasoning	
by calling individual modules.	144
6.3 Planner coordinates all the other roles based on the video and query	
context, offering three reasoning plans and a query rephrasing mech-	
anism to address diverse demands.	145
6.4 Detailed architecture of the timestamp decoder. This module accepts	
hidden states of both the frame tokens and the <REG> token, and	
decodes them into the corresponding start and end timestamps. . . .	146
6.5 The grounder generates multiple candidate moments, which are then	
refined by the verifier via zooming-in to select the best moment. . .	150
6.6 Visualization of VideoMind’s reasoning process. Through chaining	
the grounder, verifier, and answerer, our model accurately localizes	
the critical moment and selects the correct answer, avoiding confu-	
sion from incorrect segments.	162

List of Tables

2.1	Comparison with representative methods on QVHighlights test split. MR and HD denote moment retrieval and highlight detection, respectively. w/ PT means pre-training with ASR captions.	22
2.2	Comparison with representative methods on Charades-STA test split. All the models use the officially released VGG and/or optical flow features of Charades.	23
2.3	Experimental results on YouTube Highlights (metric: mAP). Above are the methods using visual features only, the others are using visual-audio features.	23
2.4	Experimental results on TVSum (metric: Top-5 mAP). Above are the methods using visual features only, the others are using visual-audio features.	24
2.5	The effectiveness of multi-modal learning on YouTube Highlights, TVSum, and QVHighlights val split. MR and HD denote moment retrieval and highlight detection, respectively.	25
2.6	Comparison with Moment-DETR using different training task combinations on QVHighlights val split. MR and HD denote moment retrieval and highlight detection, respectively.	26

2.7 Effectiveness justification of the offset loss for moment retrieval on QVHighlights val split. Both models are trained using the co- optimization recipe.	27
2.8 The influence of weakly relevant (TVSum) or highly relevant (QVHigh- lights) text queries on highlight detection.	27
2.9 Comparison of cross-modal fusion methods on YouTube Highlights, TVSum, and QVHighlights val split.	28
2.10 Ablation study the on number of tokens in the bottleneck transformer on QVHighlights val split (metric: mAP).	28
3.1 Capability comparison among different image-to-video transfer learn- ing frameworks. Our proposed R ² -Tuning is the only one satisfying all the principles.	49
3.2 Hyperparameters used for different datasets. LR Drop means the epoch that drops learning rate by 1/10. * Video clips in YouTube Highlights have overlaps. We adopt the same setting as UMT.	51
3.3 Video moment retrieval (MR) and highlight detection (HD) results on QVHighlights test split. Note that EaTR and UnLoc only reported their results on val split.  and  are CLIP-B/32 and B/16,  denotes SlowFast R-50, and  is PANN for audio features extraction. * means estimated parameters. The best and second-best results are marked with bold and <u>underline</u> , respectively.	52

3.4	Video moment retrieval results on Ego4D-NLQ, Charades-STA, and TACoS datasets.  means using both CLIP-B/32 and SlowFast R-50 as feature extractors.  indicates using CLIP-B/32 only. The best and second-best results are marked with bold and <u>underline</u> , respectively. Our method shows significant advantages in high-quality retrievals (R@0.7) even without a video backbone.	53
3.5	Class-wise video highlight detection results (mAP) on YouTube Highlights dataset.  ,  , and  denote image, video, and audio backbones, respectively.	54
3.6	Class-wise video summarization results (Top-5 mAP) on TVSum dataset.  ,  , and  denote image, video, and audio backbones, respectively. Our method does not require any additional video/audio features.	55
3.7	Effectiveness justification of the combination of granularity calibration losses on QVHighlights val split.	57
3.8	MR mAP with different query lengths on QVHighlights val split.	57
3.9	Comparison among different PETL and METL methods on QVHighlights val split. #Params and Memory represent the number of learnable parameters and peak GPU memory (224×224 inputs with 32 batch size & FP16), respectively.	58
3.10	Performance comparison with different orders of attention layers. Self and Cross represent self-attention and cross-attention, respectively.	59
3.11	Effectiveness justification of the recurrent design (parameter sharing across layers) in R ² Block.	59

3.12 Performance comparison with different numbers of transformer layers for temporal refinement.	60
4.1 Comparison between E.T. Bench and existing Video-LLM benchmarks.	71
4.2 Pre-filtering criteria for E.T. Bench generation. <u>N/A</u> means no filtering.	75
4.3 Instruction templates for E.T. Bench. Green text indicates the candidate expressions for difference domains. Blue text means the placeholder to be filled according to each sample. <time> denotes the timestamp representation.	79
4.4 Task and sample distribution in E.T. Instruct 164K.	88
4.5 Instruction and response templates in E.T. Instruct 164K (Part I). Blue text means the placeholder to be filled according to each sample, and <time> denotes the timestamp representation.	89
4.6 Instruction and response templates in E.T. Instruct 164K (Part II). Blue text means the placeholder to be filled according to each sample, and <time> denotes the timestamp representation.	90
4.7 Hyperparameter settings.	92
4.8 Architectures of the models on E.T. Bench. Size means the LLM size.	93
4.9 Performance of representative MLLMs on E.T. Bench. We evaluate both open-source and proprietary models. The best and second-best results are marked with bold and underline, respectively.	95
4.10 Comparison on architectural designs.	97
4.11 Choices of frame compressor.	97

4.12 Choices of layers for matching. Layer_{vid} and Layer_{frm} denote the index of LLM layer utilized for matching. -1 means using the final-layer hidden states.	98
4.13 Comparison on learnable modules. ATTN and FFN represent the attention and feed-forward layers in Q-Former, respectively.	98
4.14 Effect of α for label smoothing.	99
4.15 Joint effect of model and instruction-tuning (IT) dataset.	100
4.16 Ablation study on instruction-tuning tasks.	100
4.17 Licenses of source datasets in E.T. Bench.	103
5.1 The distribution of training datasets for UniPixel. We use different background colors to denote <i>object referring</i> , <i>object segmentation</i> , <i>regional understanding</i> , <i>memory pre-filling</i> , and <i>general image & video understanding</i> data, respectively.	120
5.2 Comparison with state-of-the-art methods on ReVOS val split.	121
5.3 Comparison on referring video object segmentation (RVOS) and motion-grounded video reasoning datasets, including MeViS (val), Ref-YouTube-VOS (val), Ref-DAVIS17 (val), and GroundMoRe (test).	122
5.4 Experimental results on MeViS val ^u set. Post means applying post optimization after mask prediction.	123
5.5 Performance comparison on Ref-SAV val set. FT means fine-tuning after pre-/co-training.	123
5.6 Performance comparison with representative methods on referring expression segmentation (RES) and reasoning segmentation datasets, including RefCOCO/+g and ReasonSeg (val).	124

5.7	Fine-tuned performance on referring expression segmentation (RES) datasets, including RefCOCO/+/g.	125
5.8	Performance comparison on referring expression comprehension (REC) datasets, including RefCOCO/+/g.	125
5.9	Comparison with state-of-the-art methods on VideoRefer-Bench ^D .	126
5.10	Comparison with state-of-the-art methods on VideoRefer-Bench ^Q (<i>mask prompts</i>). MF denotes multi-frame mode.	127
5.11	Evaluation results on PixelQA. All the visual prompts are applied in a single frame.	127
5.12	Comparison with representative methods on MVBench. Note that UniPixel is the only model supporting pixel-level referring & segmen- tation.	128
5.13	Key ablation studies with UniPixel-2B on PixelQA (<i>mixed</i>).	128
5.14	Effectiveness justification of multi-stage training. The three-stage training recipe leads to the optimal performance.	129
5.15	Ablation study on the number of hidden tokens for each <SEG> token. Performance gains are negligible with more than 2 tokens/object.	129
5.16	Ablation study on M→L projector. Init and PT denote weight initial- ization from V→L projector and extra pre-training, respectively.	129
5.17	Ablation study on training data used in stage 3. Gradually adding more pixel-level data brings performance gains.	130
6.1	Training datasets for different roles. Source datasets were repur- posed for training planner and verifier. <i>mr</i> and <i>step</i> denote the mo- ment retrieval and step localization subsets, respectively.	149

6.2	Statistics of evaluation benchmarks. The datasets encompass three representative tasks – Grounded VideoQA, Video Temporal Grounding, and General VideoQA, with video durations ranging from several seconds to more than 1 hour.	152
6.3	Performance comparison on Grounded VideoQA on CG-Bench. Despite its smaller size, VideoMind surpasses GPT-4o and open-source baselines on this challenging long video benchmark (avg. duration: 27 min).	154
6.4	Performance comparison on Grounded VideoQA on ReXTime. FT indicates whether fine-tuned on the downstream training set.	155
6.5	Performance comparison on Grounded VideoQA on NExT-GQA.	156
6.6	Performance of video temporal grounding on Charades-STA.	157
6.7	Performance of video temporal grounding on ANet-Captions.	157
6.8	Video temporal grounding on TACoS. Note that our method was co-trained on this dataset.	157
6.9	Performance of video temporal grounding on Ego4D-NLQ.	157
6.10	Fine-tuned video temporal grounding results on QVHighlights.	158
6.11	Performance comparison on reasoning temporal localization on ActivityNet-RTL. Our zero-shot method outperforms the fine-tuned baseline LITA by a considerable margin.	159
6.12	VideoQA performance on Video-MME, MLVU, and IVBench.	159
6.13	Performance of VideoQA on LongVideoBench val split.	159
6.14	Performance comparison on general VideoQA on MVBench.	160
6.15	Comparison with different test-time scaling strategies.	160

6.16 Effects of individual roles. The top half evaluates the full system	
on grounded VideoQA, while the bottom half details the impact of	
the grounder and verifier on video temporal grounding. Ground%	
indicates the percentage of samples processed with the grounder.	. . 161
6.17 Effect of the temporal feature pyramid on Charades-STA.	 161
6.18 Effect of different verifier styles on Charades-STA.	 161

Chapter 1

Introduction

Our experience of the world is interactive – we perceive what we desire, reason on what we observe, and generate thoughts through an iterative cycle of perception and reasoning. Human cognition naturally involves a dynamic interplay between these processes, shaping and refining our understanding over time [96, 175, 73, 44, 5]. Take a crowd scene, for instance: we might immediately notice a familiar face, a vivid outfit, or an unusual movement that align with our current goals or expectations. These perceptions trigger interpretations and questions, which in turn affect where we direct our attention next. Therefore, cognition is not a passive reception of sensory inputs, but an active, task-driven process guided by memory, context, and desire. Psychologists suggest that this interactivity improves attentional efficiency [181], supports adaptive behaviors in dynamic environments [65], and facilitates meaningful perception guided by prior knowledge [45, 77].

In contrast, contemporary multi-modal AI systems predominantly operate under a *single-pass* processing paradigm, in which perception and reasoning are performed sequentially and passively, without the opportunity for iterative refinement. In this framework, a model typically comprises a backbone (e.g., ResNet [84] or ViT [56]) for feature extraction, a series of neck modules for further processing, and multiple

task-specific heads for output prediction. The model is trained end-to-end to map the input signals directly to output results. Such a straightforward paradigm has been proven effective for constrained, well-defined tasks such as object detection [215, 211, 244, 347], image/video super-resolution [55, 262, 26], and audio recognition [86, 74, 113]. However, when facing complex, open-ended scenarios, *e.g.*, open-ended question answering on high-resolution images or hours-long videos, these *content-as-input* approaches exhibit significant limitations in both efficiency [27, 343, 258, 275] and generalizability [158, 194, 161], which we attribute to the fundamental constraints of the single-pass pipeline in comprehensively encoding information-dense content for diverse downstream objectives.

In this thesis, we systematically analyze and address this problem by introducing a new paradigm called ***interactive multi-modal content understanding***. Inspired by human cognitive processes, we argue that multi-modal content (*e.g.*, images and videos) should be treated not as static inputs, but as *interactive environments*. Given a task instruction (*e.g.*, a question about the content), the AI agent actively explores this environment, engaging in an iterative loop of perception and reasoning until it finalizes its response. This *content-as-environment* paradigm offers several advantages over conventional solutions:

1. **Task Adaptivity:** The optimal representation of a signal is inherently task-dependent. For instance, evaluating artistic style requires holistic perception, while assessing drawing skill demands attention to fine-grained local details. Conditioning perception on the task at hand allows the model to adapt its strategy, focus, and reasoning effectively across diverse scenarios.
2. **Resource Efficiency:** Through appropriate planning before perception, the model can prioritize the most relevant aspects of the content, significantly reducing compute, memory, and the strain on model capacity without compromising performance. This selective sensing mechanism mirrors human efficiency in focusing on key information while ignoring irrelevant details.

3. **Reflection Capability:** Treating perception and reasoning as an interleaved, recursive process enables models to self-monitor and refine their understanding over time. During reasoning, the agent can revisit ambiguous or underexplored regions of the content, identify inconsistencies, and iteratively update its internal representation before producing the final response. This reflective process enhances robustness, interpretability, and generalization.

The proposed framework fundamentally redefines how AI systems conduct multi-modal content understanding – through task-guided planning, step-by-step exploration, and flexible reflection when solving different problems.

1.1 What Do We Mean by “Interactive”?

Before proceeding, we would first like to provide a precise definition of “interactive multi-modal content understanding”, shown as follows.

Interactive multi-modal content understanding is a task-driven process in which an AI system actively explores the content through iterative cycles of perception, reasoning, and reflection.

Based on this definition, we broadly refer “interactive” as the mechanism of adaptive multi-pass processing for understanding. This involves several essential capabilities that an agent should possess: **(1) Perception:** The ability to adaptively localize relevant regions or moments within multi-modal content, guided by the specific task context. **(2) Reasoning:** The capacity to solve open-ended tasks based on the selectively perceived information. **(3) Interaction:** The integration of perception and reasoning in a recursive and verifiable manner, enabling step-by-step completion of complex tasks through continuous feedback and refinement.

As illustrated by the roadmap in Figure [1.1](#), this thesis has made progresses on all

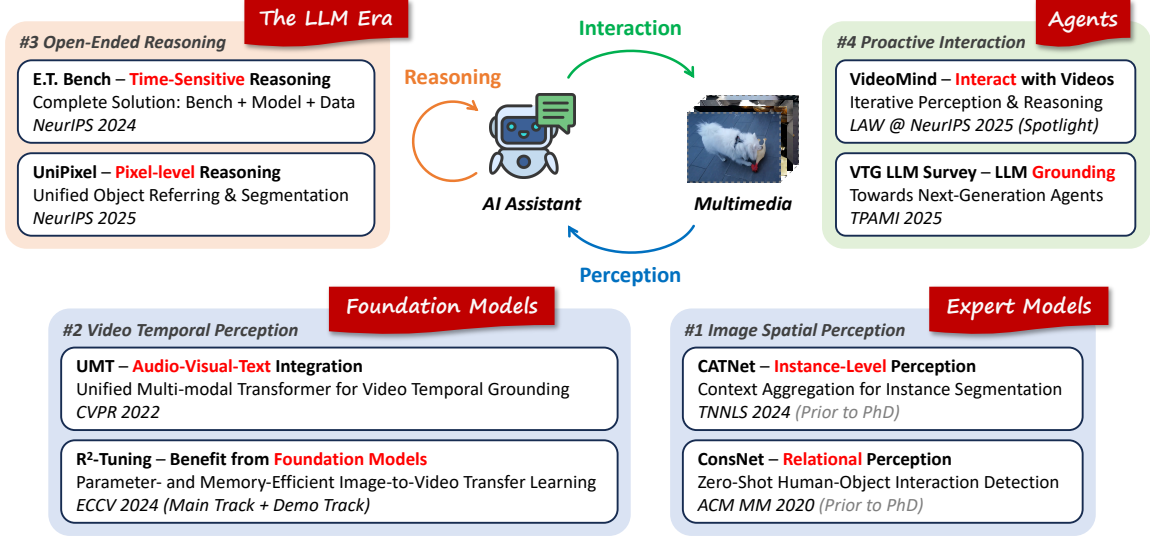


Figure 1.1: A roadmap towards interactive multi-modal content understanding. In this thesis, we comprehensively discuss and introduce a step-by-step path towards developing an interactive content understanding system.

three aspects toward building an interactive content understanding system. In particular, we study this problem through the lens of image and video understanding, addressing the technical challenges of spatial-temporal awareness, multi-modal integration, and long-form vision-centric reasoning.

1.2 Perception: Basic Information Acquisition

As previously discussed, perception serves as the initial stage of information acquisition directly from multi-modal content. In our proposed framework, this process goes beyond the passive reception of images or videos, and becomes a proactive mechanism involving key regions/moments localization and encoding. Our previous works [154, 159] have explored the topic of *image spatial perception*. Here, in **Part I**, we investigate *video temporal perception*, which aims to temporally localize relevant moments in videos based on flexible natural language queries.

In addition to the extra temporal dimension, videos also contain an audio modality, which leads to a natural research question: *How to integrate audio, visual, and textual information into a unified model for video temporal grounding?* In **Chapter 2**, we propose Unified Multi-modal Transformer (UMT) [155], a unified framework for video moment retrieval and highlight detection. The proposed model supports flexible modality combinations as inputs, including *video + audio*, *video + text*, and *video + audio + text*. This is realized by a novel modality fusion module called bottleneck transformer, which employs a small set of learnable tokens as the bridge for modality-specific information compression and expansion. This design allows joint augmentation of visual and audio features with linear computational complexity, while effectively filtering out noise from both modalities.

While UMT has demonstrated significance in audio-visual-text integration, it relies on two backbones for feature extraction and several sophisticated modules for downstream processing. With the emergence of vision foundation models, **Chapter 3** addresses a new question: *Is it possible to efficiently transfer a foundation model from image to video understanding?* Our preliminary experiments reveal that existing methods merely utilize the last-layer features of CLIP [205], which contains only global semantics and is sub-optimal for fine-grained query-event alignment. Therefore, we propose Reversed Recurrent Tuning (R²-Tuning) [153], a parameter- and memory-efficient transfer learning framework for video temporal grounding. Our method learns a lightweight R² Block comprising only 1.5% of the total parameters to perform progressive spatial-temporal modeling from coarse to fine. Strong performance has been achieved on several public benchmarks.

1.3 Reasoning: Open-Ended Task-Solving

Taking one step beyond perception, we also study how to integrate *open-ended reasoning* with fine-grained perception techniques. Thanks to the advancement of

large language models (LLMs) [2, 8, 50, 245, 1], it has become feasible to leverage multi-modal large language models (MLLMs) [188, 15, 350] as a foundation for open-ended image and/or video understanding. Specifically, we focus on equipping MLLMs with fine-grained *temporal* and *spatio-temporal* awareness.

First, we are interested in *How to enable time awareness in MLLMs for precise event referring and grounding?* A comprehensive discussion and investigation is provided in **Chapter 4**, where we introduce a complete solution for event-level and time-sensitive video understanding [158], consisting of a benchmark (E.T. Bench), a model (E.T. Chat), and an instruction-tuning dataset (E.T. Instruct 164K). Our benchmark results show that most existing MLLMs for holistic image/video understanding fail to comprehend time information in videos. We attribute it to the improper modeling of timestamps, and propose to reformulate timestamp prediction as an embedding matching problem. The effectiveness of our architectural and data innovation has been verified on our benchmark.

We then explore a more fine-grained scenario to enable *region-level spatial-temporal reasoning in videos*. In **Chapter 5**, we introduce UniPixel [157], a unified MLLM framework capable of flexibly interpreting visual prompt inputs and generating mask-grounded responses. The model accepts various visual prompts (*i.e.*, points, boxes, and masks) as spatial references, and integrates them seamlessly into open-ended video understanding. Its outputs may also include special tokens that can be decoded into masks. This flexibility empowers the model to focus on particular regions or moving objects, enabling more precise and object-centric reasoning.

1.4 Interaction: Proactive Sensing & Thinking

Upon resolving the critical challenges in task-driven perception and reasoning, we culminate our efforts in **Chapter 6** with the development of VideoMind [156], an

agentic framework for multi-modal content understanding. VideoMind integrates the previously introduced techniques in this thesis into a cohesive system comprising four carefully designed roles: Planner, Grounder, Verifier, and Answerer, each responsible for task planning, relevant moment grounding, verification of localization results, and moment-specific question answering, respectively.

To support flexible and efficient role-switching at inference time, we also introduce a novel Chain-of-LoRA mechanism, which loads multiple LoRA adapters [88] and activates them on demand. This modular role design empowers VideoMind to perform iterative perception and reasoning, significantly improving its performance on long video understanding tasks. Notably, our 2B model surpasses GPT-4o [188] on several long video understanding benchmarks [28, 343, 258].

Part I

Perception

There is nothing in the intellect that was not first in the senses.

— Thomas Aquinas, *Quaestiones Disputatae de Veritate*

Chapter 2

Unified Multi-modal Transformers for Video Content Analytics

Finding relevant moments and highlights in videos according to natural language queries is a natural and highly valuable common need in the current video content explosion era. Nevertheless, jointly conducting moment retrieval and highlight detection is an emerging research topic, even though its component problems and some related tasks have already been studied for a while. In this chapter, we present the first unified framework, named Unified Multi-modal Transformers (UMT), capable of realizing such joint optimization while can also be easily degenerated for solving individual problems. As far as we are aware, this is the first scheme to integrate multi-modal (visual-audio) learning for either joint optimization or the individual moment retrieval task, and tackles moment retrieval as a keypoint detection problem using a novel query generator and query decoder. Extensive comparisons with existing methods and ablation studies on QVHighlights, Charades-STA, YouTube Highlights, and TVSum datasets demonstrate the effectiveness, superiority, and flexibility of the proposed method under various settings.

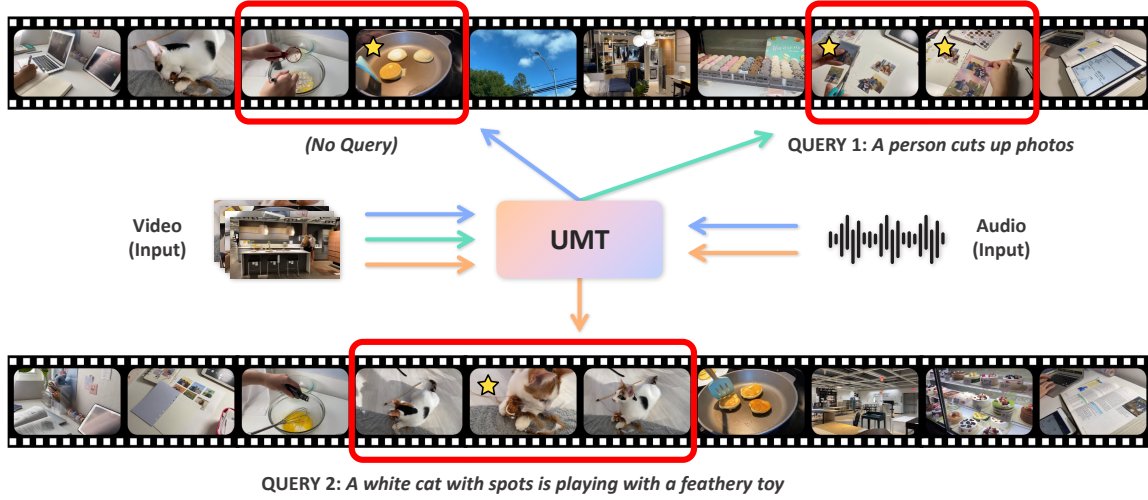


Figure 2.1: The proposed UMT is a unified and flexible framework which can handle different input modality combinations, and output video moment retrieval and/or highlight detection results (marked by red rectangles and golden stars, respectively). Note that different text queries lead to different outcomes from the same video. Arrows in different colors denote different input-output combinations.

2.1 Introduction

Video has already become the major media in content production, distribution, and consumption in our daily lives. It has the unique advantage of being able to include visual, audio, and linguistic information in the same media, in line with our natural experiences. Such an advantage on information richness, however, is also a challenging factor limiting its production and consumption, as it brings about very high costs on satisfying two critical needs. The first one is to find relevant moments in existing videos for producing new content or just getting creation hints from such references. The second one is to glance at large amounts of video content quickly by scanning video highlights rather than watching the entire original videos or video moments at a normal speed, which is needed by both video producers and consumers in such a content explosion era.

These two critical needs lead to two important research topics: video moment retrieval [6, 70] and video highlight detection [253, 234, 306], respectively. Although one may realize that these two tasks are closely related (especially when a text query is given), they have not yet been explicitly jointly studied until a very recent work [120] which builds a novel dataset called QVHighlights for this purpose and presents the first model Moment-DETR optimized for jointly solving both problems. Nevertheless, this seminal work has several limitations. It assumes a text query always exists and it has only applied to the visual modality of each video. Moreover, it is still a very basic model called a strong baseline, although it adopts a transformer framework, the latest and fast-rising neural network architecture type.

This chapter goes deeper into designing joint video moment retrieval and highlight detection approaches by mainly exploring two aspects: *multi-modal learning* and *flexibility*, as shown in Figure 2.1. Apart from text and video (*i.e.*, visual information), audio is also treated as an important input. Moreover, a unified yet flexible framework called Unified Multi-modal Transformers (UMT) is proposed to handle different modality reliability situations and combinations. For example, when the text input is unavailable, the task degenerates to highlight detection only. When there is some significant distraction in the text, its reliability will be compromised. Moreover, the audio may also be noisy, which may limit effective exploration. UMT covers all these natural variations which conventionally need to be resolved by different specifically designed models.

To demonstrate the effectiveness and superiority of the proposed framework, we conduct experiments not only on the QVHighlights dataset [120], the only one built for joint video moment retrieval and highlight detection, but also on popular public datasets for moment retrieval (Charades-STA [70]) and highlight detection (TVSum [230], YouTube Highlights [234]), with or without text guidance. Detailed ablation studies are also carried out to evaluate the essential components of the proposed scheme and to reveal meaningful insights.

2.2 Related Work

2.2.1 Video Moment Retrieval & Highlight Detection

Moment retrieval is a recently studied research topic that focuses on retrieving related moments in a video given a natural language query. Most existing works [6, 70] assume there is only a single moment in a video corresponding to a given text query, and such queries are usually about activities. The recently proposed QVHighlights dataset [120] goes beyond that by annotating multiple moments in a video for each query and breaks the former moment distribution bias. Video retrieval via text query [191] is a similar task, but it retrieves whole videos rather than video moments. Some works on language grounding [241, 212] align textual phrases to temporal video segments, which operate at a finer scale and target at different applications. Highlight detection concerns about detecting interesting or salient video segments in a video. It has a long history of about two decades with a rich literature, covering various domains of videos, including sports [253, 238], social media [234], and first-person [306]. QVHighlights is the only dataset supporting highlight detection conditioned on text-guided moment retrieval results. Video summarization is a closely related task that targets at summarizing a long video with short segments. It focuses on representativeness, diversity, and storyline, and thus it tends to be considered as a downstream application of highlight detection [306]. Dynamic video thumbnail generation [293] is another downstream task, which selects attractive video highlights and reforms them into a short segment to serve as the thumbnail. Among all these tasks, moment retrieval and highlight detection are two fundamental ones and can get highly correlated when a text query is given. This study follows QVHighlights on modeling both tasks together within a single framework. Different from Moment-DETR [120], our model has the flexibility to perform moment retrieval or highlight detection only.

2.2.2 Text Query Based Models

While text query is a must for moment retrieval, it seldom appears in the studies for video highlight detection, though we believe that providing text queries leads to better results as highlights are usually subjective and interest-dependent. An early work [115] proposes using text to find video highlights, but it is just about using a text ranking algorithm to rank video descriptions in the text domain for providing supervision to video shot ranking, not directly matching text and highlights. The only text-guided highlight detection exists in the very recent work [120]. In the closely related field of video thumbnail generation, text queries were first investigated in [317], where a graph convolutional network is used to model the word-by-clip interactions. Later on, sentence-guided temporal modulation mechanism [218] is proposed to modulate an encoder-decoder based network. All these works assume the reliability of the text query, while our proposed framework can easily work without text queries or with unreliable text queries.

2.2.3 Multi-modal Learning

Multi-modal learning approaches have been explored for highlight detection by jointly modeling visual and audio modalities. The earliest work seems to be MININET [87], which simply concatenates the feature vectors of both modalities. Very recently, two more sophisticated modality fusion models [307, 12] have been proposed and significantly boosted the performances. One of them invents a visual-audio tensor fusion mechanism [307] to learn cross-modality relationships with tensor decomposition and low-rank constraints. The other does the fusion via cross-modal bidirectional attention layers [12] which extract audio-attended visual features and visually-attended audio features. Though all the three approaches share the same idea with us on learning multi-modal fused representations for highlight detection, only the attention-based work build their model under the same

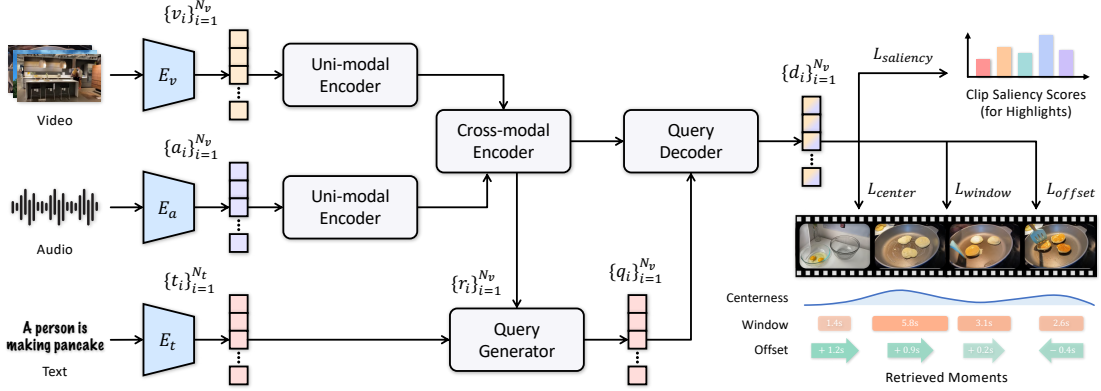


Figure 2.2: Overall architecture of our framework. When either the video or audio is unavailable, the corresponding uni-modal encoder and cross-modal encoder are deactivated. If text queries are not provided, the model would simply use learnable moment queries instead.

supervised-learning setting as ours. To the best of our knowledge, our approach is the first one to solve joint moment retrieval and highlight detection with multi-modal (visual-audio) learning.

2.3 Method

2.3.1 Overview

Given an untrimmed video V containing N_v clips and a natural language query T with N_t tokens, the goal of joint video moment retrieval and highlight detection is to localize all the moments (represented by temporal boundaries $b \in \mathbb{R}^2$) in V , in which the visual and/or audio contents are highly relevant to T , while predicting clip-level saliency scores $\{s_i\}_{i=1}^{N_v}$ for each moment simultaneously.

As shown in Figure 2.2, the overall architecture of our framework derives from the transformer encoder-decoder structure, and can be divided into five parts, *i.e.*,

uni-modal encoder, cross-modal encoder, query generator, query decoder, and prediction heads. The input video and text are firstly processed by pre-trained feature extractors. Specifically, we use three different models (E_v , E_a , and E_t) to extract visual, audio, and textual features, respectively. Each video-text pair can be therefore represented by three collections of feature vectors, namely visual features $\{v_i\}_{i=1}^{N_v}$, audio features $\{a_i\}_{i=1}^{N_v}$, and textual features $\{t_i\}_{i=1}^{N_t}$. The visual and audio features are fed into separate uni-modal encoders to be contextualized under global receptive field, then be fused by the cross-modal encoder for visual-audio joint representations $\{r_i\}_{i=1}^{N_v}$. These representations, together with textual features, are used to generate clip-level moment queries $\{q_i\}_{i=1}^{N_v}$ that can be utilized to retrieve moments and highlights from joint representations in the query decoder. After decoding query-guided video features $\{d_i\}_{i=1}^{N_v}$, we use two prediction heads to obtain the final moment retrieval and highlight detection results.

2.3.2 Uni-modal Encoder

Most existing feature extractors for videos and audios [247, 24, 113] are under the sliding window scheme, thus these methods only consider local temporal correlations, without being aware of the global context information, which is of great essence for video understanding tasks. Detecting queried moments and highlights in a video also requires an overall view of the global content. Therefore, to augment the features with global context within each modality, we adopt uni-modal encoder to process the input visual and audio features. This module is constructed by stacking standard transformer encoder layers [248], each consisting of a multi-head self-attention block and a feed-forward network. In each attention head, self-attention for either of the visual or audio modality $x \in \{v, a\}$ can be computed as

$$x'_i = x_i + w_z \sum_{j=1}^{N_v} \frac{\exp(w_q x_i \times w_k x_j)}{\sum_{m=1}^{N_v} \exp(w_q x_i \times w_k x_m)} w_v x_j \quad (2.1)$$

where x_i and x'_i are the input and output features of clip i , and $w_{\{q,k,v,z\}}$ indicates linear transform weights for the query, key, value, and output matrices. More details about self-attention are referred in [248]. The above formula computes the embedded gaussian correlations among clips, and aggregates the global context information into each clip. Subsequently, a two-layer feed-forward network formed by $\text{Linear} \rightarrow \text{ReLU} \rightarrow \text{Dropout} \rightarrow \text{Linear}$ is used to further project the features.

2.3.3 Cross-modal Encoder

Previous studies [12, 180] have claimed that jointly modeling multi-modal features can better obtain the overall representations. Hence after the uni-modal encoders, an extra cross-modal encoder is utilized to jointly capture the global correlations across modalities. Here, the exact form of the cross-modal encoder is not crucial. A straightforward approach is to apply cross-modal attention [12]. However, such a strategy has two weaknesses. First, as typical natural signals, both visual and audio features have heavy spatial-temporal redundancy and noisy information that are useless for other modalities. Second, the computation of cross-modal attention is costly, with square complexity when calculating clip-to-clip correlations. A recent work [180] tried to tackle the first problem by introducing attention bottlenecks that can be regarded as the information bridge across modalities. Although promising results have been achieved, this module still suffers from high computational cost since inter- and cross-modal correlations are jointly modeled. In this work, we extend this idea and propose to disentangle these strategies, thus the resulting bottleneck transformer module can be divided into two stages, *i.e.*, feature compression and expansion, shown in Figure 2.3.

Feature Compression Following [180], we introduce bottleneck tokens $\{z_i\}_{i=1}^{N_b}$ to capture the compressed features from all modalities. Here N_b is a number much smaller than the number of video clips N_v . The feature compression is realized

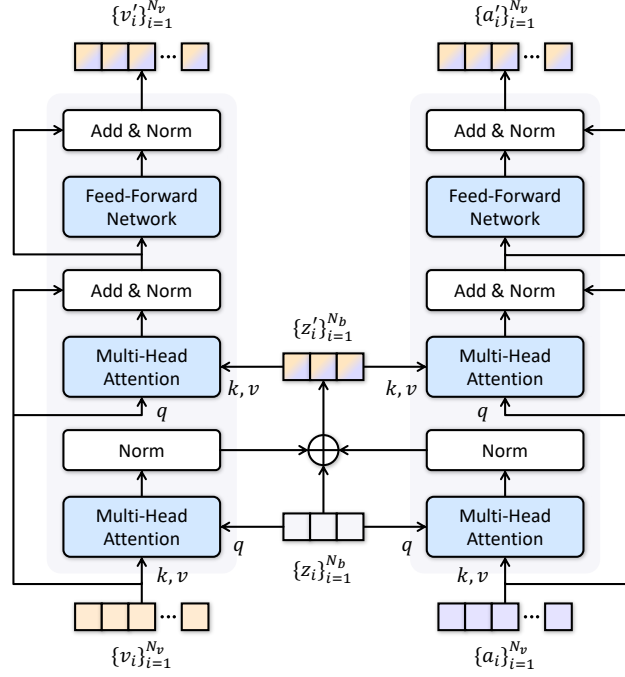


Figure 2.3: The architecture of bottleneck transformer module. We introduce bottleneck tokens for cross-modal feature compression and expansion, largely reducing the computational cost.

by several multi-head attentions between bottleneck tokens and the features from different modalities. Since there are only visual and audio modalities in this case, the compression process can be represented by

$$z'_i = z_i + w_z \sum_{j=1}^{N_v} \frac{\exp(w_q z_i \times w_k x_j)}{\sum_{m=1}^{N_v} \exp(w_q z_i \times w_k x_m)} w_v x_j \quad (2.2)$$

where z_i and z'_i are input and output features of bottleneck tokens. Other notations are consistent with Eq. 2.1. The only difference between Eq. 2.1 and Eq. 2.2 is that the query matrix is replaced by z_i , aiming to aggregate features into bottleneck tokens. We apply this operation for both visual and audio features, so that multi-modality information is refined and compressed into bottleneck tokens.

Feature Expansion After compressing the multi-modal information, we expand the features and propagate them into each modality using another multi-head at-

tention. Formally, the computation is as follows.

$$x'_i = x_i + w_z \sum_{j=1}^{N_v} \frac{\exp(w_q x_i \times w_k z_j)}{\sum_{m=1}^{N_v} \exp(w_q x_i \times w_k z_m)} w_v z_j \quad (2.3)$$

Here, x'_i represents the cross-modality enhanced features of clip i . These features are then fed into feed-forward networks for further projection. Leveraging such a two-stage feature propagation across modalities, visual and audio features are augmented under linear computational complexities, while the noises from both modalities are effectively discarded.

2.3.4 Query Generator

As transformers are firstly introduced for language translation tasks, the lengths of the input and output sequences may not be the same, where the length of the output sequence is determined by the query embeddings fed into the decoder. When generalized to vision tasks, query embeddings are randomly initialized and learned during training. Such a scheme may not be suitable for video highlight detection, since the outputs ought to be strictly aligned with the input tokens. Besides, query embeddings shall naturally guide the process of representation decoding. Therefore, we introduce a query generator to adaptively generate temporally aligned moment queries depending on the natural language input. This module is also constructed by a multi-head attention layer, in which visual-audio joint representations $\{r_i\}_{i=1}^{N_v}$ act as query, textual features are key and value. Our hypothesis is that by computing the attention weights between video clips and text queries, each clip can learn whether it contains which of the concepts described in the text, and predict a query embedding that can be used to decode the learned information for different needs. Note that when text queries are not available, joint representations and learnable positional encodings are summed up to serve as moment queries instead.

2.3.5 Query Decoder and Prediction Heads

Query decoder takes visual-audio joint representations $\{r_i\}_{i=1}^{N_v}$ and text-guided moment queries $\{q_i\}_{i=1}^{N_v}$ as inputs, and decodes the video features for joint moment retrieval and highlight detection. The output sequence of the query decoder has the same length as the encoder input. Such a peculiarity has two advantages: (1) We may obtain the clip-level saliency (highlight) scores as simple as adding a linear projection layer with sigmoid activation. 2) The dynamic length of output sequence also enables us to define moment retrieval as a *keypoint detection* problem [119, 347]. That is, each moment can be represented by its temporal center and duration (window), where the center point can be estimated by predicting a temporal heatmap and extracting local maxima. The window can be further regressed from features of the center. Note that the points in the heatmap are discrete, which may be misaligned with the real temporal center and inevitably damage the retrieval performances. An extra offset term used to adjust the center should also be predicted. We adopt four linear projection layers to predict the saliencies, centers, windows, and offsets, respectively.

During training, the clip-level saliency score prediction is optimized using a binary cross-entropy loss $\mathcal{L}_s$. For each ground truth moment with center $p \in [1, N_v]$ and window d , we quantize the center point to $\tilde{p}$ and fill the heatmap $H \in [0, 1]^{N_v}$ using a 1D gaussian kernel $H_x = \exp(-\frac{(x-\tilde{p})^2}{2\sigma_p^2})$ with radius r_p . Here, x indicates the temporal coordinate aligned with clip indices, r_p and σ_p are window-adaptive parameters that can be computed as

$$r_p = \mu \cdot d \quad (2.4)$$

$$\sigma_p = \rho \cdot (r_p + 1) \quad (2.5)$$

where μ and ρ are hyperparameters controlling the corresponding values. We add 1 to r_p in Eq. 2.5 to ensure that the output σ_p is not too small, preventing extremely large values in the heatmap. We observe that the performance is not sensitive to

these hyperparameters, thus both of them are set to 0.2 in all experiments. We optimize the center point prediction using the gaussian focal loss [143] as

$$\mathcal{L}_c = -\frac{1}{N} \sum_x \begin{cases} (1 - \hat{H}_x)^\alpha \log(\hat{H}_x) & \text{if } H_x = 1 \\ (1 - H_x)^\gamma \hat{H}_x^\alpha \log(1 - \hat{H}_x) & \text{otherwise} \end{cases} \quad (2.6)$$

Here, N is the number of moments, α and γ denote the weighting and the exponent of the modulating factors in the focal loss, which are set to 2.0 and 4.0 in practice. For window and offset regression, we simply adopt L1 losses to optimize the actual values for all ground truth centers as

$$\mathcal{L}_w = -\frac{1}{N} \sum_p |w_p - \hat{w}_p| \quad (2.7)$$

$$\mathcal{L}_o = -\frac{1}{N} \sum_p |(o_p - \tilde{p}) - \hat{o}_p| \quad (2.8)$$

where w_p , $\hat{w}_p$, o_p , and $\hat{o}_p$ are the ground truth and predicted windows and offsets respectively. The overall training loss would be the weighted sum of all the losses above as

$$\mathcal{L} = \lambda_s \mathcal{L}_s + \lambda_c \mathcal{L}_c + \lambda_w \mathcal{L}_w + \lambda_o \mathcal{L}_o \quad (2.9)$$

where $\lambda_{\{s,c,w,o\}}$ are the weights for saliency, center, window, and offset losses, respectively. When testing, the moment boundaries are obtained by combining the center, window, and offset terms as introduced in [119, 347].

2.4 Experiments

2.4.1 Datasets and Experimental Settings

Datasets QVHighlights [120] is the only existing public dataset that has ground-truth annotations for both moment retrieval and highlight detection, thus being

suitable for evaluating the full version of our proposed model. This dataset contains videos cropped into 10,148 short (150s-long) segments, and had each segment annotated with at least one text query depicting its relevant moments. There are averagely about 1.8 disjoint moments per query, annotated on non-overlapping 2s-long clips. In total, there are 10,310 queries with 18,367 annotated moments. We follow the original QVHighlights data splits in all experiments.

We also utilize three more datasets: Charades-STA [70], YouTube Highlights [234], and TVSum [230] for further evaluation on the moment retrieval or highlight detection task only, as our model has the flexibility for tasks. Charades-STA contains 16,128 query-moment pairs annotating different actions. YouTube Highlights has 6 domains with 433 videos currently available. TVSum includes 10 domains with 5 videos each. We follow the tradition to work on a random 0.8/0.2 training/test split. Note that the annotators for TVSum were aware of the video titles, so we believe that these titles can serve as noisy text queries. Our model’s flexibility to handle this situation is studied.

Evaluation Metrics We use the same evaluation metrics used in existing works. Specifically, for QVHighlights, Recall@1 with IoU thresholds 0.5 and 0.7, mean average precision (mAP) with IoU thresholds 0.5 and 0.75, and the average mAP over a series of IoU thresholds [0.5:0.05:0.95] are used for moment retrieval. For highlight detection, mAP and HIT@1 are utilized, where a clip prediction is treated as a true positive if it has the saliency score of *Very Good*. For Charades-STA, Recall@1 and Recall@5 with IoU thresholds 0.5 and 0.7 are used. For YouTube Highlights and TVSum, mAP and Top-5 mAP are adopted, respectively.

Implementation Details On QVHighlights, we simply leverage the pre-extracted features using SlowFast [63] and CLIP [205]. Official VGG [227] and optical flow features, as well as GloVe [197] embeddings, are used for Charades-STA. On YouTube Highlights and TVSum, we obtain clip-level visual features using an I3D

Table 2.1: Comparison with representative methods on QVHighlights test split. MR and HD denote moment retrieval and highlight detection, respectively. w/ PT means pre-training with ASR captions.

Method	MR					HD	
	R1		mAP			$\geq$ Very Good	
	@0.5	@0.7	@0.5	@0.75	Avg.	mAP	HIT@1
BeautyThumb [229]	–	–	–	–	–	14.36	20.88
DVSE [152]	–	–	–	–	–	18.75	21.79
MCN [6]	11.41	2.72	24.94	8.22	10.67	–	–
CAL [59]	25.49	11.54	23.40	7.65	9.89	–	–
XML [121]	41.83	30.35	44.63	31.73	32.14	34.49	55.25
XML+ [120]	46.69	33.46	47.89	34.67	34.90	35.38	55.06
Moment-DETR [120]	52.89	33.02	54.82	29.40	30.73	35.69	55.60
Moment-DETR w/ PT	59.78	40.33	60.51	35.36	36.14	37.43	60.17
UMT (Ours)	56.23	41.18	53.83	37.01	36.12	38.18	59.99
UMT (Ours) w/ PT	60.83	43.26	57.33	39.12	38.08	39.12	62.39

[24] pre-trained on Kinetics 400 [24]. Since each feature vector captures 32 consecutive frames, we follow [12] and consider the feature vector belonging to a clip if their overlap is more than 50%. We also use CLIP to extract the title features in TVSum. Audio features are extracted by a PANN [113] model pre-trained on AudioSet [72]. Visual and audio features are temporally aligned at clip level.

All the models in our experiments contain one uni-modal and cross-modal encoder layer each. The number of decoder layers is set to 3 for QVHighlights and CharadesSTA, and 1 for YouTube Highlights and TVSum since they have smaller scales. The number of bottleneck tokens N_b is insensitive thus be set to 4. The weights of losses are set as $\lambda_s = 3.0$, $\lambda_c = 1.0$, $\lambda_w = 0.1$, and $\lambda_o = 1.0$, while λ_w and λ_o are reduced to 0.05 and 0.5 specially for CharadesSTA. Following [120], we set the hidden dimensions to 256, with $4\times$ dimension expansions in feed forward networks. Learnable positional encodings, pre-norm style layer normalizations [11], 8 attention heads, and 0.1 dropout rates are used in all transformer layers. We also adopt extra pre-dropouts with rate 0.5 for visual and audio inputs, and 0.3 for text inputs. In

Table 2.2: Comparison with representative methods on Charades-STA test split. All the models use the officially released VGG and/or optical flow features of Charades.

Method	R@1		R@5	
	IoU=0.5	IoU=0.7	IoU=0.5	IoU=0.7
SAP [34]	27.42	13.36	66.37	38.15
SM-RL [259]	24.36	11.17	61.25	32.08
MAN [326]	41.24	20.54	83.21	51.85
2D-TAN [334]	40.94	22.85	83.84	50.35
FVMR [71]	42.36	24.14	83.97	50.15
UMT [†] (Ours)	48.31	29.25	88.79	56.08
UMT [‡] (Ours)	49.35	26.16	89.41	54.95

[†] video + audio, [‡] video + optical flow

Table 2.3: Experimental results on YouTube Highlights (metric: mAP). Above are the methods using visual features only, the others are using visual-audio features.

Method	Dog	Gym.	Par.	Ska.	Ski.	Sur.	Avg.
RRAE [301]	49.0	35.0	50.0	25.0	22.0	49.0	38.3
GIFs [82]	30.8	33.5	54.0	55.4	32.8	54.1	46.4
LSVM [234]	60.0	41.0	61.0	62.0	36.0	61.0	53.6
LIM-S [287]	57.9	41.7	67.0	57.8	48.6	65.1	56.4
SL-Module [292]	70.8	53.2	77.2	72.5	66.1	76.2	69.3
MINI-Net [87]	58.2	61.7	70.2	72.2	58.7	65.1	64.4
TCG [307]	55.4	62.7	70.9	69.1	60.1	59.8	63.0
Joint-VA [12]	64.5	71.9	80.8	62.0	73.2	78.3	71.8
UMT (Ours)	65.9	75.2	81.6	71.8	72.3	82.7	74.9

all experiments, we use Adam [110] optimizer with 1e-3 learning rate and 1e-4 weight decay. The model is trained with batch size 32 for 200 epochs on QVHighlights, batch size 8 for 100 epochs on Charades-STA, batch size 4 for 100 epochs on YouTube Highlights, and batch size 1 for 500 epochs on TVSum, respectively.

Following [120], we also consider the weakly-supervised pre-training on QVHighlights with automatic speech recognition (ASR) captions. During pre-training, the saliency loss is turned off since the supervision is only for moment retrieval. We observe that UMT converges much faster than Moment-DETR due to the novel formulation of moment retrieval, thus we increase the batch size to 2048 and pre-train

Table 2.4: Experimental results on TVSum (metric: Top-5 mAP). Above are the methods using visual features only, the others are using visual-audio features.

Method	VT	VU	GA	MS	PK	PR	FM	BK	BT	DS	Avg.
sLSTM [331]	41.1	46.2	46.3	47.7	44.8	46.1	45.2	40.6	47.1	45.5	45.1
SG [172]	42.3	47.2	47.5	48.9	45.6	47.3	46.4	41.7	48.3	46.6	46.2
LIM-S [287]	55.9	42.9	61.2	54.0	60.4	47.5	43.2	66.3	69.1	62.6	56.3
Trailer [254]	61.3	54.6	65.7	60.8	59.1	70.1	58.2	64.7	65.6	68.1	62.8
SL-Module [292]	86.5	68.7	74.9	86.2	79.0	63.2	58.9	72.6	78.9	64.0	73.3
MINI-Net [87]	80.6	68.3	78.2	81.8	78.1	65.8	57.8	75.0	80.2	65.5	73.2
TCG [307]	85.0	71.4	81.9	78.6	80.2	75.5	71.6	77.3	78.6	68.1	76.8
Joint-VA [12]	83.7	57.3	78.5	86.1	80.1	69.2	70.0	73.0	97.4	67.5	76.3
UMT (Ours)	87.5	81.5	88.2	78.8	81.4	87.0	76.0	86.9	84.4	79.6	83.1

the model for 100 epochs to prevent overfitting.

2.4.2 Results on Joint Moment Retrieval & Highlight Detection

We first evaluate our proposed UMT on QVHighlights test split. The results are shown in Table 2.1. On both moment retrieval and highlight detection tasks, our proposed model outperforms all the existing approaches, including the previous state-of-the-art method Moment-DETR [120] under both settings (with or without pre-training with automatic speech recognition captions).

2.4.3 Results on Video Moment Retrieval

Table 2.2 shows the comparison of UMT with some representative methods on Charades-STA test split. Our approach performs better than previous methods under different metrics. We also tried adopting optical flow instead of audio and obtained similar performances, demonstrating the generalizability of the proposed bottleneck transformer when fusing different modality combinations.

Table 2.5: The effectiveness of multi-modal learning on YouTube Highlights, TVSum, and QVHighlights val split. MR and HD denote moment retrieval and highlight detection, respectively.

Method	YouTube Highlights	TVSum	QVHighlights	
	mAP	Top-5 mAP	MR (mAP)	HD (mAP)
MINI-Net [†] [87]	61.38	69.79	–	–
Joint-VA [†] [12]	70.50	74.80	–	–
Moment-DETR [†] [120]	–	–	32.20	36.52
UMT[†] (Ours)	73.48	81.89	37.79	38.97
MINI-Net [‡] [87]	52.23	59.72	–	–
Joint-VA [‡] [12]	67.00	68.70	–	–
Moment-DETR [‡] [120]	–	–	16.69	26.00
UMT[‡] (Ours)	65.61	76.51	13.73	23.91
MINI-Net [87]	64.36	73.24	–	–
Joint-VA [12]	71.80	76.30	–	–
Moment-DETR ⁺ [120]	–	–	34.05	37.67
UMT (Ours)	74.93	83.14	38.59	39.85

[†] video only, [‡] audio only, ⁺ w/ bottleneck transformer

2.4.4 Results on Video Highlight Detection

Highlight detection results on YouTube Highlights and TVSum are presented in Table 2.3 and Table 2.4, respectively. On both datasets, UMT performs better than not only representative methods which only use video, but also existing multi-modal ones that utilize both video and audio.

2.4.5 Ablation Studies

Multi-modality (Visual-Audio) Table 2.5 shows the performances of all the multi-modal methods when different modalities are used. Note that Moment-DETR⁺ is a multi-modal extension of original Moment-DETR [120] by implementing a similar bottleneck structure as UMT. Clearly, multi-modal learning can significantly boost most methods’ performance on all the datasets and tasks in comparison with using

Table 2.6: Comparison with Moment-DETR using different training task combinations on QVHighlights val split. MR and HD denote moment retrieval and highlight detection, respectively.

Method	Training		MR			HD	
	MR	HD	R1 @0.5	R1 @0.7	mAP Avg.	mAP	HIT@1
Moment-DETR [120]	✓		44.84	25.87	25.05	–	–
		✓	–	–	–	36.52	56.45
	✓	✓	53.94	34.84	32.20	35.65	55.55
UMT (Ours)	✓		54.14	33.82	34.02	–	–
		✓	–	–	–	40.22	65.03
	✓	✓	60.26	44.26	38.59	39.85	64.19

a single modality, since it can capture more useful information. Compared with the most similar competitor Moment-DETR⁺, UMT can better explore the complementary information from different modalities and suppress the possible noise.

Multi-task Co-optimization Given a text query for a video, retrieving the related moments and detecting salient highlights in such moments seem to be highly correlated tasks. Therefore, it is interesting to see how the multi-task co-optimization performs in comparison with training for each individual task when the same framework and backbone are used. We conduct single-task experiments by turning off the losses corresponding to each task and training the rest of the model. Note that moment retrieval is considered to be a harder task than highlight detection as explained in Section 2.1. The results in Table 2.6 show that the co-optimization not only generates results for both tasks simultaneously, but also significantly boosts the performance on moment retrieval. This is clear for both our UMT and Moment-DETR [120]. Note that when training for moment retrieval only, UMT performs much better than Moment-DETR, indicating its superiority on the model design. Our UMT better models the moment retrieval task as a keypoint detection problem [119, 347] rather than set prediction or clip classification. Moreover, the in-

Table 2.7: Effectiveness justification of the offset loss for moment retrieval on QVHighlights val split. Both models are trained using the co-optimization recipe.

Losses	R1		mAP		
	@0.5	@0.7	@0.5	@0.75	Avg.
$\mathcal{L}_c + \mathcal{L}_w$	62.32	43.23	57.78	38.61	37.36
$\mathcal{L}_c + \mathcal{L}_w + \mathcal{L}_o$	60.26	44.26	56.70	39.90	38.59

Table 2.8: The influence of weakly relevant (TVSum) or highly relevant (QVHighlights) text queries on highlight detection.

Text Query	TVSum	QVHighlights	
		mAP	HIT@1
	81.42	25.14	33.42
✓	83.14	39.85	64.19

puts to the UMT decoder are clip-aligned text-guided queries instead of positional encodings, which enables more flexible output sequence lengths and may provide stronger query information for each clip. Therefore, we believe that UMT can model the relationship between the two tasks better than Moment-DETR does.

Moment Retrieval Losses Table 2.7 presents the performances of UMT when different combinations of moment retrieval losses are used. Since the center loss $\mathcal{L}_c$ and the window loss $\mathcal{L}_w$ are mandatory for representing a moment, only the necessity of the offset loss $\mathcal{L}_o$ is justified. As the comparison shows, modeling the temporal offset does make the moment boundary prediction more accurate.

Justification of Text Queries We believe that highlight detection based on text queries is an important setting for highlight detection, as different interests can favor quite different highlights from the same video. Table 2.8 reports the results of our model, with or without the text queries. It can be seen that when the query is relevant, it does improve the highlight detection performance, and such an improvement is more significant when the relevance is greater.

Table 2.9: Comparison of cross-modal fusion methods on YouTube Highlights, TVSum, and QVHighlights val split.

Method	YouTube Highlights	TVSum	QVHighlights	
	mAP	Top-5 mAP	MR (mAP)	HD (mAP)
Concat	73.32	80.26	37.03	38.74
Mean	73.29	81.76	37.04	38.91
Sum	73.53	81.77	37.33	38.88
Bottleneck	74.93	83.14	38.59	39.85

Table 2.10: Ablation study the on number of tokens in the bottleneck transformer on QVHighlights val split (metric: mAP).

#Tokens	4	8	16	32	64	128
MR	38.59	37.19	38.29	37.15	37.20	38.18
HD	39.85	39.64	39.26	39.37	39.22	39.26

Cross-modal Fusion Methods Table 2.9 shows the comparison among bottleneck tokens and its baselines for cross-modal feature fusion on multiple datasets. Compared with simple fusion operations such as concatenation, mean, and sum, utilizing our the bottleneck transformer can improve the performance on both moment retrieval and highlight detection tasks.

Number of Bottleneck Tokens Table 2.10 examines the influence of the number of learnable bottleneck tokens. As the number increases from 4 to 128, the performance on moment retrieval and highlight detection remains relatively stable. Notably, using only 4 tokens yields the best performance. This is because the compression and expansion of features help eliminate undesirable noise.

2.4.6 Visualization

Figure 2.4 and Figure 2.5 presents some qualitative results on QVHighlights [120]. The results show that visual and audio features contribute to different moment re-

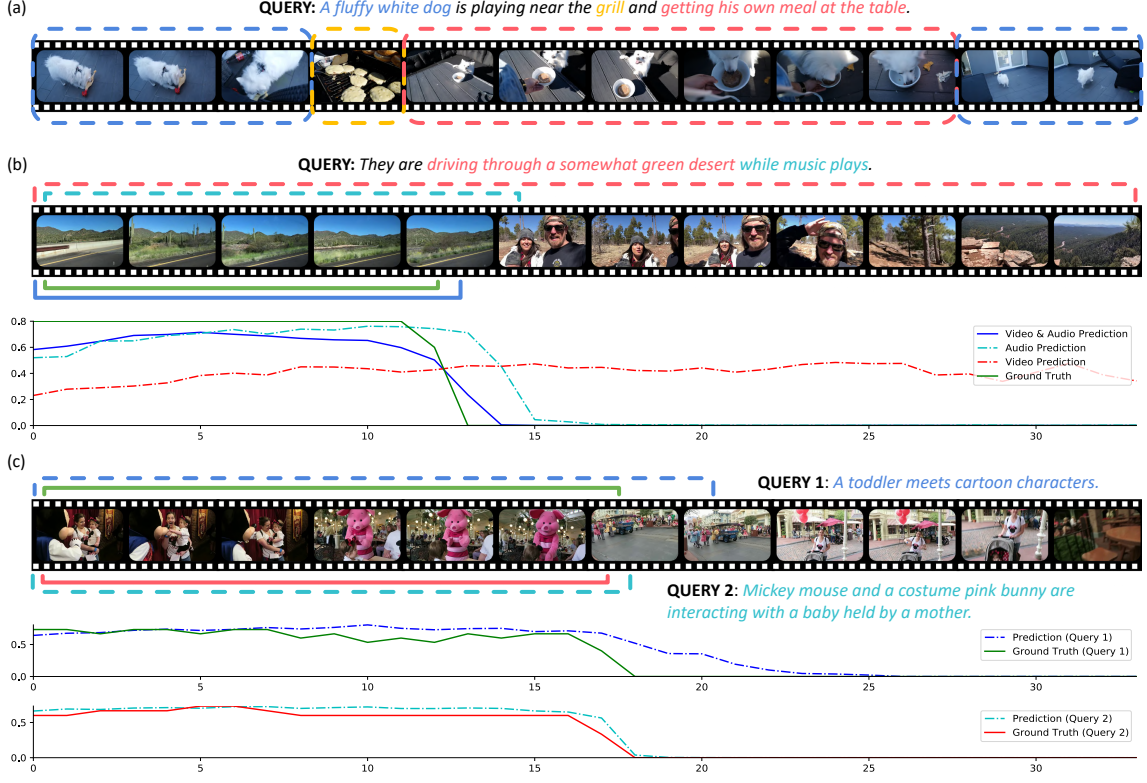


Figure 2.4: Visualizations on QVHighlights. The predicted moments and saliency scores are shown by brackets and lines. (a) All the highlight clips are presented, indicating that UMT can learn implicit correlations between video content and query semantics. (b) Different modality combinations guide the model pay attention to different moments. (c) Our model can handle multiple queries in a single video.

trieval and highlight detection outcomes. For example, in Figure 2.5 (b), the video-only model fails to refine the retrieved moment given the determiner “indicating we are listening to his audio”, while the audio-only model can not distinguish the moment boundaries. Combining the visual and audio information can effectively improve the performances on both tasks.

Figure 2.6 presents some failure cases on QVHighlights [120]. Figure 2.6 (a) shows that our model fails to comprehend the time adverbial clause “after a tiring trip”. Instead, it pays more attention to “a young mother and her family” and predicts

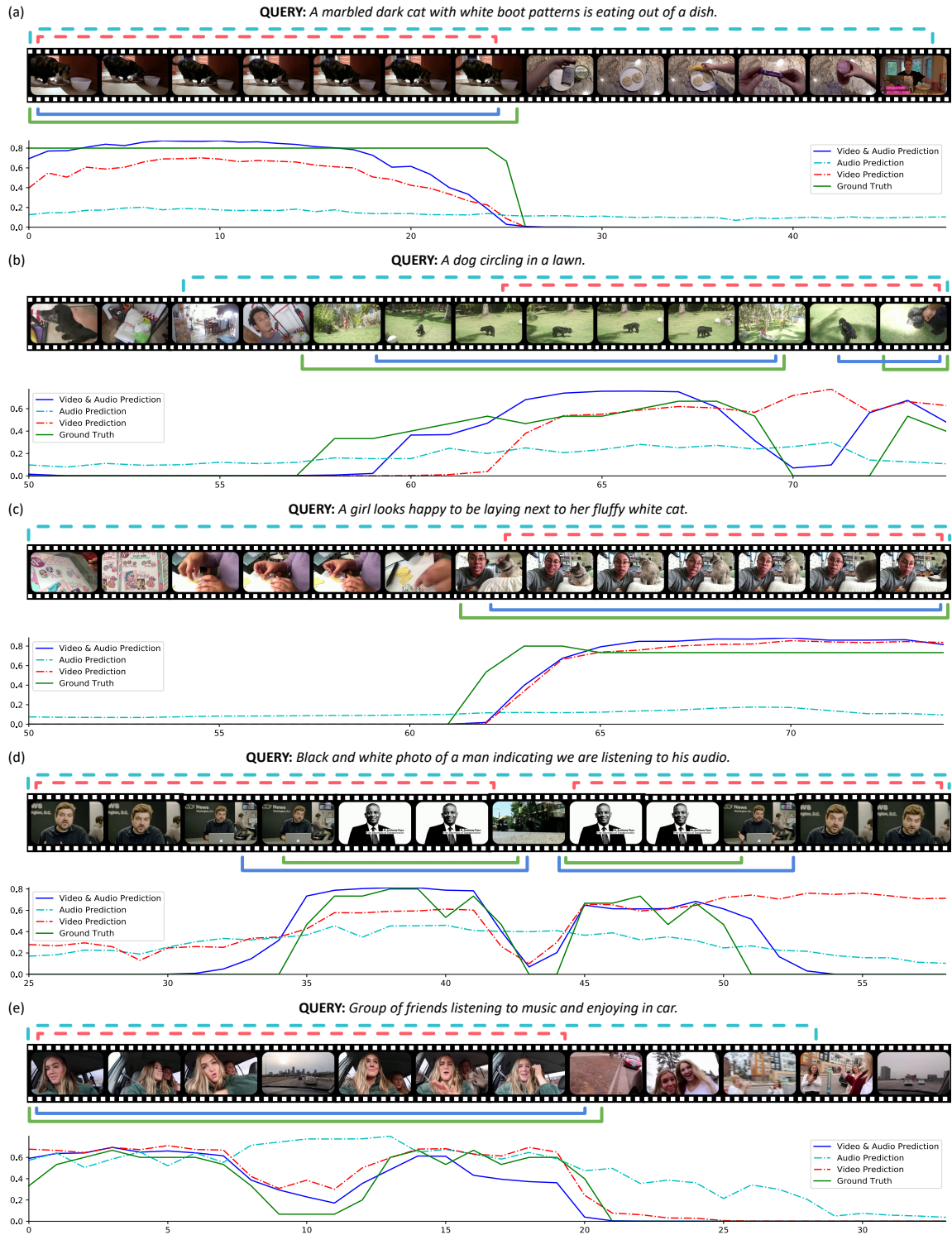


Figure 2.5: More visualization results on QVHighlights val split.

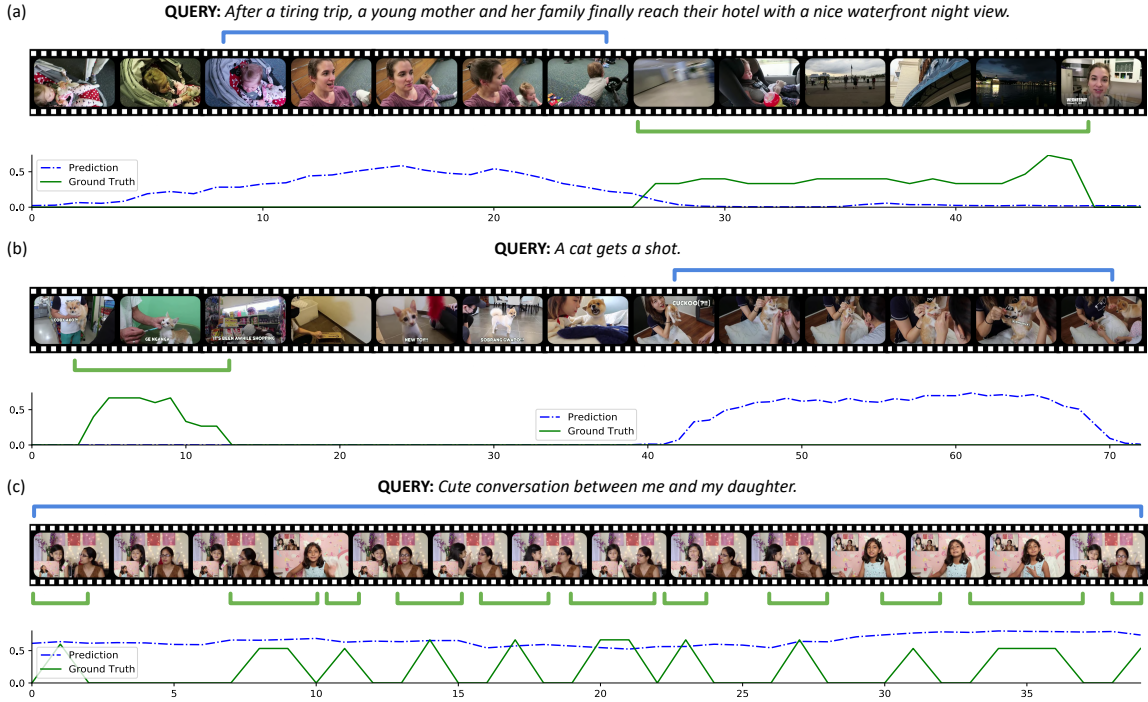


Figure 2.6: Failure cases on QVHighlights val split.

irrelevant moments. In Figure 2.6 (b), the visual appearances of the retrieved moment and the ground truth are similar. There are few visual clues that can be used to separate “shot” and other actions. Figure 2.6 (c) indicates that our model understands nouns, but cannot comprehend abstract words well. We argue that most failure cases are caused by the incomplete understanding of text queries, which may be remitted by using a stronger language model.

2.5 Summary

In this chapter, we explore the integration of vision, audio, and text modalities for video temporal understanding, which forms the foundation of our interactive content understanding framework. This framework is designed to enable robust perception under varying levels of modality reliabilities. Our contributions in this chapter are summarized as follows:

1. The first unified multi-modal transformer-based framework (UMT) for joint video moment retrieval and highlight detection, along with their individual component tasks, under different modality reliability conditions.
2. A cross-modality fusion module (Bottleneck Transformer) for facilitating effective and efficient audio-visual integration. This module operates with linear complexity and substantially mitigates noise propagation.
3. A task-guided multi-modal query generation scheme for better task unification and event-text alignment. Through this design, the encoder-decoder architecture can be utilized for both moment retrieval and highlight detection.

These contributions are based on the following technical novelties:

1. The token-level feature compression-expansion scheme in the bottleneck transformer fundamentally alleviates the high computational complexity issue, providing an efficient alternative to traditional cross-attention.
2. Our embedding-as-query approach unifies moment retrieval and highlight detection within a single encoder-decoder architecture, offering a more natural formulation compared with conventional DETR-based models.
3. Our model is grounded in a novel formulation that represents moments as keypoints for precise boundary prediction. This approach simplifies the design of the boundary regression head while achieving stronger performance.

2.6 Limitations & Future Work

Although task and modality unifications have been achieved, the proposed framework remains limited in its feature extraction process. Due to the absence of task-specific backbone architecture design, our model relies on two frozen backbones (*i.e.*, CLIP [205] and SlowFast [63]) for fine-grained video feature extraction. This is both sub-optimal and redundant. In the next chapter, we will propose a simple yet

effective solution to eliminate the dependency on the auxiliary SlowFast encoder, by effectively and efficiently transferring image-based CLIP encoders to video temporal grounding scenarios.

Chapter 3

Efficient Image-to-Video Transfer Learning for Temporal Grounding

Video temporal grounding (VTG) is a fine-grained video understanding problem that aims to ground relevant clips in untrimmed videos given natural language queries. Most existing VTG models are built upon frame-wise final-layer CLIP features, aided by additional temporal backbones (*e.g.*, SlowFast) with sophisticated temporal reasoning mechanisms. In this chapter, we claim that CLIP itself already shows great potential for fine-grained spatial-temporal modeling, as each layer offers distinct yet useful information under different granularity levels. Motivated by this, we propose Reversed Recurrent Tuning (R^2 -Tuning), a parameter- and memory-efficient transfer learning framework for video temporal grounding. Our method learns a lightweight R^2 Block containing only 1.5% of the total parameters to perform progressive spatial-temporal modeling. Starting from the last layer of CLIP, R^2 Block recurrently aggregates spatial features from earlier layers, then refines temporal correlation conditioning on the given query, resulting in a coarse-to-fine scheme. R^2 -Tuning achieves state-of-the-art performance across three VTG tasks (*i.e.*, moment retrieval, highlight detection, and video summarization) on six

public benchmarks (*i.e.*, QVHighlights, Charades-STA, Ego4D-NLQ, TACoS, YouTube Highlights, and TVSum) even without the additional backbone, demonstrating the significance and effectiveness of the proposed scheme.

3.1 Introduction

Video is becoming the major content media in our daily lives. The variety of video domains has extended beyond daily activities [24, 120] but broader scenes such as egocentric [76] and surveillance [316]. They maintain large information capacities within multi-granularities, and thus can convey both high-level context and low-level details effectively.

In the content production and consumption industry, such a flourishing is accompanied by the increasing demand for browsing untrimmed videos efficiently with different user interests. This derives the topic of video temporal grounding (VTG) [140], which is a fine-grained video-language understanding problem aiming to ground video clips conditioning on natural language queries. As shown in Figure 3.1, VTG can be disentangled into three tasks based on different output formats, *i.e.*, moment retrieval (MR) [70, 120, 155, 76] for regressing start-end timestamps, highlight detection (HD) [234, 120] for predicting frame-level saliency curves, and video summarization (VS) [81, 230, 9] for classifying disjoint shots.

Encouraged by the recent success in adopting vision-language models (VLMs) for video understanding [145, 183, 106, 165, 17, 92], most existing VTG methods [120, 155, 294, 178, 98, 177, 233] are built upon frame-wise final-layer features from CLIP [205]. However, due to CLIP’s misaligned pre-training objective (image-text contrast), these methods fail to capture temporal correlations well. As a feasible compromise, an additional backbone (*e.g.*, SlowFast [63]) is incorporated to complement the temporal information, followed by carefully designed modules such as text-guided queries [155], dynamic anchors [178], and event reasoning [98].

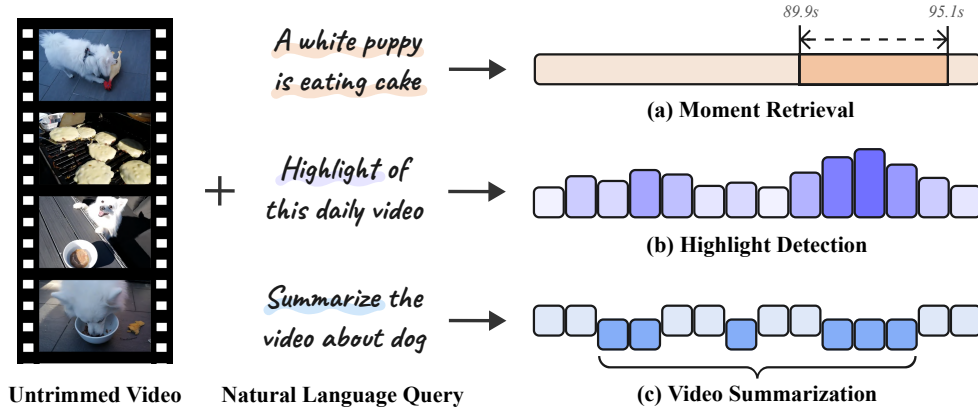


Figure 3.1: Video temporal grounding (VTG) contains three video-language understanding problems, *i.e.*, moment retrieval (MR), highlight detection (HD), and video summarization (VS).

We refer to the paradigm above as a *post-processing* scheme (shown in Figure 3.2 (a)), whereas two natural drawbacks exist due to the sub-optimal design. First, leveraging two backbones with similar capabilities is unintuitive and inefficient during inference. A single model with both vision-text alignment and spatial-temporal modeling abilities is more preferred. Second, queries for VTG could be of different granularities from coarse (*e.g.*, *the family is traveling*) to fine (*e.g.*, *when did I take the golf club from the man with white hair*). Leveraging only frame-wise & final-layer features is not granularity flexible, as it would focus more on high-level scene transitions while overlooking low-level details. Preliminary experiments in Figure 3.3 also demonstrate that the potential of spatial-temporal modeling for CLIP is not fully exploited by existing methods. Some recent works [298, 106, 92] tried to tackle the first problem by fine-tuning (part of) the CLIP encoders (Figure 3.2 (b) and (c)), but they are all inefficient in terms of data, learnable parameters, or memory.

This chapter intends to answer the research question: *how to efficiently transfer an image-language foundation model to video temporal grounding?* We consider the above question by exploring two aspects: *efficiency* and *granularity flexibility*. To

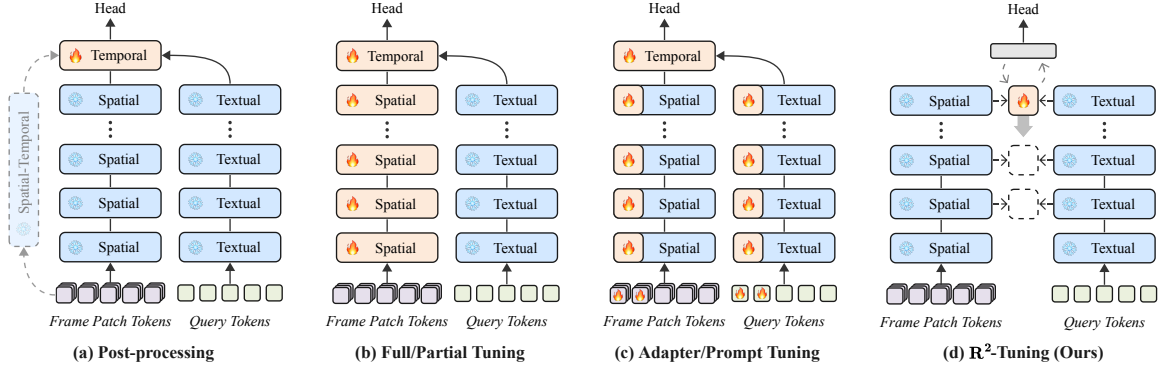


Figure 3.2: Different architectural designs for CLIP-based image-to-video transfer learning. The gray rectangle in (d) denotes the progressively refined spatial-temporal features.

address these issues, a novel image-to-video transfer learning framework called Reversed Recurrent Tuning (R²-Tuning) is proposed for fine-grained understanding on untrimmed videos. Our insight is that multi-layer CLIP features offer distinct yet useful information, while their integration should be tailored to the downstream task. As illustrated in Figure 3.2 (d), based on a frozen CLIP [205], our method learns a side-block (R² Block) containing only 1.5% of the total parameters to perform spatial-temporal modeling. R² Block is recurrently attached to the last few layers of CLIP encoders from back to front, performing *query-modulated spatial pooling* and *recurrent temporal refinement* from coarse to fine. Aside from the new architecture, we also introduce video-level and layer-wise constraints to calibrate the granularities of visual and text encoders. During training, gradient flows do not pass through the CLIP encoders, thus our scheme is both parameter- and memory-efficient. It is also granularity-flexible as R² Block can adaptively control the spatial pooling strategy conditioning on queries.

We conduct extensive experiments across three VTG tasks on six public benchmarks, including QVHighlights [120], Ego4D-NLQ [76], Charades-STA [70], TACoS [212], YouTube Highlights [234], and TVSum [230]. Without bells and whistles, R²-Tuning

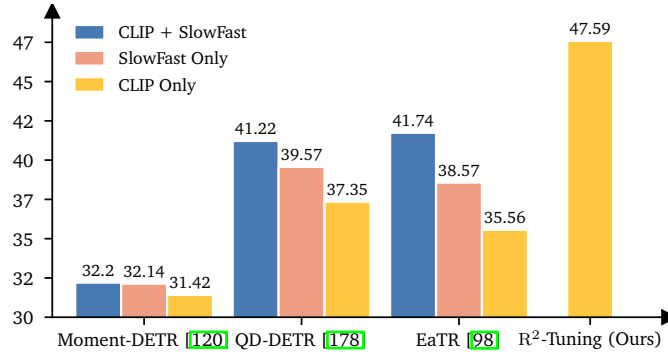


Figure 3.3: Moment retrieval mAP with different backbones on QVHighlights val split. CLIP’s potential of temporal modeling was not fully exploited.

achieves more than 3 MR mAP gain on QVHighlights test split compared with $4\times$ heavier counterparts [177] with additional temporal backbones and carefully designed post-processing modules. Further analysis also shows that our method can better handle multi-granularity information. Overall, our contributions are summarized as: (1) We introduced R²-Tuning, a novel image-to-video transfer learning framework tailored for video temporal grounding. (2) We designed two effective strategies, *i.e.*, *query-modulated spatial pooling* and *recurrent temporal refinement*, to model spatial-temporal information from coarse to fine. (3) To calibrate the granularities of CLIP visual and text encoders, we further introduce video-level and layer-wise contrastive constraints to distill distinct information from each layer. (4) Extensive experiments across three tasks on six benchmarks demonstrate the significance and effectiveness of R²-Tuning.

3.2 Related Work

3.2.1 CLIP for Video Understanding

With powerful transfer learning abilities, CLIP [205] has been widely used for many image-language tasks [150, 103], and how to extend them to video understanding

is an emerging and crucial research topic. Early attempts have been made to transfer CLIP to trimmed videos for action recognition [145, 183, 208] and video-text retrieval [106, 165, 17, 92]. Specifically, Efficient-Prompt [106] introduces learnable prompt tokens to the text encoder, and ViFi-CLIP [208] further adopts learnable prompts at the visual encoder. Other works also tried to learn external modules temporal modeling. EVL [145] proposes a temporal decoder in parallel with the main network. CLIP4Clip [165] investigated multiple temporal fusion strategies for video frames. Nevertheless, how to effectively transfer CLIP to untrimmed videos remains unexplored. To our best knowledge, R²-Tuning is the first solution for fine-grained video temporal grounding via memory-efficient transfer learning.

3.2.2 Video Temporal Grounding

Video temporal grounding (VTG) [140] is a bundle of video understanding problems including moment retrieval (MR) [70, 120, 155, 294, 178, 98, 298], highlight detection (HD) [234, 87, 292, 12], and video summarization (VS) [81, 230]. Although these tasks are closely related, they have not been jointly studied until recently. QVHighlights [120] is the first dataset that supports both MR and HD, breaking the moment distribution bias in previous MR datasets [212, 70]. The authors also proposed Moment-DETR, a strong baseline method for MR/HD. Most following works were built upon this framework with specially designed modules. Notably, UMT [155] tackles flexible MR and HD through a unified multi-modal architecture. QD-DETR [178] benefits from query-dependent video representations and dynamic anchors. EaTR [98] incorporates moment reasoning to provide referential search areas for moment queries. Nevertheless, their design space is still within pre-extracted CLIP [205] and SlowFast [63] features. A recent work [298] revealed the possibility of fine-tuning CLIP for MR, while their solution (post pre-training) is extremely resource-consuming. Compared with existing works, R²-Tuning can effectively learn strong temporal modeling abilities without any pre-training on videos.

3.2.3 Parameter-Efficient Transfer Learning

Foundation models [205, 245] have achieved remarkable successes in vision and language understanding. Efforts have been made to efficiently transfer knowledge from these models to new scenarios. One line of research is to conduct context-based tuning (e.g., prompt-tuning [345, 344, 101]), which does not modify the model architectures but incorporates learnable embeddings in the inputs. Another paradigm is introducing an extra lightweight adapter to the original model, and keeping the rest of the parameters frozen [192]. Recent works also consider memory-efficient transfer learning (METL) that introduces parameters whose gradient flows do not go through the main model [235, 145, 202]. R²-Tuning is also a METL framework, where we adopt a novel recurrent tuning strategy to ensure parameter- and memory-efficient.

3.3 Method

3.3.1 Problem Formulation

Given a video $V = \{v_i\}_{i=1}^T$ and a natural language query $Q = \{q_i\}_{i=1}^L$, where T and L are the numbers of video frames and text tokens, VTG aims to densely predict a set of labels (b_i, s_i, f_i) for each frame, defined as follows:

- **Moment Retrieval** is to find the most relevant moments (i.e., sets of consecutive frames) in V according to Q , so that $b_i = [b_i^s, b_i^e] \in \mathbb{R}^2$ represents the temporal displacements from frame v_i to the start and end timestamps of the nearest target moment.
- **Highlight Detection** requires estimating frame-level relevancies between V and Q , thus $s_i \in [0, 1]$ is a continuous saliency score denoting to what extent frame v_i is semantically aligned with Q .

- **Video Summarization** aims to select a subset of video frames according to Q to form a concise summary, therefore, $f_i \in \{0, 1\}$ is a binary score indicating whether frame v_i belongs to the summary.

3.3.2 Preliminary: The CLIP Model

The philosophy of CLIP [205] is to regard images and their corresponding descriptions as different views of the same concept, and to align them in the semantic space. The model has a two-stream architecture, consisting of a visual encoder (e.g., ViT [56]) and a text encoder (e.g., Transformer [248]). The input image I is first split into non-overlapping patches, while the text T is also tokenized into discrete tokens. They are then mapped into a shared semantic space by the two encoders. This process is optimized using a symmetric InfoNCE [186] loss $\mathcal{L}$:

$$\mathcal{L}_{I \rightarrow T} = -\frac{1}{B} \sum_{i=1}^B \log \frac{\exp(\mathbf{v}_i^\top \mathbf{q}_i / \tau)}{\sum_{j=1}^B \exp(\mathbf{v}_i^\top \mathbf{q}_j / \tau)} \quad (3.1)$$

$$\mathcal{L}_{T \rightarrow I} = -\frac{1}{B} \sum_{i=1}^B \log \frac{\exp(\mathbf{q}_i^\top \mathbf{v}_i / \tau)}{\sum_{j=1}^B \exp(\mathbf{q}_i^\top \mathbf{v}_j / \tau)} \quad (3.2)$$

$$\mathcal{L} = \frac{1}{2}(\mathcal{L}_{I \rightarrow T} + \mathcal{L}_{T \rightarrow I}) \quad (3.3)$$

Here, $\mathbf{v}$ and $\mathbf{q}$ are encoded image and text embeddings, i and j are indices of samples in a batch with size B , and τ is a temperature parameter controlling the smoothness of the softmax distribution. After training, the paired $\mathbf{v}$ and $\mathbf{q}$ shall be well-aligned in their shared semantic space.

3.3.3 Overview

Figure 3.4 shows an overview of the proposed framework. Our model derives from a pre-trained and frozen CLIP [205] with ViT [56] backbone, which has a two-stream architecture for spatial and textual encoding, respectively. A learnable $\mathbf{R}^2$ block is

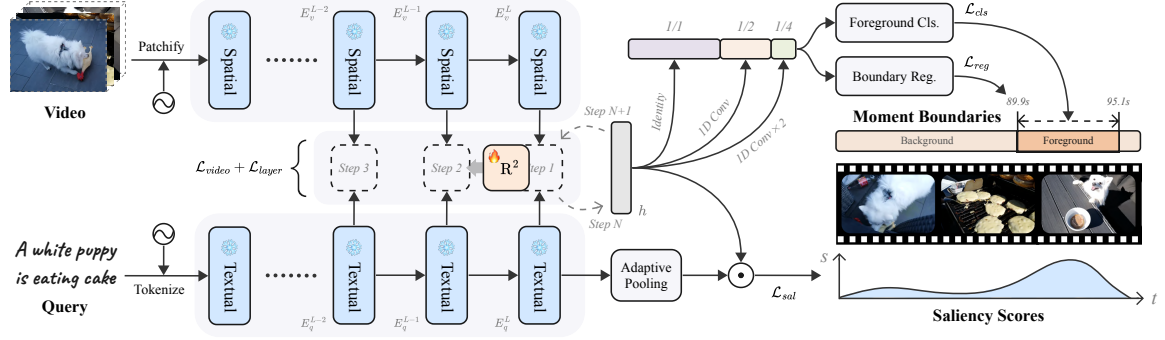


Figure 3.4: Overall architecture of our framework. The input video and query are first encoded by frozen CLIP [205] encoders. Their multi-layer outputs are then recurrently fused and refined by a learnable R^2 Block to construct spatial-temporal representations h , which would be scaled up/down to construct a temporal feature pyramid, followed by three heads for MR, HD, and VS, respectively.

iteratively attached to the last K encoder layers to refine the temporal correlation. The resulting features are then scaled up/down to build a feature pyramid, followed by three heads to decode the task-specific predictions.

Specifically, the input video V and query Q are first tokenized into frame patches and word tokens, then sent into the visual encoder E_v and query encoder E_q . The encoded visual and textual features can be denoted as $e_v \in \mathbb{R}^{B \times N \times T \times (P+1) \times D_v}$ and $e_q \in \mathbb{R}^{B \times N \times L \times D_q}$, where B , N , T , P , L , D_v and D_q indicate batch size, #encoder layers, #video frames, #patches per frame, #query tokens, and the dimensions of visual/query features, respectively. These features are recurrently fused and refined by R^2 block to construct spatial-temporal features $h \in \mathbb{R}^{B \times T \times C}$, in which each token preserves the C -dimensional features for a frame.

3.3.4 Reversed Recurrent Tuning

Given pre-trained visual and query encoders, existing works [120, 155, 140, 178, 98] merely take features from the last hidden layer, which is sub-optimal due to the

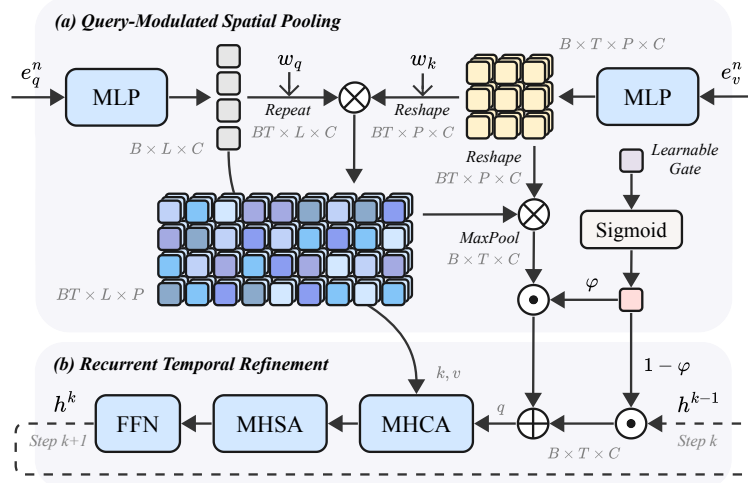


Figure 3.5: Detailed architecture of the R^2 Block. It can be split into two parts: (a) *query-modulated spatial pooling*, and (b) *recurrent temporal refinement*. Note that the [CLS] tokens of visual features are omitted for clarity.

limited information. In this work, we propose R^2 -Tuning to exploit the potential of spatial-temporal modeling based on multi-layer CLIP features. Our scheme learns a lightweight R^2 Block which maintains a zero-initialized hidden state $h \in \mathbb{R}^{B \times T \times C}$ as frame-level spatial-temporal features. The R^2 Block is recurrently attached to the last K layers of CLIP encoders from back to front to refine the hidden state h for K steps, with each step defined as follows:

$$h^k = \mathcal{F}_\theta(e_v^n, e_q^n, h^{k-1}) \quad (3.4)$$

Here, $\mathcal{F}_\theta$ is the refinement operation parameterized by θ . $k \in [1, K]$ is the index of current step. $n = N - k + 1$ denotes the CLIP layer index for step k , thus e_v^n and e_q^n are visual and query features from the n -th CLIP encoder layer. The refinement operation $\mathcal{F}_\theta$ can be decomposed into two parts: (1) *query-modulated spatial pooling*, and (2) *recurrent temporal refinement*. Details are discussed as follows.

Query-Modulated Spatial Pooling Figure 3.5 (a) presents the data flow of query-modulated spatial pooling. The goal of this process is to adaptively pool spatial

features from patch-level representations e_v^n to a single token e_{pool}^n conditioning on the query e_q^n . We first adopt two separate MLPs to map e_v^n and e_q^n into the same space, formulated as follows:

$$\hat{e}_v^n = \text{MLP}(e_v^n) \in \mathbb{R}^{B \times T \times (P+1) \times C} \quad (3.5)$$

$$\hat{e}_q^n = \text{MLP}(e_q^n) \in \mathbb{R}^{B \times L \times C} \quad (3.6)$$

Here, C is the reduced hidden size. We further align the shape of $\hat{e}_v^n$ and $\hat{e}_q^n$ by merging dimension T into B and repeat $\hat{e}_q^n$ for T times in an interleaved style, resulting $\hat{e}_v^n \in \mathbb{R}^{(B \times T) \times (P+1) \times C}$ and $\hat{e}_q^n \in \mathbb{R}^{(B \times T) \times L \times C}$. Then, we compute the similarities for patch-token pairs using normalized Embedded Gaussian [248]:

$$a = \text{softmax}\left(\frac{(w_q \hat{e}_q^n)^\top w_v \hat{e}_v^n}{\sqrt{C}}\right) \in \mathbb{R}^{(B \times T) \times L \times P} \quad (3.7)$$

Here, w_v and w_q are learnable matrices for projecting features. This operation derives from cross-attention but discards the linear projection on *value*. We then multiply the similarities a with visual features $\hat{e}_v^n$ to pool them into each token, such that tokens can interact with patches independently. After that, a max pooling along the L dimension is applied:

$$e_{token}^n = \text{MaxPooling}(a \hat{e}_v^n + \hat{e}_q^n) \in \mathbb{R}^{B \times T \times C} \quad (3.8)$$

Here, $\hat{e}_q^n$ serves as residuals [84] to stabilize training. Our visualizations in Figure 3.7 show that this strategy can guide the model to focus more on query-related regions. Finally, we combine e_{token}^n with the [CLS] token to generate query-modulated spatial features. This is realized by a weighted-sum operation:

$$e_{pool}^n = e_v^{n,0} + g^k \cdot e_{token}^n \in \mathbb{R}^{B \times T \times C} \quad (3.9)$$

Here, $g^k \in (-1, 1)$ is a zero-initialized learnable gate for step k constrained by Tanh. We allow negative values here to remove useless information from [CLS] tokens. After that, e_{pool}^n is then used to model temporal correlations and are refined in a recurrent manner.

Recurrent Temporal Refinement Figure 3.5 (b) displays the flow of recurrent temporal refinement. Specifically, the pooled visual features e_{pool}^n is first fused with the hidden state h^{k-1} from the previous step:

$$\hat{h}^{k-1} = \varphi^k \cdot e_{pool}^n + (1 - \varphi^k) \cdot h^{k-1} \quad (3.10)$$

where $\varphi^k \in (0, 1)$ is the learnable gate for step k . Then, we adopt a sequence of standard multi-head cross-attention ($\hat{e}_q^n$ as k, v), multi-head self-attention, followed by a feed-forward network [248] to update the hidden state:

$$h^k = \text{FFN}(\text{MHSA}(\text{MHCA}(\hat{h}^{k-1}, \hat{e}_q^n))) \quad (3.11)$$

For each block, we adopt DropPath [117] with $p = 0.1$ to prevent overfitting. The query features $\hat{e}_q^n$ are utilized both in spatial and temporal modeling as guidance.

Granularity Calibration The refinement process above is based on the assumption that visual and query features from the same layer of CLIP [205] are well aligned at the same granularity level. However, this cannot be guaranteed as the two encoders are learned in isolation during pre-training. Therefore, we need to add manual constraints to calibrate their granularities.

We apply a *video-level constraint* and a *layer-wise constraint* by designing two contrastive losses. We first denote $e_v^0 \in \mathbb{R}^{B \times K \times T \times C}$ as the features of K [CLS] tokens from Eq. 3.9. Then, we select all positive frames (i.e., frames within $[b_i^s, b_i^e]$ or with s_i or f_i higher than a threshold) from e_v^0 and perform average pooling to obtain video-level representations $\tilde{e}_v \in \mathbb{R}^{B \times K \times C}$.

$$\tilde{e}_v = \text{AvgPool}(\{e_v^{0,i}\}_{i \in \Omega}) \quad (3.12)$$

Here, Ω is the set of positive frame indices. To obtain query-level representations $\tilde{e}_q \in \mathbb{R}^{B \times K \times C}$, a token-wise adaptive pooling is applied following previous work [140]. We adopt InfoNCE loss [186] along two dimensions to calculate the video-

level and layer-wise constraints:

$$\mathcal{L}_{video} = \lambda_{video} \frac{1}{B} \sum_{b \in B} \text{InfoNCE}(\tilde{e}_v^b, \tilde{e}_q^b) \quad (3.13)$$

$$\mathcal{L}_{layer} = \lambda_{layer} \frac{1}{K} \sum_{k \in K} \text{InfoNCE}(\tilde{e}_v^k, \tilde{e}_q^k) \quad (3.14)$$

Here, $\mathcal{L}_{video}$ performs contrast among samples in the same batch (diversifying features are diverse among moment-query pairs), and averages the loss across K layers. It also implicitly models temporal correlations between videos and queries. $\mathcal{L}_{layer}$ performs contrast among layers and averages across the batch. λ_{video} and λ_{layer} serve as re-weighting terms for the losses. The temperature is set to 0.07.

3.3.5 Prediction Heads

After refining spatial-temporal features h , a temporal feature pyramid is constructed by applying 1D convolutions with $stride = 2$ on h . The number of convolutions used for each level is subjected to $l - 1$, where l is the level index starting from 1. We concatenate features from all levels by the temporal dimension to form $\hat{h}$ and predict once in all heads. Following a similar but simplified design from [140], we adopt three heads for video temporal grounding, described in detail as follows.

Foreground-Background Classification Given $\hat{h}$, a two-layer 1D convolution module with (1×3) kernels is adopted to predict $\hat{f}_i$ for each frame. It is optimized using a focal loss [143] with $\alpha = 0.9$ and $\gamma = 2.0$:

$$\mathcal{L}_{cls} = -\lambda_{cls} \alpha (1 - \hat{f}_i)^\gamma \log(\hat{f}_i) \quad (3.15)$$

We find that focal loss perform better than binary cross-entropy when the numbers of foreground/background frames are severely imbalanced as noted in [140].

Boundary Regression Similar to foreground-background classification, boundary regression for moments is also realized by a two-layer 1D convolution module. The

only difference is that its output dimension is set to 2 instead of 1, representing the boundary displacements $[\hat{b}_i^s, \hat{b}_i^e]$ for start-end timestamps. The outputs from head is supervised by an L1 loss:

$$\mathcal{L}_{reg} = \lambda_{reg}(|b_i^s - \hat{b}_i^s| + |b_i^e - \hat{b}_i^e|) \quad (3.16)$$

where b_i^s and b_i^e are the ground truths. This loss is only applied to frames inside the ground truth boundaries. We observe that using an L1 loss already works better than the combination of Smooth L1 Loss and GIoU Loss as in [140].

Saliency Prediction To obtain the saliency scores $\hat{s}_i$ for HD, we calculate the cosine similarities between the adaptive pooled query features at the last refinement step $\tilde{e}_q^K$ and each token in the spatial-temporal features h :

$$\hat{s}_i = \frac{h_i^\top \tilde{e}_q^K}{\|h_i\|_2 \|\tilde{e}_q^K\|_2} \quad (3.17)$$

The training objective is applied through a contrastive loss between sampled positive frames (with index $p \in P$) and the adaptively pooled query $\tilde{e}_q^K$:

$$\mathcal{L}_{sal} = -\lambda_{sal} \log \frac{\exp(\hat{s}_p/\tau)}{\exp(\hat{s}_p/\tau) + \sum_{i \in \Theta} \exp(\hat{s}_i/\tau)} \quad (3.18)$$

Here, Θ is the set of frame indices where $s_i < s_p$, and τ is a fixed temperature parameter set to 0.07.

3.3.6 Training and Inference

The whole model is jointly optimized using a sum of the five losses mentioned above. In practice, the loss weights are set as $\lambda_{video} = 0.1$, $\lambda_{layer} = 0.1$, $\lambda_{cls} = 1.0$, $\lambda_{reg} = 0.1$, and $\lambda_{sal} = 0.1$. During inference, we form an moment retrieval prediction by combining f_i and b_i , i.e., calculating the start-end timestamps from frame index i and boundary displacements $[b_i^s, b_i^e]$, while regarding f_i as the confidence.

Non-maximum suppression (NMS) with IoU threshold $\theta_{IoU} = 0.7$ is applied to reduce duplicates. For highlight detection and video summarization, we directly use the frame-level output from saliency prediction s_i .

3.3.7 Discussion: Detailed Comparison with Counterparts

In order to further justify our novelty and distinguish our design from previous methods, we provide an in-depth comparison between our method and existing ones. To evaluate the effectiveness of a transfer learning method for VTG, we define four design principles as follows:

- P1. Strong Spatial-Temporal Modeling Ability:** VTG requires the model to regress moment temporal boundaries or densely predict frame-level scores. Hence, strong spatial-temporal modeling abilities are needed to accurately ground the video.
- P2. Parameter-Efficient:** As fine-tuning all the parameters of CLIP is resource-consuming and may even lead to catastrophic forgetting [112], the transfer learning framework should only learn a few parameters to take full advantage of the pre-trained knowledge.
- P3. Memory-Efficient:** Although some recent transfer learning frameworks are parameter-efficient [101, 192], they might not be memory-efficient as computing the gradients of these learnable parameters requires back-propagating through the whole network, consuming a large amount of GPU memory.
- P4. Granularity-Flexible:** Queries for VTG might be of different granularity levels. The capability of parsing multi-granularity queries and extracting appropriate features accordingly is essential for flexible VTG.

We then systematically compare our proposed scheme with previous ones from two aspects, *architecture* and *capabilities*, respectively.

Table 3.1: Capability comparison among different image-to-video transfer learning frameworks. Our proposed R²-Tuning is the only one satisfying all the principles.

Framework	P1	P2	P3	P4
(a) Post-processing [120, 155, 178, 98]	✗	✓	✓	✗
(b) Full/Partial Tuning [208, 298]	✓	✗	✗	✗
(c) Adapter/Prompt Tuning [101, 192]	✓	✓	✗	✗
(d) R ² -Tuning (Ours)	✓	✓	✓	✓

Architecture As illustrated in Figure 3.2, **Post-processing** methods [120, 155, 294, 178, 98, 165] follow a straightforward strategy that simply regarding CLIP as a spatial & textual feature extractor. All the temporal modeling capabilities are realized by carefully designed modules after CLIP. An auxiliary backbone (e.g., SlowFast [63]) is also utilized to complement the spatial-temporal information. **Full/Partial Tuning** methods [208, 298] unfreeze the visual and/or textual stream of CLIP, and jointly learn it with temporal modeling modules. These models are usually combined with post pre-training due to the potential catastrophic forgetting [112]. **Adapter/Prompt Tuning** methods [101, 192, 106, 17, 208, 92] learn lightweight adapters or prompt tokens to transfer the pre-trained knowledge. These learnable weights may perturb the original attention patterns [198]. Our **R²-Tuning**, instead, learns an adapter that be used as a side module [235, 202, 53] for the main network, preserving the information patterns while taking advantages of the multi-level representations. Compared with the recent side-tuning methods, our approach shows its advantages by leveraging the *reversed* fusion path and the *recurrent* refinement strategy.

Capabilities Table 3.1 compares the capabilities of different image-to-video transfer learning frameworks. **Post-processing** methods [120, 155, 294, 178, 98, 165] are naturally parameter- and memory-efficient, as they only require to learn temporal modules on top of pre-extracted CLIP features. Nevertheless, these methods largely rely on the auxiliary temporal backbone as frame-level CLIP embeddings

limit the temporal modeling capabilities. Merely using the last hidden features is not granularity-flexible as well. **Full/Partial Tuning** methods [208, 298] may have slightly better spatial-temporal modeling as they fine-tuned the spatial encoders on videos. But the large amount of learnable parameters make the fine-tuning process inefficient. **Adapter/Prompt Tuning** methods [101, 192, 106, 17, 208, 92] could model spatial-temporal information by designing special adapters [192] or prompts [92]. However, training these adapters/prompts might consume high GPU memory, as the learnable parameters are tightly bound with frozen ones, thus the gradient computation cannot be disentangled. To the best of our knowledge, **R²-Tuning** is the only scheme that satisfies all the design principles.

3.4 Experiments

3.4.1 Datasets & Evaluation Metrics

We conduct experiments on three VTG tasks across six datasets covering various domains including daily vlogs & news, in-door scenes, egocentric videos, and sports. They are introduced in detail as follows.

Joint Video Moment Retrieval and Highlight Detection We adopt QVHighlights [120] since this is the only dataset that supports both moment retrieval and highlight detection. It contains 10,148 videos, each trimmed to around 150 seconds long. The domains of the videos cover from daily vlogs, travel vlogs, to news events. A total of 10,310 queries with 18,367 disjoint moments are manually annotated. This dataset also has a private test split for reliable evaluation.

Video Moment Retrieval As for moment retrieval, we adopt Ego4D-NLQ (Natural Language Query) [76], Charades-STA [70], and TACoS [212] datasets. Ego4D-NLQ contains 1.3K videos with 8~20 minutes durations under daily egocentric

Table 3.2: Hyperparameters used for different datasets. LR Drop means the epoch that drops learning rate by $1/10$. * Video clips in YouTube Highlights have overlaps. We adopt the same setting as UMT [155].

Dataset	Pyramid Depth	FPS	Batch Size	Learning Rate	Epochs	Warmup Iters	LR Drop
QVHighlights [120]	4	0.5	128	$5e^{-4}$	30	500	20
Ego4D-NLQ [76]	4	0.5	32	$2.5e^{-4}$	30	500	20
Charades-STA [70]	4	1.0	32	$2.5e^{-4}$	50	500	30
TACoS [212]	4	0.5	32	$2.5e^{-4}$	100	500	50
YouTube Highlights [234]	1	*	4	$5e^{-4}$	200	50	–
TVSum [230]	1	0.5	4	$5e^{-4}$	500	50	–

scenarios. 15.2K queries in the form of questions (*e.g.*, *where did I put the plier*) are annotated with precise moments. Charades-STA is an extension of the Charades [225] dataset with moment annotations. It has 9,848 unique in-door videos with an average length of 30.6s, and a total of 16,128 query-moment pairs. TACoS is a small-scale dataset with 273 videos, each with around 4.8 minutes long. There are 9,790, 4,436, and 4,001 queries for training, validation, and testing, respectively.

Video Highlight Detection We utilize YouTube Highlights [234] dataset to evaluate the highlight detection performance. This dataset contains videos under 6 domains, *i.e.*, *dog*, *gymnastics*, *parkour*, *skating*, *skiing*, and *surfing*. As some of the videos are no longer available, we follow previous works [155, 140] to use a subset that contains 435 videos. The name of video domains are used as queries directly.

Video Summarization TVSum [230] dataset is leveraged for video summarization evaluation. It contains 10 domains, each with 5 videos. We follow the same setting as previous works [155, 140] that use 4 videos for training and 1 video for testing. The video titles are used as queries. Note that TVSum [230] is regarded as a highlight detection dataset in [140] but as a video summarization dataset in our work. We claim that both choices are reasonable as HD and VS essentially share similar problem formulations. The only differences are that: (1) only a small pro-

Table 3.3: Video moment retrieval (MR) and highlight detection (HD) results on QVHighlights test split. Note that [98] and [298] only reported their results on val split. $\textcircled{P}$ and $\textcircled{G}$ are CLIP-B/32 and B/16 [205], $\textcircled{S}$ denotes SlowFast R-50 [63], and $\textcircled{A}$ is PANN [113] for audio features extraction. * means estimated parameters. The best and second-best results are marked with **bold** and underline, respectively.

Method	Backbone	Post Pre-train	MR					HD		#Params
			R1		mAP			≥ Very Good		
			@0.5	@0.7	@0.5	@0.75	Avg.	mAP	HIT@1	
BeautyThumb [229]	🟡🔴	✗	–	–	–	–	–	14.36	20.88	–
DVSE [152]	🟡🔴	✗	–	–	–	–	–	18.75	21.79	–
MCN [6]	🟡🔴	✗	11.41	2.72	24.94	8.22	10.67	–	–	–
CAL [59]	🟡🔴	✗	25.49	11.54	23.40	7.65	9.89	–	–	–
XML [121]	🟡🔴	✗	41.83	30.35	44.63	31.73	32.14	34.49	55.25	–
XML+ [120]	🟡🔴	✗	46.69	33.46	47.89	34.67	34.90	35.38	55.06	–
Moment-DETR [120]	🟡🔴	✗	52.89	33.02	54.82	29.40	30.73	35.69	55.60	4.8M
UMT [155]	🟡🟢🟠	✗	56.23	41.18	53.83	37.01	36.12	38.18	59.99	14.9M
MomentDiff [128]	🟡🟢🟠	✗	58.21	41.48	54.57	37.21	36.84	–	–	–
QD-DETR [178]	🟡🔴	✗	62.40	44.98	62.52	39.88	39.86	38.94	62.40	7.6M
MH-DETR [294]	🟡🔴	✗	60.05	42.48	60.75	38.13	38.38	38.22	60.51	8.2M
UniVTG [140]	🟡🔴	✗	58.86	40.86	57.60	35.59	35.47	38.20	60.96	41.3M
TR-DETR [233]	🟡🔴	✗	64.66	48.96	63.98	43.73	42.62	39.91	63.42	7.9M
CG-DETR [177]	🟡🔴	✗	<u>65.43</u>	48.38	<u>64.51</u>	42.77	42.86	40.33	<u>66.21</u>	12.0M
EaTR [98]	🟡🔴	✗	61.36	45.79	61.86	41.91	41.74	37.15	58.65	9.0M
Moment-DETR [120]	🟡🔴	236K ASR Cap.	59.78	40.33	60.51	35.36	36.14	37.43	60.17	4.8M
UMT [155]	🟡🟢🟠	236K ASR Cap.	60.83	43.26	57.33	39.12	38.08	39.12	62.39	14.9M
UniVTG [140]	🟡🔴	4.2M Corpus	<u>65.43</u>	50.06	64.06	<u>45.02</u>	<u>43.63</u>	<u>40.54</u>	66.28	41.3M
UnLoc [298]	🟢	650K Videos	64.50	48.80	–	–	–	–	–	87.9M*
R²-Tuning (Ours)	🟡	✗	68.03	<u>49.35</u>	69.04	47.56	46.17	40.75	64.20	2.7M

portion (e.g., 20%) of the video clips are regarded as summary for VS, while there is no limitation on number of highlight clips for HD, and (2) VS requires to quantize the output scores to binary values in practice. Previous works on HD [155, 292, 12] and VS [81, 9] also mixed these datasets when conducting evaluations.

Evaluation Metrics We adopt the same evaluation metrics with previous works [120, 140, 155]. Specifically, we compute Recall@1 with IoU threshold $\theta_{IoU} = 0.5$ and 0.7, mAP with $\theta_{IoU} = 0.5$ and 0.7, and mAP with a series of thresholds

Table 3.4: Video moment retrieval results on Ego4D-NLQ, Charades-STA, and TACoS datasets.  means using both CLIP-B/32 and SlowFast R-50 as feature extractors.  indicates using CLIP-B/32 only. The best and second-best results are marked with **bold** and underline, respectively. Our method shows significant advantages in high-quality retrievals (R@0.7) even without a video backbone.

Method	Ego4D-NLQ [76]				Charades-STA [70]				TACoS [212]			
	R@0.3	R@0.5	R@0.7	mIoU	R@0.3	R@0.5	R@0.7	mIoU	R@0.3	R@0.5	R@0.7	mIoU
 2D-TAN [334]	4.33	1.83	0.60	3.39	58.76	46.02	27.50	41.25	40.01	27.99	12.92	27.22
 VSLNet [329]	4.54	2.40	1.01	3.54	60.30	42.69	24.14	41.58	35.54	23.54	13.15	24.99
 Moment-DETR [120]	4.34	1.81	0.65	3.53	65.83	52.07	30.59	45.54	37.97	24.67	11.97	25.49
 UniVTG [140]	7.28	<u>3.95</u>	<u>1.32</u>	<u>4.91</u>	<u>70.81</u>	<u>58.01</u>	<u>35.65</u>	<u>50.10</u>	51.44	<u>34.97</u>	<u>17.35</u>	<u>33.60</u>
 R ² -Tuning (Ours)	<u>7.20</u>	4.49	2.12	4.94	70.91	59.78	37.02	50.86	<u>49.71</u>	38.72	25.12	35.92

[0.5:0.05:0.95] for MR on QVHighlights [120]. mAP and HIT@1 where positive samples are defined as with the saliency score of *Very Good* are adopted for HD. On Ego4D-NLQ [76], Charades-STA [70], and TACoS [212] datasets, we utilize Recall@1 with $\theta_{IoU} = \{0.3, 0.5, 0.7\}$ and mIoU to measure the MR performance. On YouTube Highlights [234] for HD and TVSum [230] for VS, we follow [155] and use the same train/val split with mAP and Top-5 mAP as metrics, respectively.

3.4.2 Implementation Details

In all experiments, we adopt the visual and text encoders of CLIP ViT-B/32 [205] as our backbones. The whole CLIP model is frozen during training, *i.e.*, only the parameters in R² Block, temporal feature pyramid, prediction heads are learnable. Without further specification, we set $K = 4$, *i.e.*, attaching R² Block to the last 4 layers of CLIP encoders. In R² Block, the hidden size is set to 256. A one-layer transformer encoder with 8 attention heads is used for temporal modeling.

Table 3.2 lists the dataset-specific hyperparameters. The strides of temporal feature

Table 3.5: Class-wise video highlight detection results (mAP) on YouTube Highlights dataset. $\textcircled{\text{P}}$, $\textcircled{\text{V}}$, and $\textcircled{\text{A}}$ denote image, video, and audio backbones, respectively.

Method	Dog	Gym.	Par.	Ska.	Ski.	Sur.	Avg.
$\textcircled{\text{V}}$ RRAE [301]	49.0	35.0	50.0	25.0	22.0	49.0	38.3
$\textcircled{\text{V}}$ GIFs [82]	30.8	33.5	54.0	55.4	32.8	54.1	46.4
$\textcircled{\text{V}}$ LSVM [234]	60.0	41.0	61.0	62.0	36.0	61.0	53.6
$\textcircled{\text{V}}$ LIM-S [287]	57.9	41.7	67.0	57.8	48.6	65.1	56.4
$\textcircled{\text{V}}$ SL-Module [292]	70.8	53.2	77.2	72.5	66.1	76.2	69.3
$\textcircled{\text{V}}$ PLD [271]	74.9	70.2	77.9	57.5	70.7	79.0	73.0
$\textcircled{\text{V}}$ MINI-Net [87]	58.2	61.7	70.2	72.2	58.7	65.1	64.4
$\textcircled{\text{V}}$ TCG [307]	55.4	62.7	70.9	69.1	60.1	59.8	63.0
$\textcircled{\text{V}}$ Joint-VA [12]	64.5	71.9	80.8	62.0	73.2	78.3	71.8
$\textcircled{\text{V}}$ CO-AV [129]	60.9	66.0	89.0	74.1	69.0	81.1	74.7
$\textcircled{\text{V}}$ UMT [155]	65.9	75.2	81.6	71.8	72.3	82.7	74.9
$\textcircled{\text{V}}$ UniVTG [140]	71.8	76.5	73.9	73.3	73.2	82.2	75.2
$\textcircled{\text{P}}$ R ² -Tuning (Ours)	75.6	73.5	73.0	74.6	74.8	84.8	76.1

pyramid is set to (1, 2, 4, 8). For all videos, we directly resize and crop the frames to 224×224 without augmentations. The same regularization methods with CLIP pre-training is adopted to aligned the inputs. The maximum number of query tokens is set to 77. We train our model with FP16 on a single NVIDIA A100 (80G) GPU.

3.4.3 Comparison with State-of-the-Arts

Joint Moment Retrieval and Highlight Detection We first evaluate our method on QVHighlights [120] test split, the only dataset that supports both moment retrieval and highlight detection. The results are shown in Table 3.3. The first group of methods utilizes more than one feature extractor. The second group is that with large-scale post pre-training (*i.e.*, the extra pre-training stage between loading backbone weights and training on downstream datasets.). Without any additional backbone and extra pre-training, R²-Tuning achieves start-of-the-art performance compared with all previous methods with the fewest learnable parameters.

Table 3.6: Class-wise video summarization results (Top-5 mAP) on TVSum dataset.

⬤, ⬤, and ⬤ denote image, video, and audio backbones, respectively. Our method does not require any additional video/audio features.

Method	VT	VU	GA	MS	PK	PR	FM	BK	BT	DS	Avg.
⬤ sLSTM [331]	41.1	46.2	46.3	47.7	44.8	46.1	45.2	40.6	47.1	45.5	45.1
⬤ SG [172]	42.3	47.2	47.5	48.9	45.6	47.3	46.4	41.7	48.3	46.6	46.2
⬤ LIM-S [287]	55.9	42.9	61.2	54.0	60.4	47.5	43.2	66.3	69.1	62.6	56.3
⬤ Trailer [254]	61.3	54.6	65.7	60.8	59.1	70.1	58.2	64.7	65.6	68.1	62.8
⬤ SL-Module [292]	86.5	68.7	74.9	86.2	79.0	63.2	58.9	72.6	78.9	64.0	73.3
⬤ PLD [271]	84.5	80.9	70.3	72.5	76.4	<u>87.2</u>	71.9	74.0	74.4	<u>79.1</u>	77.1
⬤ MINI-Net [87]	80.6	68.3	78.2	81.8	78.1	65.8	57.8	75.0	80.2	65.5	73.2
⬤ TCG [307]	85.0	71.4	81.9	78.6	80.2	75.5	71.6	77.3	78.6	68.1	76.8
⬤ Joint-VA [12]	83.7	57.3	78.5	<u>86.1</u>	80.1	69.2	70.0	73.0	97.4	67.5	76.3
⬤ CO-AV [129]	90.8	72.8	84.6	85.0	78.3	78.0	72.8	77.1	89.5	72.3	80.1
⬤ UMT [155]	<u>87.5</u>	81.5	88.2	78.8	81.4	87.0	<u>76.0</u>	86.9	84.4	79.6	<u>83.1</u>
⬤ UniVTG [140]	83.9	<u>85.1</u>	<u>89.0</u>	80.1	<u>84.6</u>	81.4	70.9	91.7	73.5	69.3	81.0
⬤ R ² -Tuning (Ours)	85.0	85.9	91.0	81.7	88.8	87.4	78.1	<u>89.2</u>	<u>90.3</u>	74.7	85.2

Video Moment Retrieval We then evaluate our model on moment retrieval task in egocentric [76] and in-door [212, 70] domains. The results are shown in Table 3.4. We follow [140] and compare only with the methods using CLIP + SlowFast features. R²-Tuning still works better than all baselines with extra features. We also observe that R²-Tuning shows its significance on high-quality retrievals (R1@0.7), which requires accurate temporal modeling. This benefits from the fine-grained temporal modeling ability of reversed and recurrent designs.

Video Highlight Detection & Video Summarization The performances of highlight detection on YouTube Highlights [234] and extractive video summarization on TVSum [230] are reported in Table 3.5 and Table 3.6, respectively. Following previous works [155, 140], we trained the model separately on each domain. The first group of methods is based on video backbones, which naturally have temporal modeling abilities. The second group is enhanced by extra image or audio features. R²-Tuning still reaches the best performance when trained on small-scale data.

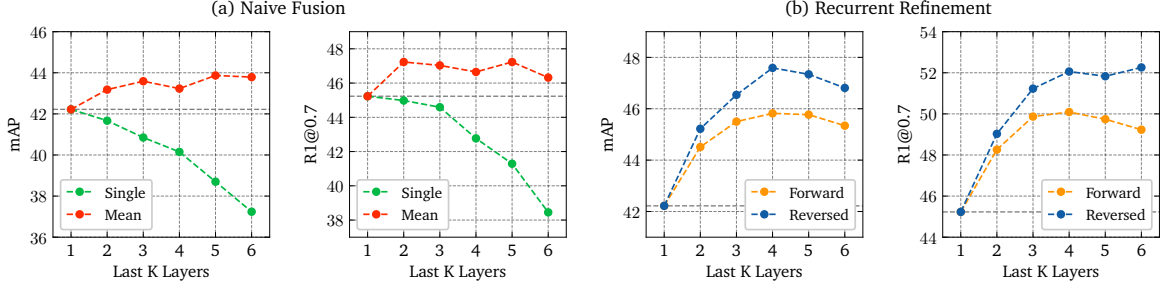


Figure 3.6: Different feature refinement strategies for MR on QVHighlights val split. Single means using the K -th layer only. Mean denotes averaging the features from last K layers. Forward and Reversed indicate different refinement directions.

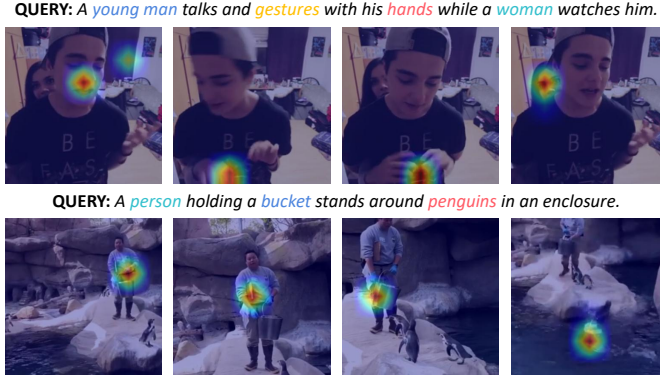


Figure 3.7: Visualization of the attention maps for query-modulated spatial pooling. Diverse regions are learned for different queries.

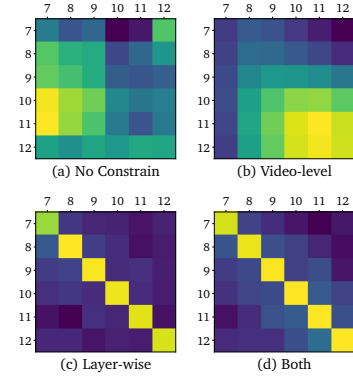


Figure 3.8: Cosine similarities of video (x-axis) and text (y-axis) among 7~12 CLIP layers.

3.4.4 Ablation Studies

Multi-Granularity Features To investigate the significance of granularity flexibility, we set up a simple baseline that only attaches R^2 Block to one layer. Then, we fuse the multi-layer information by averaging the features from the last K layers. These two variants are compared in Figure 3.6 (a). For the single-layer setting, higher-level features are more discriminative than lower-level ones. However, even a naive multi-layer fusion strategy brings significant performance gains.

Table 3.7: Effectiveness justification of the combination of granularity calibration losses on QVHighlights val split.

$\mathcal{L}_{video}$	$\mathcal{L}_{layer}$	MR			HD	
		R1@0.5	R1@0.7	mAP	mAP	HIT@1
✓	✓	64.48	48.60	44.01	37.94	62.67
		<u>67.68</u>	<u>51.61</u>	<u>46.74</u>	<u>39.81</u>	<u>65.16</u>
✓	✓	64.71	48.84	44.60	38.91	63.35
✓	✓	68.71	52.06	47.59	40.59	64.32

Table 3.8: MR mAP with different query lengths on QVHighlights val split.

Method	#Words in the Query				
	1-10	11-20	21-30	31-40	≥ 41
QD-DETR [178]	40.30	42.32	29.01	0.10	26.67
EaTR [98]	38.94	39.54	24.58	<u>12.03</u>	<u>41.43</u>
UniVTG [140]	32.38	32.65	20.88	0.00	31.11
R²-Tuning (Ours)	47.08	49.13	<u>28.79</u>	67.24	72.38

Reversed Recurrent Feature Refinement We then further justify our design by comparing two refinement directions (*i.e.*, forward and reversed) in Figure 3.6 (b). Note that when $K = 1$, all four designs in Figure 3.6 are of the same architecture. When utilizing more layers, both forward and reversed refinement benefit from more information, while the reversed scheme statically performs better, as the multi-granularity features shall be refined from coarse to fine.

Query-Modulated Spatial Pooling We visualize the attention maps learned by query-modulated spatial pooling in Figure 3.7. A query generally has multiple keywords that provide discriminative information to localize moments. The token-wise attention in Eq. 3.7 can guide the model to focus on multiple regions simultaneously, as can be seen in the diverse but meaningful patterns.

Granularity Calibration To verify the effectiveness of granularity calibration, we also visualize the cosine distances of all the visual-query pairs from 7~12 lay-

Table 3.9: Comparison among different PETL and METL methods on QVHighlights val split. #Params and Memory represent the number of learnable parameters and peak GPU memory (224×224 inputs with 32 batch size & FP16), respectively.

Method	#Params (M)	Memory (GB)	MR			HD	
			R1@0.5	R1@0.7	mAP	mAP	HIT@1
w/o Tuning	<u>2.7</u>	4.3	64.19	45.03	42.22	39.05	63.16
Full-Tuning	147.0	58.5	64.69	45.86	42.41	38.55	62.50
LoRA [88]	4.4	55.4	63.70	44.93	41.29	38.47	61.09
VPT [101]	2.8	42.9	64.55	45.97	42.36	38.56	62.48
VoP [92]	14.5	45.2	65.23	46.01	42.45	38.70	62.75
ST-Adapter [192]	12.2	44.3	65.19	<u>48.58</u>	44.27	39.52	63.49
E ³ VA [309]	3.3	10.8	65.10	48.38	<u>45.06</u>	39.45	63.38
LoSA [93]	4.8	10.9	65.06	48.43	44.97	39.38	63.35
LST [235]	7.4	10.8	<u>65.26</u>	48.42	45.03	<u>39.81</u>	64.39
R²-Tuning (Ours)	2.7	<u>10.6</u>	68.71	52.06	47.59	40.59	<u>64.32</u>

ers in Figure 3.8. Before calibration, the visual-text features are not well-aligned. Adding the video-level constraint makes higher-level features aligned with each other (b), while adding layer-wise constraint makes the model distill diverse information across different layers (c). The diagonal in (d) verifies that both calibration constraints can maximize the mutual similarities of paired visual and text features. Some quantitative results are presented in Table 3.7. Both video-level and layer-wise contrastive can help align the multi-granularity visual & query semantics, while their combination can further boost the performances. We also group the queries in QVHighlights into 5 bins with different lengths (acting as multi-granularities) and compute the means of MR mAP in Table 3.8. Although most training queries are coarse-grained (≤ 30 words), R²-Tuning can easily generalize to fine-grained queries (≥ 31 words), surpassing previous methods.

Comparison with Other Transfer Learning Methods Table 3.9 compares R²-Tuning with existing representative PETL (LoRA [88], VPT [101], VoP [92], and ST-Adapter [192]) and METL (E³VA [309], LoSA [93], LST [235]) methods on

Table 3.10: Performance comparison with different orders of attention layers. Self and Cross represent self-attention and cross-attention, respectively.

Order	MR@QVHighlights		HD@QVHighlights		MR@Ego4D-NLQ	
	R1@0.7	mAP	mAP	HIT@1	R1@0.7	mIoU
Self $\rightarrow$ Cross	49.42	45.21	39.10	63.35	2.01	4.75
Cross $\rightarrow$ Self	52.06	47.59	40.59	64.32	2.12	4.94

Table 3.11: Effectiveness justification of the recurrent design (parameter sharing across layers) in R^2 Block.

Shared	MR@QVHighlights		HD@QVHighlights		MR@Ego4D-NLQ	
	R1@0.7	mAP	mAP	HIT@1	R1@0.7	mIoU
	51.35	47.48	40.28	65.42	2.06	4.83
✓	52.06	47.59	40.59	64.32	2.12	4.94

QVHighlights. “w/o Tuning” means the entire CLIP is frozen and without any adapters, *i.e.*, only the temporal feature pyramid and prediction heads are trainable. The *rank* values for LoRA [88], E³VA [309], and LoSA [93] are set to 16. Our model consumes the least learnable parameters and GPU memory, while achieving the best MR and HD performances, demonstrating its parameter- and memory-efficiency.

Order of Attention Layers Table 3.10 compares different orders of attention layers. The standard transformer decoder [248] (Self $\rightarrow$ Cross) performs temporal modeling before aggregating query information, while the temporal modeling in our design (Cross $\rightarrow$ Self) can benefit from the guidance from query.

Effect of Recurrent Refinement Table 3.11 presents the justification of *recurrent* design of R^2 Block. We compare it with a baseline that use a unique R^2 Block for each layer, *i.e.*, parameters for different refinement steps are not shared. Even with much fewer parameters, adopting the *recurrent* design leads to slight performance gains on most tasks.

Table 3.12: Performance comparison with different numbers of transformer layers for temporal refinement.

#Layers	MR@QVHighlights		HD@QVHighlights		MR@Ego4D-NLQ	
	R1@0.7	mAP	mAP	HIT@1	R1@0.7	mIoU
1	52.06	47.59	40.59	64.32	2.12	4.94
2	52.14	47.62	40.63	64.28	2.15	4.97
3	52.71	47.68	40.80	64.35	2.21	5.03
4	52.32	47.81	40.85	63.30	2.19	4.91
5	52.13	47.53	40.78	64.41	2.10	4.88
6	51.16	47.42	40.61	64.33	2.04	4.82

Number of Temporal Transformer Layers We study how the number of temporal transformer layers in R^2 Block affects the model performance. The comparison is shown in Table 3.12. It can be observed that adding the number of transformer layers for temporal refinement only brings negligible performance gains. The best results appear when it has 3 or 4 layers. We set it to 1 for simplicity.

3.4.5 Visualization

We visualize some model predictions in Figure 3.9 and Figure 3.10. Compared with a strong baseline UniVTG [140], R^2 -Tuning can regress moment boundaries and detect highlights more accurately due to the novel design. We also show some failure cases in Figure 3.11. In the first case, when facing complex queries with multiple actions, our method focuses more on a single action (*i.e.*, *writing*). In the second case, the moment prediction shall be better than the ground truth.

3.5 Summary

In this chapter, we further study the problem of video temporal grounding from a parameter- and memory-efficient transfer learning perspective. Several task-specific



Figure 3.9: Visualization of joint moment retrieval and highlight detection results on QVHighlights val split. We compare our method with UniVTG [140], which is a strong baseline leveraging both additional temporal backbone and large-scale post pre-training. R²-Tuning can accurately regress the boundaries of moments and predict highlight saliency scores through its novel feature refinement design.

modules such as temporal feature pyramid are also presented to enhance video temporal grounding performance. Experiments across three video temporal grounding tasks on six public benchmarks demonstrate the significance and effectiveness of the proposed scheme. We summarize our contributions below:

1. A comprehensive analysis and comparison of existing image-to-video transfer learning approaches. We identified their limitations in spatial-temporal modeling capability, parameter and memory efficiency, and granularity flexibility.
2. A parameter- and memory-efficient transfer learning framework (R²-Tuning) specifically designed for the fine-grained video temporal grounding tasks. This framework, for the first time, remove the need of an auxiliary temporal backbone while achieving significantly better performance.
3. A lightweight reversed recurrent block (R² Block) for query-modulated spatial

Chapter 3. Efficient Image-to-Video Transfer Learning for Temporal Grounding

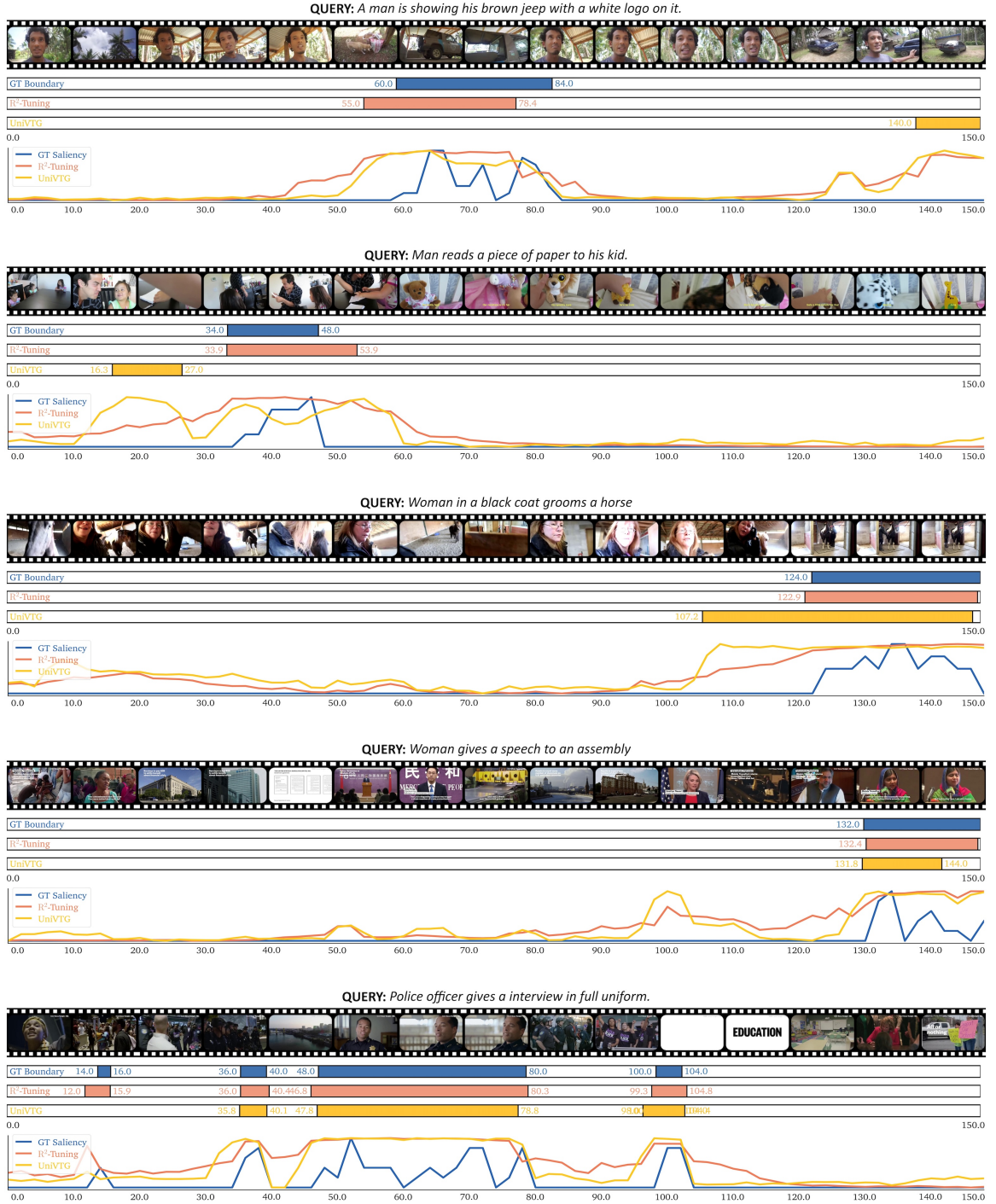


Figure 3.10: More visualizations of joint moment retrieval and highlight detection results on QVHighlights val split. R²-Tuning can accurately regress the boundaries of moments and predict highlight saliency scores.

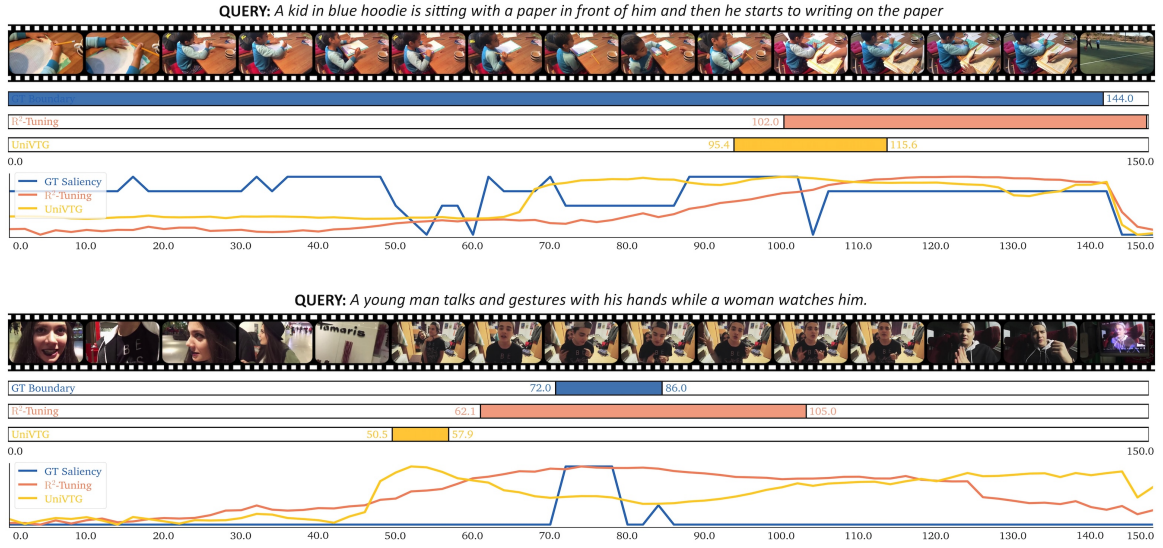


Figure 3.11: Failure cases of predictions from R^2 -Tuning on QVHighlights val split.

pooling and recurrent temporal refinement.

4. A temporal feature pyramid architecture that can ensure accurate boundary regression for moments with lengths. This has been a standard module in most video temporal grounding models.

We also list our novelties behind each successful design:

1. In the proposed R^2 Block, we adopt a novel query-modulated spatial pooling strategy that adaptively compresses high-resolution spatial features into a single compact token conditioning on the input query. Such a mechanism ensures the model’s robustness to diverse user queries.
2. During model inference, a recurrent temporal refinement scheme is designed to effectively aggregate multi-layer information from CLIP encoders. This design produces high-quality features for diverse downstream tasks.
3. In prediction heads, boundary regression and foreground-background classification are jointly conducted in parallel, enabling multi-scale prediction while preserving computational efficiency.

3.6 Limitations & Future Work

We acknowledge that while the proposed method achieves significant performance gains across various video temporal grounding scenarios, its scalability and generalizability remain limited due to the relatively small number of trainable parameters (2.7M only). We believe this limitation can be effectively mitigated by employing more powerful backbones (*e.g.*, LLMs) for video/text encoding and incorporating a large-scale pre-training stage to better facilitate grounding capability learning. In the next chapter, we present our attempt along this direction and our solution for integrating video temporal grounding with open-ended question answering.

Part II

Reasoning

Thoughts without content are empty, intuitions without concepts are blind.

— Immanuel Kant, *Kritik der reinen Vernunft*

Chapter 4

Event-Level & Time-Sensitive Video-Language Understanding

Recent advances in Video Large Language Models (Video-LLMs) have demonstrated their great potential in general-purpose video understanding. To verify the significance of these models, a number of benchmarks have been proposed to diagnose their capabilities in different scenarios. However, existing benchmarks merely evaluate models through video-level question answering, lacking fine-grained event-level assessment and task diversity. To fill this gap, we introduce **E.T. Bench** (Event-Level & Time-Sensitive Video Understanding **Benchmark**), a large-scale and high-quality benchmark for open-ended event-level video understanding. Categorized within a 3-level task taxonomy, E.T. Bench encompasses 7.3K samples under 12 tasks with 7K videos (251.4h total length) under 8 domains, providing comprehensive evaluations. We extensively evaluated 8 Image-LLMs and 12 Video-LLMs on our benchmark, and the results reveal that state-of-the-art models for coarse-level (video-level) understanding struggle to solve our fine-grained tasks, *e.g.*, grounding event-of-interests within videos, largely due to the short video context length, improper time representations, and lack of multi-event training data. Focusing on

these issues, we further propose a strong baseline model, **E.T. Chat**, together with an instruction-tuning dataset **E.T. Instruct 164K** tailored for fine-grained event-level understanding. Our simple but effective solution demonstrates superior performance in multiple scenarios.

4.1 Introduction

The recent advent of Multi-modal Large Language Models (MLLMs) [2, 213, 148, 349, 171, 41] has led to a substantial paradigm shift in visual-language understanding, moving away from designing task-specific models and collecting domain-specific data, towards developing general-purpose task-solvers for open-ended scenarios. By integrating LLMs with visual encoders, these models jointly benefit from perception abilities and powerful reasoning skills, showing remarkable performance in even unseen applications, demonstrating great potential for such scheme.

To effectively evaluate the capabilities of these models, a number of benchmarks [320, 123, 160, 32, 310, 126, 184, 130, 161] have been introduced to study their feasibilities in different scenarios. Yet, most of the benchmarks are focusing on image or short (seconds-long) video understanding, which require strong static scene understanding abilities but overlooking the fine-grained temporal information. Some recent works [228, 173] tend to evaluate MLLMs on longer videos, but they still leverage multiple-choice question answering (MCQ) as their main task, lacking flexibilities for open-ended tasks. Nevertheless, none of the existing benchmarks are designed for multi-event or time-sensitive scenarios, thus they suffer from severe single-frame biases, as can be seen in the comparable performances between Image- and Video-LLMs in these benchmarks [123, 126].

To address these issues and better understand the open-ended capabilities of these models, we propose **E.T. Bench**, a comprehensive benchmark for event-level and

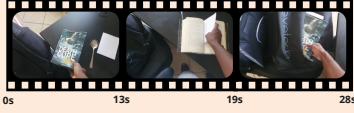
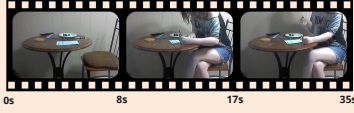
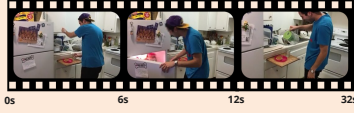
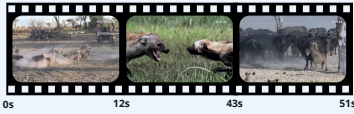
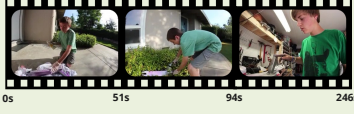
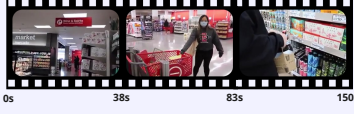
[RAR] Referred Action Recognition What type of action is the camera wearer performing at around 23s?  0s 13s 19s 28s (A) Taking something out of something (B) Opening something (C) Putting something on top of something (D) Closing something Correct Answer: (B)	[ECA] Event-Caption Alignment Which segment can be best described by "person sit by the table while eating chocolate"?  0s 8s 17s 35s (A) From 1.2 to 7.6 seconds (B) From 10.2 to 15.3 seconds (C) From 18.6 to 24.1 seconds (D) From 12.5 to 28.4 seconds Correct Answer: (C)	[RVQ] Referred Video Question-Answering Watch the video at 1.4s - 5.2s and answer "Which object was taken by the person?"  0s 6s 12s 32s (A) The dish (B) The kettle (C) The plate (D) The refrigerator (E) Unable to answer Correct Answer: (E)
[TVG] Temporal Video Grounding Localize the event about "buffalo is running".  0s 12s 43s 51s Correct Answer: The event happens in 44.1 - 50.1 seconds.	[EPM] Episodic Memory Where did I find the blue T-shirt in this shop?  0s 196s 425s 480s Correct Answer: The place can be viewed in 330.7 - 350.2 seconds.	[TAL] Temporal Action Localization Detect all the moments containing "cricket shot".  0s 56s 87s 110s Correct Answer: It happens in 27.5 - 29.2 and 48.1 - 50.3 seconds.
[EVS] Extractive Video Summarization Summarize this video to 15% of the total length.  0s 12s 43s 156s Correct Answer: The video summary locates in 4 - 12, 28 - 32, 56 - 58, 64 - 66, and 88 - 100 seconds.	[VHD] Video Highlight Detection Identify the highlights of "girl making chocolate juice".  0s 23s 98s 150s Correct Answer: The highlights of the given query happen at 18, 76, 84, and 105 seconds.	[DVC] Dense Video Captioning Densely describe all the events in the video.  0s 112s 248s 360s Correct Answer: 0 - 56 seconds, a woman is talking to the camera, 58 - 94 seconds, she introduce the ingredients, 105 - 170 seconds...
[SLC] Step Localization and Captioning Identify and summarize all the key steps in the video.  0s 51s 94s 246s Correct Answer: 0 - 4 seconds, shake and spray it. 42 - 68 seconds, rub it around. 78 - 90 seconds, wash with water. 98 - 112 seconds...	[TEM] Temporal Event Matching Find a similar moment to the one in 56 - 64 seconds.  0s 38s 83s 150s Correct Answer: A similar event (a person with mask pushing a shopping cart) can be found in 18 - 30 seconds.	[GVQ] Grounded Video Question-Answering How many screws did I pick just now?  0s 184s 309s 480s (A) Five (B) Two (C) Three (D) One Correct Answer: (C) The relevant event can be found in 235.5 - 241.6 seconds.

Figure 4.1: Task definitions in E.T. Bench. The 12 tasks derives from 4 essential capabilities for time-sensitive video understanding: *referring*, *grounding*, *dense captioning*, and *complex understanding*.

time-sensitive video understanding. As shown in Figure 4.1 and compared in Table 4.1, our benchmark significantly diverse from previous ones that it focus on time-sensitive understanding on long and multi-event videos. Our motivation is that a well-performing Video-LLM should possess the capability to precisely refer to and localize any events that align with user interests. Based on this assumption, we build our task taxonomy by summarizing four essential capabilities required for time-sensitive video understanding: *referring*, *grounding*, *dense captioning*, and

complex understanding. Then, each capability is delineated with carefully designed tasks. The diversity of scenarios is ensured by meticulously collecting videos from 15 datasets covering 8 domains. A comprehensive data cleaning, annotation repurposing, instruction design, manual verification, and sampling pipeline is leveraged to generate 7.8K high-quality annotations.

We extensively evaluate 20 models, including 7 open-source Image-LLMs, 9 open-source Video-LLMs, and 4 proprietary MLLMs, on E.T. Bench. The results reveal that state-of-the-art models on existing VideoQA benchmarks [123, 184, 126] struggle on our E.T. Bench, especially on *grounding*, *dense captioning*, and *complex understanding* tasks. We attribute it to two key limitations of existing development pipelines for MLLMs. First, the discrete next-token prediction paradigm has natural drawbacks in numerical calculations [78, 100], limiting timestamp understanding and generation. Second, most existing video instruction-tuning datasets [125, 171, 127] predominantly comprise short videos with coarse-level annotations, bringing significant gap between training and real-world applications. To tackle these problems, we propose **E.T. Chat**, a novel time-sensitive Video-LLM that reformulates timestamp prediction as an embedding matching problem, serving as a strong baseline on E.T. Bench. As for data, we also curate **E.T. Instruct 164K**, an instruction-tuning dataset tailored for multi-event and time-sensitive scenarios. Extensive comparisons demonstrate the effectiveness of the proposed scheme.

4.2 Related Work

4.2.1 Video Large Language Models.

Video-LLMs [171, 135, 131, 328, 228, 216, 90] represent a class of intelligent chatbots capable of understanding videos and perform various open-ended tasks. Generally, a Video-LLM comprises a visual encoder [205, 322] for perception, a projec-

tor [95, 124] for feature alignment, and an LLM [245, 245, 43, 23] for reasoning and response generation. VideoChat [125] and Video-ChatGPT [171] are two earliest attempts in this direction. Following works tend to provide better solutions via adding audio modality [328], joint training on images and videos [135, 105], or performing alignment before projection [135]. A recent trend [89, 216, 90, 200] involves the integration of Video-LLMs with time-sensitive understanding capabilities, while their solutions remain sub-optimal. Therefore, we propose to reframe timestamp generation as an embedding matching problem.

4.2.2 Benchmarks for Video-LLMs

The increasing number of Video-LLMs motivate the development of benchmarks [123, 37, 184, 173, 126, 161]. Among the earliest is SEED-Bench [123], an MLLM benchmark that supports both Image-LLMs and Video-LLMs and offers three evaluation dimensions in the realm of temporal modeling. AutoEval-Video [37] and Video-Bench [184] are designed specifically for videos. They employ LLMs for QA generation or model evaluation. MVBench [126] provides a novel scheme to repurpose existing datasets for Video-LLM evaluation. Recent studies also expand their scope and consider the ability of understanding extremely long videos [173, 228] or comprehending fine-grained temporal order information [161]. Nevertheless, none of the benchmarks have been designed for multi-event and time-sensitive understanding. In response to this gap, we introduce E.T. Bench, the first benchmark providing comprehensive evaluations on these scenarios.

4.3 E.T. Bench

In this section, we illustrate the detailed pipeline employed to develop our E.T. Bench. As shown in Figure 4.2 (right), the pipeline begins with the definition of

Table 4.1: Comparison between E.T. Bench and existing Video-LLM benchmarks.

Benchmark	Domain	Annotator	#Tasks	#Samples	#Videos	Avg./Max. Duration	Long Video	Event Level	Time Sensitive	Answer Type	Evaluation Method
SEED-Bench [123]	Action	LLM	3	3,757	3,757	8 frames	✗	✗	✗	MCQ	Likelihood
EgoSchema [173]	Egocentric	LLM	1	5,031	5,031	180s / 180s	✓	✗	✗	MCQ	Likelihood
AutoEval-Video [37]	Open	Human	9	327	327	15s / 101s	✗	✗	✗	Open	GPT-4
Video-Bench [184]	Open	LLM	1	17,054	5,917	56s / 3,599s	✓	✗	✗	MCQ	Mixed‡
TempCompass [161]	Open	LLM	4	7,540	410	12s / 35s	✗	✗	✗	MCQ/Open	Rule/GPT
MVBench [126]	Open	Human†	1	4,000	3,673	15s / 116s	✗	✗	✗	MCQ	Rule
E.T. Bench (Ours)	Open	Human†	12	7,289	7,002	129s / 795s	✓	✓	✓	MCQ/Open	Rule

† Repurposed from existing datasets ‡ Including next-token likelihood, T5 sentence similarity, and GPT-3.5 assisted evaluation

four essential capabilities for event-level and time-sensitive video understanding, *i.e.*, *referring*, *grounding*, *dense captioning*, and *complex understanding*, arranged in increasing order of difficulty. For each capability, we design a series of tasks for effective assessment. For each task, we meticulously select existing datasets with timestamp annotations provided by human annotators, and rewrite them into instruction-following formats according to task formulation, ensuring high quality and verisimilitude. The diversity of E.T. Bench is guaranteed by carefully choosing variable-length videos from different domains. Finally, a thorough manual check, filtering, and sampling process is conducted to eliminate unsatisfactory samples. The detailed processes for each step are introduced as follows.

4.3.1 Hierarchical Task Taxonomy

To evaluate the open-ended Video-LLMs from various perspectives, we design a three-level task taxonomy depicted in Figure 4.2 (left). Definitions for capabilities and tasks are as follows.

Referring means the ability to comprehend time information from user inputs. For example, given a question “*What is the person doing from 23s to 35s?*”, the model has to understand which part of the video is the user referring to, and provide response with more consideration on that segment. For better quantifying the model perfor-

mances, we formulate all the *referring* tasks as multiple-choice question-answerings (MCQs), including (1) [RAR] *Referred Action Recognition*: Identify the action given a coarse timestamp hint (e.g., “around 12s”). The model has to determine the actual reference according to both the video and options. (2) [ECA] *Event-Caption Alignment*: Select the correct temporal boundary for a given caption. The model need to understand and distinguish multiple timestamps in the options. (3) [RVQ] *Referred Video Question Answering*: Answer the question conditioning on a given segment. Each question is supplied with four candidates and an “unable to answer” option, denoting the case when it cannot be answered with given the segment.

Grounding indicates the ability to localize event- or moment-of-interests with accurate timestamps. It diverse from previous works [126, 266] that only consider coarse-level grounding, e.g., “at the beginning/middle/end of the video”. Outputs for *grounding* tasks are open-ended, processed by rule-based parsers and evaluated with continuous metrics. The definitions of tasks are (1) [TVG] *Temporal Video Grounding*: Determine the temporal boundary of a single event according to the text description. (2) [EPM] *Episodic Memory*: Localize the event that can answer the given question in egocentric scenarios, e.g., “Where is my backpack?”. (3) [TAL] *Temporal Action Localization*: Detect and localize a series of segments containing the given action, e.g., finding all “golf swing” segments in a long video. (4) [EVS] *Extractive Video Summarization*: Provide a list of segments that can be merged to form a compact video summary (with around 15% of the total duration). (5) [VHD] *Video Highlight Detection*: Cherry-pick a single timestamp (e.g., “15s”) that can best reflect the highlight moment corresponding to a query. Note that the formulations of [TAL], [EVS], and [VHD] have been modified (compared with previous works [223, 230, 120]) to fit the nature of LLMs.

Dense Captioning is a more complicated ability that requires jointly localize key-events and generate descriptions/summaries for each segment. This is more practical for storytelling or key-information extraction from long videos in comparison

with video-level captioning on trimmed clips [290]. We define two *dense captioning* tasks according to different goals: (1) [DVC] *Dense Video Captioning*: Comprehensively describe all the events happened in the video. This is a general case with the goal of covering as much events as possible. (2) [SLC] *Step Localization and Captioning*: Identify and describe only the key-steps in instructional videos. In this case, the segments are shorter & disjoint and the step descriptions are more precise compared with [DVC].

Complex Understanding refers to the versatile integration of the aforementioned time comprehension and event localization, requiring the model to demonstrate proficient event-level and time-sensitive understanding. The two tasks are: (1) [TEM] *Temporal Event Matching*: Find and locate a similar event in the same video conditioning on the given segment. This involves a two-stage reasoning process that first identify the event in the given timestamps, then localize another segment with the most similar content. (2) [GVQ] *Grounded Video Question Answering*: Answer the given multiple-choice question by selecting an option and localizing a segment that supports the answer. This is also a complex scenario requiring both understanding and localization abilities. Validating the localization results can help diagnose the reasoning process of Video-LLMs.

4.3.2 Data Collection and Annotation

The key challenge of data collection is how to obtain videos with precise temporal boundary annotations. Previous works either prompt LLMs with frame-level information extracted from collections of experts [265, 200, 266, 90] or transform human-annotated moment tags (e.g., 5.2s) to boundaries (e.g., 3.2s to 7.2s) using pre-defined rules [139, 206]. These solutions can only generate temporal boundaries with substantial noise, which are not suitable for accurate model evaluation. Therefore, we meticulously curate multi-event videos from existing datasets with

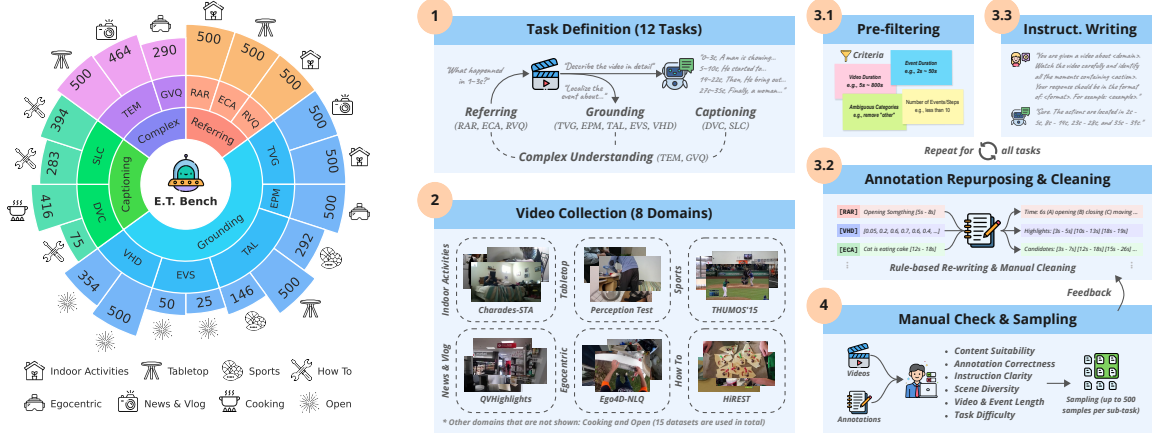


Figure 4.2: **Left:** Task taxonomy and sample distribution. **Right:** Generation pipeline for E.T. Bench. We conduct a thorough process of pre-filtering, annotation repurposing, instruction writing, manual check, and sampling to obtain high-quality fine-grained annotations. Details are discussed in Section 4.3.

high-quality human-annotated timestamps, and repurpose the annotations by transforming them according to our task formulations. To ensure the diversity of scenarios, we carefully select 15 datasets from 8 domains, *i.e.*, *indoor activities*, *tabletop*, *sports*, *egocentric*, *cooking*, *news & vlogs*, *how-to*, and *open*. All the videos are collected from val or test splits to prevent potential data leakage. For annotation generation, we develop for each task a thorough process containing pre-filtering, manual annotation repurposing, and instruction template design, presented in Figure 4.2 (right).

Dataset Pre-filtering Table 4.2 presents the pre-filtering criteria for each source dataset when generating E.T. Bench. We design the criteria according each task and source dataset, leveraging metrics such as *video/moment duration*, *video domain*, and *number of steps per video* to select high-quality samples for benchmarking.

Annotation Repurposing and Cleaning We also manually design principles to repurpose the original dataset annotations and clean the unsuitable samples. The

Table 4.2: Pre-filtering criteria for E.T. Bench generation. N/A means no filtering.

Type	Task	Source	Domain(s)	Selection Criteria
Referring	[RAR] Referred Action Recognition	Perception Test [194]	• Tabletop • Indoor Activities • Egocentric	• Video $\notin$ [TAL] $\cup$ [TEM] • Subset = "Action Localization" • Video Duration $\in$ [20s, 600s] • #Actions per Video ≥ 3 • Action Class $\neq$ "other"
	[ECA] Event-Caption Alignment	Charades-STA [70]	• Indoor Activities	• Video $\notin$ [TVG] • Event Duration $\in$ [2s, 30s]
	[RVQ] Referred Video Question Answering	STAR [273]	• Indoor Activities	• QType $\in$ {Interaction, Sequence}
Grounding	[TVG] Temporal Video Grounding	QVHighlights [120]	• News & Vlogs	• Video $\notin$ [VHD] $\cup$ [TEM] • Moment Duration $\in$ [2s, 50s] • #Segments per Query = 1
		Charades-STA [70]	• Indoor Activities	• Video $\notin$ [ECA] • Moment Duration $\in$ [2s, 50s]
	[EPM] Episodic Memory	Ego4D-NLQ [76]	• Egocentric	• Video Duration ≤ 600s • Moment Duration $\in$ [3s, 50s]
	[TAL] Temporal Action Localization	Perception Test [194]	• Tabletop • Indoor Activities • Egocentric	• Video $\notin$ [RAR] $\cup$ [TEM] • Subset = "Action Localization" • Video Duration $\in$ [20s, 600s] • #Segments per Action ≤ 10 • Action Class $\notin$ {moving object(s) around, other}
		THUMOS'14 [104]	• Sports	• Video Duration ≤ 600s • #Segments per Action ≤ 10 • Action Class $\neq$ "ambiguous"
		THUMOS'15 [75]	• Sports	• Video Duration ≤ 600s • #Segments per Action ≤ 10 • Action Class $\neq$ "ambiguous"
	[EVS] Extractive Video Summarization	TVSum [230]	• Open	• Summary Ratio $\in$ [0.1, 0.25]
		SumMe [81]	• Open	• Summary Ratio $\in$ [0.1, 0.25]
	[VHD] Video Highlight Detection	QVHighlights [120]	• News & Vlogs	• Video $\notin$ [TVG] $\cup$ [TEM] • #Highlights per Query ≤ 2
		YouTube Highlights [234]	• Open	• Highlight Ratio $\in$ [0.05, 0.9]
Dense Captioning	[DVC] Dense Video Captioning	HiREST [321]	• How To • Indoor Activities • Cooking	• Query is Relevant to the Video • Video is Clippable
		YouCook2 [346]	• Cooking	• N/A
	[SLC] Step Localization and Captioning	CrossTask [351]	• How To • Cooking	• Step Duration ≥ 2s • Repeated/Unordered Steps $\in \emptyset$
		HT-Step [3]	• How To • Indoor Activities • Cooking	• Video Duration $\in$ [10s, 600s] • Step Duration ≥ 2s • #Steps per Video ≥ 2 • Repeated/Unordered Steps $\in \emptyset$
Complex Understanding	[TEM] Temporal Event Matching	Perception Test [194]	• Tabletop • Indoor Activities • Egocentric	• Video $\notin$ [RAR] $\cup$ [TAL] • Subset = "Action Localization" • Video Duration $\in$ [20s, 600s] • Action Duration $\in$ [2s, 50s] • #Actions per Video ≥ 2 • Action Class $\neq$ "other"
		QVHighlights [120]	• News & Vlogs	• Video $\notin$ [TVG] $\cup$ [VHD] • Segment Duration $\in$ [2s, 50s] • #Segments per Query ≥ 2
	[GVQ] Grounded Video Question Answering	QAEgo4D [20]	• Egocentric	• Segment Duration $\in$ [2s, 50s]

detailed repurposing and cleaning steps for each task are summarized as follows.

[RAR] **Referred Action Recognition** We adopt the *action localization* subset of Perception Test [194] for this task. We first select videos with at least three different actions, in which one action is sampled as ground truth. We then sample two other actions from the same video as intra-video distracters, and one action from other videos as inter-video distracter. The coarse timestamp hint is sampled from the segment containing only the ground truth action.

[ECA] **Event Caption Alignment** We utilize Charades-STA [70] as data source. For each event-query pair, we randomly generate distracters (temporal boundaries) with $0.5\times$ to $2\times$ lengths compared with the ground truth, and ensure the temporal IoUs between any two options are no more than 0.5.

[RVQ] **Referred Video Question Answering** We leverage the high quality QA pairs from *interaction* and *sequence* question types of STAR [273]. Since the original annotations only contain question-relevant temporal boundaries, we randomly pick 20% of the QA pairs and modify their boundaries to have no overlap with the original ones, in order to synthesis the case when the question cannot be answered within the given boundary. An extra “unable to answer” option is added to all QA pairs to verify the robustness.

[TVG] **Temporal Video Grounding** Two datasets, *i.e.*, Charades-STA [70] and QVHighlights [120], are chosen for this task. We filter out the samples with event duration shorter than 2s or longer than 50s, as these are generally noisy samples. On QVHighlights, only the samples with single events are selected to align with our problem formulation.

[EPM] **Episodic Memory** We employ Ego4D-NLQ [76] and conduct a thorough data cleaning & verification process. First, we perform a rule-based fix for noisy questions. Some common cases are: (1) Question starts with an additional “Query Text:” string. (2) Typos such as “I” $\rightarrow$ “i” or “l”. (3) Unclear references such as

“person x”. We also found that some questions are ambiguous in the context of long videos, so we randomly crop the all the videos to 300-second long.

[TAL] **Temporal Action Localization** We adopt Perception Test [194], THUMOS'14 [104] (test split), and THUMOS'15 [75] (val split). Videos with ambiguous action classes, *e.g.*, *moving object(s) around* and *other* in Perception Test, and *ambiguous* in THUMOS, are discarded. To reduce the difficulty, samples with more than 10 ground truth moments are filtered out as well.

[EVS] **Extractive Video Summarization** We repurpose TVSum [230] and SumMe [81] to generate samples. In conventional video summarization, each frame is annotated with a probability of being the summary, which is incompatible with Video-LLMs that cannot strictly produce temporally-aligned frame-level scores, as can be seen in the near-random results in [216, 200]. Therefore, we reformulate it to predicting a set of temporal boundaries that compose the summary. Ground truths are obtained by sorting the frame-level scores and generate boundaries for consecutive frames with top-15% scores. When a video is annotated by multiple annotators (*i.e.*, in TVSum), we simply average the scores. This reformulation helps models persistently generate reasonable results.

[VHD] **Video Highlight Detection** Samples are generated from QVHighlights [120] and YouTube Highlights [234] using a similar method as [EVS], but we only consider the frames with the highest scores. That means, ground truths are the frames with the highest saliencies. During inference, a prediction is considered correct if the timestamp falls into any of the temporal boundaries. We directly utilize the text queries in QVHighlights as highlight queries. For YouTube Highlights, the domains are used, and videos with highlights covering more than 90% are discarded.

[DVC] **Dense Video Captioning** We utilize YouCook2 [346] and HiREST [321]. Although HiREST only contains instructional videos, it is still considered as [DVC]

rather than [SLC] because of the large event coverage. We also trimmed out the opening and closing scenes of HiREST videos.

[SLC] **Step Localization and Captioning** We select CrossTask [351] and HT-Step [3] for this task. For CrossTask, we filter out the samples with wrong annotations, e.g., repeated steps, and select only the videos with all steps longer than 2s to avoid ambiguous annotations. For HT-Step, we keep only the videos with at least 2 steps and remove the samples with incorrect temporal orders.

[TEM] **Temporal Event Matching** We repurpose Perception Test [194] and QVHighlights [120] for this novel task, where the former focus on actions and the latter is for general events. We select videos with actions (excluding the *other* category) occurs multiple times from Perception Test, and sample one segment as the input reference. Other boundaries are used as ground truths. For QVHighlights, samples with one query referring to multiple disjoint moments are used.

[GVQ] **Grounded Video Question Answering** We adopt QAEgo4D [20] which naturally contains both QA pairs and corresponding timestamps derived from Ego4D-NLQ [76]. To control the task difficulty, we randomly crop all the videos to 150-second long. Typos in QA pairs are fixed.

Instruction Templates We design several instruction templates for each task, which are shown in Table 4.3. To ensure the models could give responses in desired formats, each instruction starts with a sentence introducing the domain/title of the video, followed by detailed requirements about the task. We also add an explicit statement about the format of response and an example as guidance. The model outputs are passed through rule-based parsers for answer extraction.

Manual Verification and Sampling After annotation generation, we conduct a careful manual review on the samples, focusing on content suitability, annotation correctness, instruction clarity, scene diversity, video & event length, and task dif-

Table 4.3: Instruction templates for E.T. Bench. **Green** text indicates the candidate expressions for difference domains. **Blue** text means the placeholder to be filled according to each sample. **<time>** denotes the timestamp representation.

Type	Task	Instruction Template	Example Response
Referring	[RAR]	You are given a video containing a series of actions. Watch the video carefully and identify the action around <time> by choosing from a set of options. The format of your response should be: "Best Option: (your choice)". For example: "Best Option: (B)". Now I give you the options: (A) <option> (B) <option> (C) <option> (D) <option> . Please provide your choice.	Best Option: (A)
	[ECA]	You are given a video about indoor activities. Watch the video carefully and select the moment that can be best described by the sentence " <query> ". The format of your response should be: "Best Option: (your choice)". For example: "Best Option: (A)". Now I give you the options: (A) <time> - <time> (B) <time> - <time> (C) <time> - <time> (D) <time> - <time> . Please provide your choice.	Best Option: (C)
	[RVQ]	You are given a video about indoor activities. Watch the video carefully and answer a multiple choice question solely based on the event in <time> - <time> . The format of your response should be: "Best Option: (your choice)". For example: "Best Option: (C)". You may select "unable to answer" if the question can not be answered based on the provided moment. Now I give you the question: " <question> ". The options are (A) <option> (B) <option> (C) <option> (D) <option> (E) <option> . Please provide your choice.	Best Option: (D)
Grounding	[TVG]	You are given a video about daily activities / indoor activities . Watch the video carefully and find a visual event described by the sentence: " <query> ". The format of your response should be: "The event happens in <start time> - <end time> ". You must represent start and end times in seconds. For example: "The event happens in 10.2 - 12.8 seconds".	The event happens in <time> - <time> .
	[EPM]	You are given an egocentric video about daily activities. Watch the video carefully and find a visual event that can answer the question: " <question> ". The format of your response should be: "The event happens in <start time> - <end time> ". You must represent start and end times in seconds. For example: "The event happens in 10.2 - 12.8 seconds".	The event happens in <time> - <time> .
	[TAL]	You are given a video containing a series of actions. Watch the video carefully and find all the visual events belonging to the action category: " <action> ". The format of your response should be: "The action happens in <start time> - <end time> , <start time> - <end time> , and <start time> - <end time> ". You must represent start and end times in seconds. For example: "The action happens in 4.2 - 6.8, 7.5 - 10.3, 15.1 - 18.6, and 23.4 - 27.5 seconds".	The action happens in <time> - <time> and <time> - <time> .
	[EVS]	You are given a video about <domain> . Watch the video carefully and summarize it into multiple short segments. The total length of the segments should be about 15% of the original video. The format of your response should be: "The summary locates in <start time> - <end time> , <start time> - <end time> , and <start time> - <end time> ". You must represent start and end times in seconds. For example: "The summary locates in 5.2 - 7.5, 9.4 - 12.3, 16.9 - 18.2, and 21.8 - 25.4 seconds".	The summary locates in <time> - <time> and <time> - <time> .
	[VHD]	You are given a video about daily activities. Watch the video carefully and find a highlight moment according to the sentence / its domain : " <query> ". The format of your response should be: "The highlight moment happens at <time> ". You must represent time in seconds. For example: "The highlight moment happens at 26.8 seconds".	The highlight moment happens at <time> .
Dense Captioning	[DVC]	You are given a video about " <query> ". Watch the video carefully and densely describe all the events in it. For each event, you need to determine the start and ends times and provide a concise description. The format of your response should be: " <start time> - <end time> , <description> ". For example: "90 - 102 seconds, spread margarine on two slices of white bread. 114 - 127 seconds, place a slice of cheese on the bread".	<time> - <time> , place bulgur wheat in a bowl and add boiling water. <time> - <time> , finely chop a bundle of parsley and add to a bowl. <time> - <time> , remove the leaves from stalks of mint chop finely and add to the parsley.
	[SLC]	You are given a video about " <task> ". Watch the video carefully and identify all the key steps in it. For each step, you need to determine the start and ends times and provide a concise description using a few words. The format of your response should be: " <start time> - <end time> , <description> ". You must represent start and end times in seconds. For example: "24.8 - 30.2 seconds, cut apple. 35.6 - 40.4 seconds, wash dishes".	<time> - <time> , add sugar. <time> - <time> , pour water. <time> - <time> , cut lemon. <time> - <time> , squeeze lemon. <time> - <time> , pour lemon juice. <time> - <time> , stir mixture.
Complex Understanding	[TEM]	You are given a video about daily activities / containing a series of actions . Watch the video carefully and identify the event in <time> - <time> , then localize a different moment that contains the most similar event. The format of your response should be: "The similar event happens in <start time> - <end time> ". You must represent start and end times in seconds. For example: "The similar event happens in 16.8 - 20.4 seconds".	The similar event happens in <time> - <time> .
	[GVQ]	You are given an egocentric video about daily activities. Watch the video carefully and answer a multiple choice question. Your answer should contain a choice of the best option and a relevant moment that supports your answer. The format of your response should be: "Best Option: (your choice). The relevant event happens in <start time> - <end time> ". Now I give you the question: " <question> ". The options are (A) <option> (B) <option> (C) <option> (D) <option> . Please provide your choice and the relevant moment.	Best Option: (C). The relevant event happens in <time> - <time> .

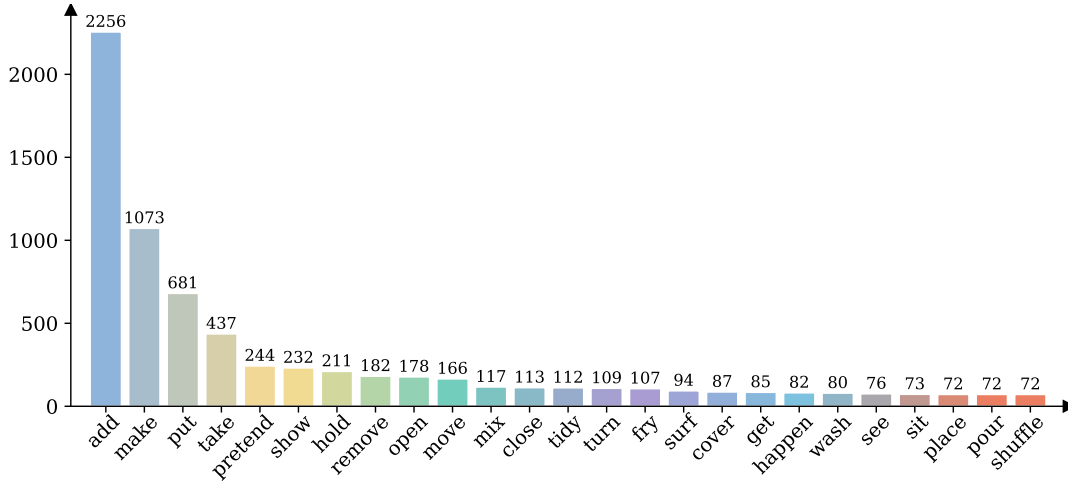


Figure 4.4: Frequency distribution of verbs (top 25 / 461) in E.T. Bench. The x- and y-axes denote the verbs and their frequencies, respectively.

proportion of samples are in the *grounding* category, as we emphasize the moment localization ability of modern Video-LLMs.

Text Queries Figure 4.3 (left) shows the word cloud of text queries in E.T. Bench. Thanks to the wide range of video domains, the queries are also diverse in terms of both nouns and verbs. Most queries are human-centric, describing human activities or human-object interactions. We visualize the frequency distribution of verbs and nouns in E.T. Bench in Figure 4.4 and Figure 4.5, respectively. The distribution histograms demonstrate the diversity of queries in E.T. Bench.

Video Durations Figure 4.3 (right) shows the distribution of averaged video durations across tasks. Our videos have a wide spectrum of durations, where *referring* tasks have relatively shorter videos, and *grounding* and *captioning* have longer ones. Our experimental results in Table 4.9 show that the duration of videos have significant influence on model performance.

Distribution of Queries We visualize the frequency distribution (top 25) of verbs and nouns in E.T. Bench in Figure 4.4 and Figure 4.5, respectively. The verbs and

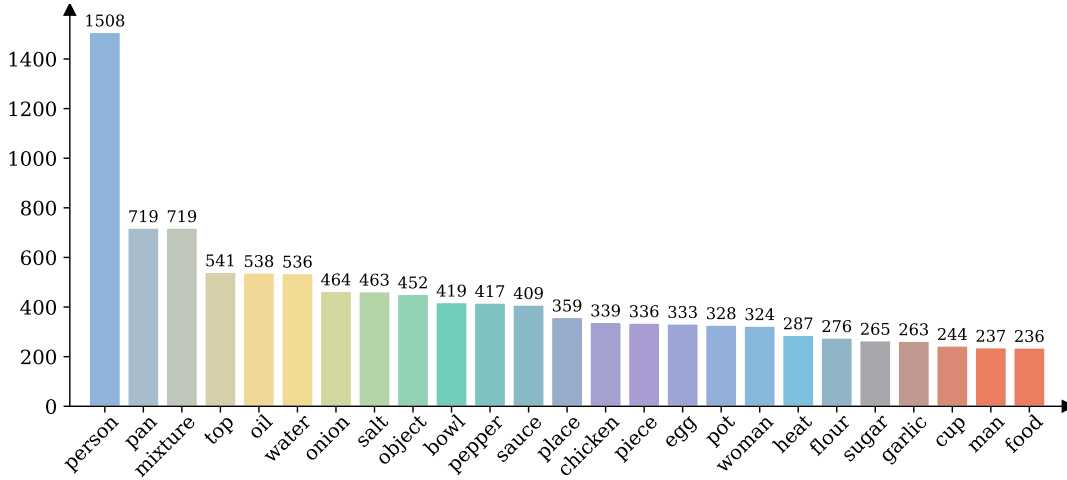


Figure 4.5: Frequency distribution of nouns (top 25 / 3,090) in E.T. Bench. The x- and y-axes denote the nouns and their frequencies, respectively.

nouns cover multiple human-centered daily activities. The distribution histograms demonstrate the diversity of queries in E.T. Bench.

4.4 Our Method

Extensive evaluations (in Table 4.9) reveal that even the state-of-the-art Video-LLMs cannot perform well on E.T. Bench, especially on the more complicated *grounding*, *dense captioning*, and *complex understanding* tasks. We attribute this phenomenon to two key limitations of existing development pipelines for Video-LLMs: (1) **Model**: Existing models fall short in numerical modeling [58, 64], which are essential capabilities for arithmetic calculations – timestamps processing in our case; (2) **Data**: Both pre-training and instruction-tuning are conducted on short & single-event videos, leading to weak general understanding abilities for multi-event videos. To address these limitations, we propose **E.T. Chat**, a novel Video-LLM that reformulates timestamp prediction as an embedding matching problem, serving as a strong baseline on E.T. Bench. we also curate **E.T. Instruct 164K**, an instruction-tuning

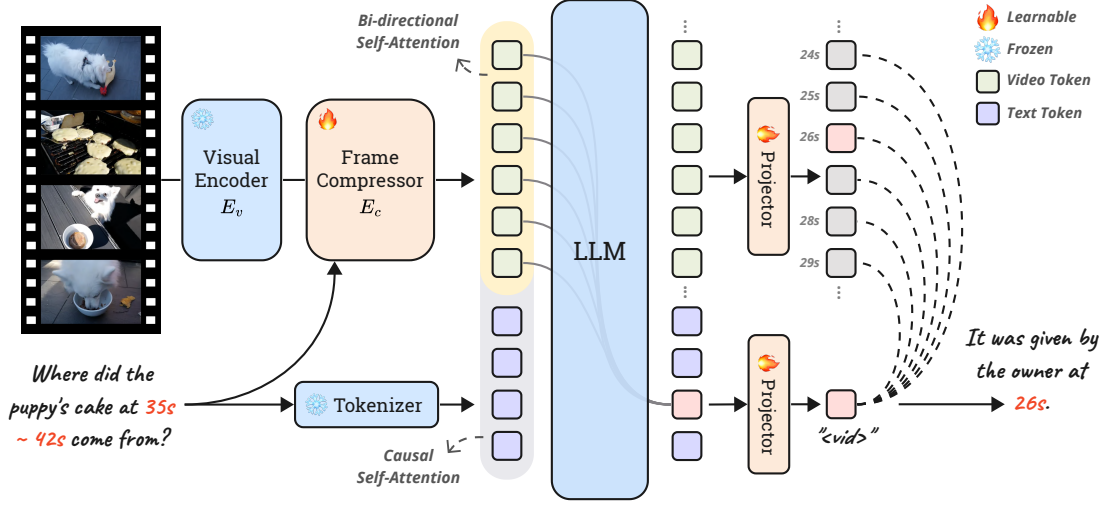


Figure 4.6: Overall architecture of E.T. Chat. We reformulate timestamp prediction as an embedding matching problem. Detailed are provided in Section 4.4.1.

dataset tailored for multi-event and time-sensitive video understanding.

4.4.1 Model

Figure 4.6 presents the overall architecture of E.T. Chat. Given a video frame $\mathbf{V}_t \in \mathbb{R}^{H \times W \times 3}$ sampled at time $t \in T$, where H and W are the height and width, we first leverage a frozen visual encoder E_v to convert it into patch embeddings $\mathbf{P}_t \in \mathbb{R}^{K \times C}$, where K and C are the number of patches and feature dimension. To preserve high temporal resolution while reducing redundant compute, we adopt a frame compressor E_c to merge and project patch embeddings to a single token $\mathbf{e}_v^t \in \mathbb{R}^{1 \times D}$, where D is the embedding dimension of LLM. The compressed frame tokens $\{\mathbf{e}_v^t\}_{t=1}^T$ are then concatenated with text tokens $\{\mathbf{e}_q^n\}_{n=1}^N$ and sent into the LLM for response generation. Two projectors are also designed to map the output tokens. The detailed process will be introduced in the following sections.

Frame Compression As illustrated in Figure 4.7, the frame compressor E_c arises from [131] and consists of a Q-Former [124] E_q with M learnable queries, a context

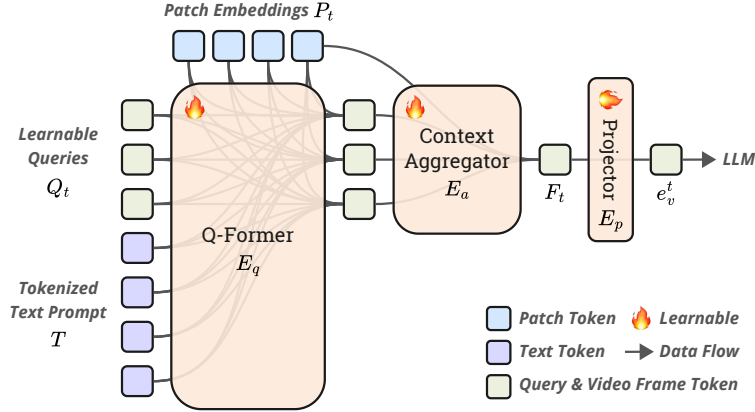


Figure 4.7: Detailed illustration of the frame compressor. It accepts video patch embeddings P_t and the text prompt T as inputs, and compress video frame features into a single token.

aggregator E_a , and a projector E_p . For each time step, E_q accepts patch embeddings P_t and the text prompt T as inputs, and re-samples them into learnable queries $Q_t \in \mathbb{R}^{M \times C}$. Then, E_a merges Q_t with P_t and compresses them into a single token. E_p finally projects it to the same embedding space as LLM. In particular, the context aggregator E_a is built upon cross attention module [248], formulated as follows:

$$\mathbf{a} = \text{Softmax}\left(\frac{(\mathbf{w}_q \times \mathbf{Q}_t)^\top \times (\mathbf{w}_k \times \mathbf{P}_t)}{\sqrt{C}}\right) \quad (4.1)$$

$$\mathbf{F}_t = \text{Mean}(\mathbf{a} \times \mathbf{P}_t + \mathbf{Q}_t) \quad (4.2)$$

Here, $\mathbf{F}_t \in \mathbb{R}^{1 \times C}$ is the compressed frame token for time t , containing text-conditioned visual information. This process can be parallelized across all frames. The projected video token sequence $\mathbf{e}_v$ is then concatenated with text tokens $\mathbf{e}_q$ to form the inputs with shape $(T + N) \times D$ for LLM.

Timestamp Prediction as Embedding Matching Our key insight focuses on the design of timestamp processing. As illustrated in Figure 4.8, existing MLLMs typically handle timestamps (for videos) or coordinates (for images) in three ways: (1) **Numerical Expressions** [196, 31, 216, 89]: Representing numbers directly in

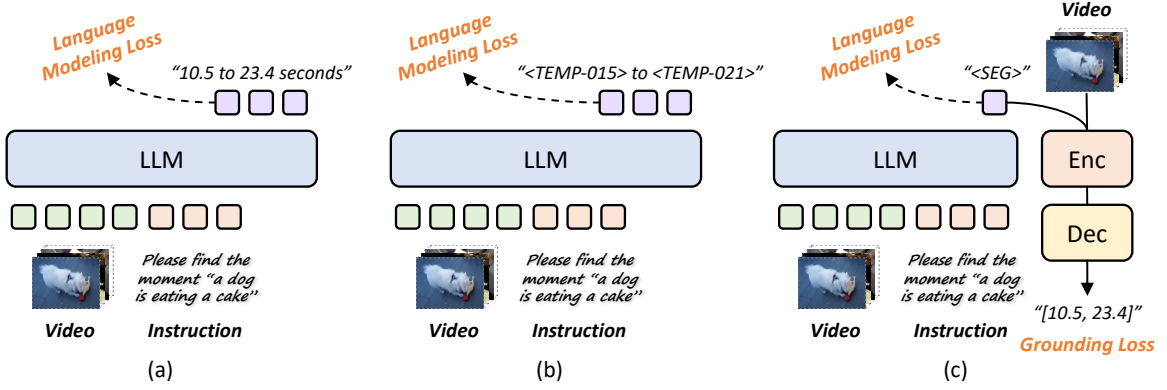


Figure 4.8: Design space for timestamp processing. Existing MLLMs handle timestamps in videos or coordinates in images via (a) numerical expressions, (b) special tokens, or (c) external modules.

the form of text. This straightforward strategy loses continuity among numbers [58, 64]. A wrong prediction could be extremely far away from the ground truth. (2) **Special Tokens** [35, 260, 90, 200]: Defining a set of special tokens to quantize time/position into a fixed number (typically 100 to 300) of bins. This solution inevitably brings severe quantization loss, and it is not flexible for videos with variable lengths. Moreover, introducing too many new tokens into the vocabulary would break the pre-trained distribution of LLMs, making them hard to optimize without post pre-training. (3) **External Modules** [116, 167, 83]: Leveraging pre-trained external models (e.g., SAM [111]) for grounding. This would introduce extra parameters and latency to LLMs. It is not directly compatible with videos as well, as existing temporal grounding models [120, 155, 178, 140, 153] are domain-specific and hard to generalize to all scenarios like SAM.

Motivated by the characteristic of Transformers that they are naturally good at selective copying rather than numerical calculations [78, 100], we propose to reformulate timestamp prediction as an *embedding matching* problem. That is, we train the model to generate/copy embeddings of video frames that it would like to refer to, and obtain timestamps by matching these embeddings back to the video.

Specifically, we define a special token $\langle \text{vid} \rangle$ used to stimulate the matching process. When $\langle \text{vid} \rangle$ is generated during inference, *e.g.*, the model outputs “*the event happens around ividi*”, this token is utilized to match a video frame token, such that the desired timestamp can be easily obtained from the matched frame index. For example, for a video sampled to 1 FPS, if $\langle \text{vid} \rangle$ is matched to the i -th frame, then $\langle \text{vid} \rangle$ means the i -th second of the video. The matching process is designed to be simple and efficient. We denote the l -th layer hidden states of $\langle \text{vid} \rangle$ token and video frame tokens as $\mathbf{h}_{vid}^l \in \mathbb{R}^{1 \times D}$ and $\mathbf{h}_{frm}^l \in \mathbb{R}^{T \times D}$, respectively. During matching, two MLPs E_{vid} and E_{frm} are first leveraged to project the hidden states to the alignment space g :

$$\mathbf{g}_{vid} = E_{vid}(\mathbf{h}_{vid}^{L-1}), \quad \mathbf{g}_{frm} = E_{frm}(\mathbf{h}_v^L) \quad (4.3)$$

Here, L refers to the total number of LLM layers. We extract $\langle \text{vid} \rangle$'s hidden states from the second-last layer to preserve a larger feature range [83]. Subsequently, we compute the cosine similarities between $\mathbf{g}_{vid}$ and all $\{\mathbf{g}_{frm}^t\}_{t=1}^T$ to obtain the matched frame index t_{match} :

$$\mathbf{s} = \frac{\mathbf{g}_{vid} \cdot \mathbf{g}_{frm}}{\|\mathbf{g}_{vid}\|_2 \cdot \|\mathbf{g}_{frm}\|_2} \in \mathbb{R}^{1 \times T}, \quad t_{match} = \text{argmax}(\mathbf{s}) \quad (4.4)$$

The frame index t_{match} is then multiplied with frame rate r to generate the real timestamp in seconds. Through this operation, the direct prediction of timestamps is replaced by embedding matching, which is easier to learn as a selective copying problem by Transformer-based models. For the case when $\langle \text{vid} \rangle$ occurs in inputs, the input features of $\langle \text{vid} \rangle$ are added with the corresponding frame features. During training, an extra binary matching loss is utilized:

$$\mathcal{L}_{matching} = -\frac{1}{T} \sum_{t=1}^T \mathbf{y}_t \cdot \log(\mathbf{s}_t) \quad (4.5)$$

Here, $\mathbf{y}_t$ denotes the binary label indicating whether t is the ground truth frame. $\mathcal{L}_{matching}$ is added with the original language modeling loss $\mathcal{L}_{language}$ to jointly optimize the model.

Numerical Continuity. The matching process above still cannot preserve numerical continuity, as the hidden states of adjacent frames may be far away from each other. We introduce two modifications to alleviate this problem. First, the causal self-attentions in LLM block out the bi-directional information flow. This is reasonable for text but limits the ability of video understanding. Thus, we allow bi-directional attentions among video tokens. Second, we introduce a smoothed label $\frac{1}{\alpha^{|t-t_{gt}|}}$ to replace the binary label y_t in Eq. 4.5, where α is a hyperparameter controlling the extent of smoothing, and t_{gt} refers to the ground truth frame index.

4.4.2 Instruction-Tuning Dataset

To fill the gap of lacking multi-event and time-sensitive training data for MLLMs, we introduce E.T. Instruct 164K, a large-scale instruction-tuning dataset with fine-grained timestamp annotations. The statistics are shown in Table 4.4.

Task Selection E.T. Instruct 164K covers 9 event-level understanding tasks, including [RVC] referred video captioning, [TVG] temporal video grounding, [TAL] temporal action localization, [EVS] extractive video summarization, [VHD] video highlight detection, [DVC] dense video captioning, [TVC] tagged video captioning, [SLC] step localization and captioning, and [GVQ] grounded video question answering. Most tasks are aligned with E.T. Bench but with different source datasets. The only exceptions are [RVC] and [TVC], where the former requires to generate captions for the the given temporal boundary, and the latter is similar to [DVC] but with start timestamps only. Note that [RAR], [ECA], [RVQ], [EPM], and [TEM] are in E.T. Bench only, serving as held-out tasks during evaluation.

Data Collection and Instruction Generation We meticulously sample videos and annotations from 14 datasets, including HowToCaption [224], DiDeMo [6], QueryD [185], TACoS [212], NaQ [206], ActivityNet [21], HACS [338], VideoXum [138],

Table 4.4: Task and sample distribution in E.T. Instruct 164K.

Task	Source	Manual Label	Avg. Duration	#Samples	Ratio
[RVC]	HowToCaption [224]	✗	176.9s	16,907	10.3%
[TVG]	DiDeMo [6]	✓	49.0s	33,000	20.1%
	QueryD [185]	✗	173.7s	4,267	2.6%
	TACoS [212]	✓	151.9s	6,693	4.1%
	NaQ [206]	✗	296.5s	10,546	6.4%
[TAL]	ActivityNet [21]	✓	118.8s	9,807	6.0%
	HACS [338]	✓	159.5s	15,218	9.3%
[EVS]	VideoXum [138]	✓	123.5s	7,989	4.9%
[VHD]	Mr. HiSum [232]	✗	196.4s	9,056	5.5%
[DVC]	ActivityNet Captions [114]	✓	118.9s	9,830	6.0%
[TVC]	ViTT [91]	✗	210.0s	2908	1.8%
[SLC]	COIN [239]	✓	138.9s	7,659	4.7%
	HowToStep [132]	✗	189.8s	20,000	12.2%
[GVQ]	EgoTimeQA [52]	✗	150.0s	10,000	6.1%
Total			146.4s	163,880	100%

Mr. HiSum [232], ActivityNet Captions [114], ViTT [91], COIN [239], HowToStep [132], and EgoTimeQA [52]. During sampling, we ensure that the videos have no overlap with E.T. Bench. Different from E.T. Bench in which all the samples are manually labeled, more than 40% of the samples in E.T. Instruct 164K are with automatically generated annotations. A similar filtering and rule-based cleaning process as E.T. Bench generation is conducted. Note that videos from EgoTimeQA are randomly cropped to 150-second long to reduce ambiguity during training.

We then convert the original annotations into instruction-following formats. Following previous works [127, 216], for each task, we write a well-designed instruction, then prompt GPT-4 [2] to extend it to multiple diverse expressions. For some tasks having overlap with previous work [216], existing instructions are also considered. We manually select and refine 6 expressions for each task to serve as the instruction templates. To obtain ground truth responses, we convert the original annotations into natural language styles using manually designed templates. The generated

Table 4.5: Instruction and response templates in E.T. Instruct 164K (Part I). Blue text means the placeholder to be filled according to each sample, and `<time>` denotes the timestamp representation.

Task	Instruction Templates	Response Format
[RVC]	- Watch the video segment in <code><time></code> - <code><time></code> and provide a concise description for it. - Watch the video portion spanning from <code><time></code> to <code><time></code> and provide a depiction of its main event. - Describe the video event happened in <code><time></code> - <code><time></code>. - Depict the event happening in the video from <code><time></code> to <code><time></code>. - Provide a description of the activity shown in the video within <code><time></code> - <code><time></code>. - Focus on the video event in <code><time></code> - <code><time></code> and describe what is happening.	<code><answer></code> .
[TVG]	- Localize the visual content described by the given textual query "<code><query></code>" in the video, and output the start and end timestamps in seconds. - Detect and report the start and end timestamps of the video segment that semantically matches the given textual query "<code><query></code>". - Give you a textual query: "<code><query></code>". When does the described content occur in the video? Please return the timestamp in seconds. - Locate the visual content mentioned in the text query "<code><query></code>" within the video using timestamps. - The given natural language query "<code><query></code>" is semantically aligned with a video moment, please give the start time and end time of the video moment. - Find the video segment that corresponds to the given textual query "<code><query></code>" and determine its start and end seconds.	The event happens in <code><time></code> - <code><time></code> .
[TAL]	- Detect and localize all the video segments containing the given action "<code><action></code>", and provide the outputs using start and end timestamps. - Find all the sections of the video where the action "<code><action></code>" occurs, and report the results with their respective start and end timestamps. - For the given action "<code><action></code>", locate all corresponding video sections and present the results with starting and ending timestamps. - Discover and determine the locations of all video portions containing the action "<code><action></code>", and return the start and end time in seconds. - Locate all instances of the action "<code><action></code>" in the video and give me the start and end times for each occurrence. - Identify and list all segments of the video where the action "<code><action></code>" takes place, providing the start and end times for each instance.	The action happens in <code><time></code> - <code><time></code> , <code><time></code> - <code><time></code> , and <code><time></code> - <code><time></code> .
[EVS]	- Summarize the video to about 15% of the original length based on the video domain "<code><domain></code>", and provide the outputs using start and end timestamps. - Condense the video to approximately 15% of its original length, focusing on the domain "<code><domain></code>", and include start and end timestamps for each summarized segment. - Reduce the video's length to roughly 15%, emphasizing the "<code><domain></code>" domain, and provide the outputs with their respective start and end timestamps. - Generate a 15% summary of the video, particularly related to the domain "<code><domain></code>", and include the start and end times for each selected portion. - Provide a summary of the video, reducing it to around 15% of its length, with a specific focus on the domain "<code><domain></code>", including start and end timestamps for the selected segments. - Summarize the video to about 15% of its total duration, specifically highlighting the domain "<code><domain></code>", and include start and end timestamps for each segment.	The summary locates in <code><time></code> - <code><time></code> , <code><time></code> - <code><time></code> , and <code><time></code> - <code><time></code> .
[VHD]	- Find a highlight moment in the video according to the given query: "<code><query></code>", and return its timestamp. - Identify a highlight in the video that matches the query "<code><query></code>", and provide the corresponding timestamp. - Detect a highlight in the video corresponding to the query "<code><query></code>", and report its timestamp. - Identify a highlight moment in the video based on "<code><query></code>", and give me the timestamp. - Localize a highlight moment that matches the given query "<code><query></code>", and return the timestamp for it. - According to the query "<code><query></code>", please find a single highlight moment in the video and provide its timestamp.	The highlight moment happens at <code><time></code> .
[DVC]	- Localize a series of activity events in the video, output the start and end timestamp for each event, and describe each event with sentences. The output format of each predicted event should be like: "start - end seconds, event description". - Determine the start and end times of various activity events in the video, accompanied by descriptions. - Capture and describe the activity events in the given video, specifying their respective time intervals, and output the time intervals in the "start - end seconds format". - Identify, timestamp, and describe various activity events occurring in the video. The timestamp should include the start time and end time in seconds. - Detect and report the start and end timestamps of activity events in the video, along with descriptions. - Pinpoint the time intervals of activity events in the video, and provide detailed descriptions for each event.	<code><time></code> - <code><time></code> , <code><event></code> . <code><time></code> - <code><time></code> , <code><event></code> . <code><time></code> - <code><time></code> , <code><event></code> .

Table 4.6: Instruction and response templates in E.T. Instruct 164K (Part II). Blue text means the placeholder to be filled according to each sample, and `<time>` denotes the timestamp representation.

Task	Instruction Templates	Response Format
[TVC]	- Densely capture all the events happened in the video, and describe them in the form of “start time, description”. - List all the events in the video in detail and format them as “start time, description”. - Chronologically describe each event in the video, noting “start time, description”. - Identify all the events in the video and provide a concise description for each of them. The response for each event should contain its start time and the description. - Pinpoint all events in the video and give a concise description with their respective start times. - Recognize every event in the video, describing each concisely and report it as “start time, description”.	<code><time></code> , <code><event></code> . <code><time></code> , <code><event></code> . <code><time></code> , <code><event></code> .
[SLC]	- Localize a series of action steps in the given video, output a start and end timestamp for each step, and briefly describe the step. - Locate and describe a series of actions or steps in the video, including their start and end timestamps. - Identify and mark the video segments corresponding to a series of actions or steps, specifying the timestamps and describing the steps. - Find, identify, and determine the temporal boundaries of a series of distinct actions or steps occurring throughout the video. For each action, output the corresponding start and end timestamps, accompanied by a concise description. - Identify and localize a series of steps or actions occurring in the video, providing start and end timestamps and related descriptions. - Locate and pinpoint a sequential series of specific actions or steps in the video, accurately specifying the start and end timestamps for each action. Additionally, provide a succinct description.	<code><time></code> - <code><time></code> , <code><step></code> . <code><time></code> - <code><time></code> , <code><step></code> . <code><time></code> - <code><time></code> , <code><step></code> .
[GVQ]	- Watch the video carefully and answer the question: “<code><question></code>”. Your response should mention the start and end timestamps as a reference. For example: “<code><answer></code>. The relevant event happens in <code><start time></code> to <code><end time></code>”. - Given the question: “<code><question></code>”, please watch the video carefully and provide both the answer and the relative moment that as a reference. - Taking the question: “<code><question></code>” into consideration, please watch the video attentively and provide your answer along with the exact timing as a reference. - Answer the following question and provide the corresponding start and end timestamps depicting the relevant moment: “<code><question></code>”. - Give a response to the question “<code><question></code>” according to the video and include the precise times marking the relevant moment. - After watching the video, provide an answer to the following question “<code><question></code>” and point out the relevant start and end times in the video.	<code><answer></code> . The relevant event happens in <code><time></code> - <code><time></code> .

instruction templates and response formats are shown in Table 4.5 & 4.6.

4.5 Experiments

4.5.1 Evaluation Settings

As different tasks in E.T. Bench are under different settings with diverse output formats. A single metric (e.g., accuracy) like existing benchmarks is not sufficient. To balance the quantity of metrics and the ease of ranking, we unify the metrics within each capability and leverage accuracy for *referring* tasks, F1 score for *grounding*

tasks, F1 score and sentence similarity for *dense captioning* tasks, and recall for *complex understanding* tasks. The evaluation metrics are as follows.

Referring All the tasks are formulated as multiple-choice question-answerings (MCQs), thus we adopt accuracy as the main metric.

Grounding We compute F1 scores averaged among IoU thresholds θ_{IoU} at four levels (0.1, 0.3, 0.5, and 0.7). For [TVG] and [EPM], only the first predicted temporal boundary is accepted. This aligns with conventional Recall@1 metric. For [TAL], all the predicted boundaries are used. In [EVS], F1 scores are computed at clip level, that is, each video is divided into 1-second long clips, and precision/recall is defined as the percentage of true positive clips with respect to all the predicted/ground truth clips. For [VHD], a prediction (single timestamp) is regarded as a true positive when it falls into any of the ground truth boundaries.

Dense Captioning Similar to *grounding*, we utilize F1 score at the same four levels of θ_{IoU} for boundary predictions in [DVC] and [SLC]. To measure the correctness of descriptions, previous works leverage traditional metrics for machine translation [193, 136, 18, 249], which cannot handle ambiguity in open-ended scenarios. Therefore, we perform evaluation at semantic level and employ sentence similarity [214] to measure the distances between predictions and ground truths. Following previous practices [52], the all-MiniLM-L6-v2 model in Sentence Transformers¹ library is used as the embedding model.

Complex Understanding We adopt Recall@1 as the metric for both [TEM] and [GVQ]. The IoU thresholds are aligned with *grounding* and *dense captioning*. For [TEM], only the first predicted temporal boundary is accepted, and it is regarded as a true positive when it has the maximum IoU among all ground truths larger than the threshold. For [GVQ], aside from boundary prediction, the MCQ answer should also be correct for a successful recall.

¹<https://github.com/UKPLab/sentence-transformers>

With the unified evaluation metrics, we are able to average them and measure a model’s general performance under each capability. To achieve this, we define 5 averaged metrics: (1) Acc_{ref} : Averaged accuracy on *referring* tasks; (2) F1_{gnd} : Averaged F1 score on *grounding* tasks; (3) F1_{cap} : Averaged F1 score on *dense captioning* tasks; (4) Sim_{cap} : Averaged sentence similarity on *dense captioning* tasks; (5) Rec_{com} : Averaged recall on *complex understanding* tasks. These metrics serve as indicators for the general performance on event-level and time-sensitive understanding.

4.5.2 Implementation Details

We adopt the pre-trained ViT-G/14 from EVA-CLIP [61] as visual encoder. The architecture of frame compressor and LLM are based on Q-Former [48, 124] and Phi-3-Mini-3.8B [1], respectively. We first pre-train the model following the stage-1 and stage-2 recipes in [131], then activate the MLP projectors E_{vid} & E_{frm} and fine-tune them on E.T. Instruct 164K. E_{vid} and E_{frm} are randomly initialized, and the embeddings for <vid> token are initialized from the averaged embeddings of all existing tokens. During fine-tuning, we freeze the visual encoder and the FFN layers in Q-Former, and introduce LoRA [88] adapters on the LLM. Therefore, only the attention layers and projectors in frame compressor E_c , LoRA adapters, and matching projectors (E_{vid} & E_{frm}) are learnable. We train the model with mixed precision (FP16) on 8 NVIDIA V100 GPUs. The training process costs around 20

Table 4.7: Hyperparameter settings.

Hyperparameter	Value
<i>Visual Encoder E_v</i>	
Frame Sampling Rate	1 FPS
Preprocessing	Center Crop
Input Resolution	224×224
Patch Size	14×14
<i>Frame Compressor E_c</i>	
Number of Learnable Queries M	32
Number of Layers	12
Hidden Size	768
<i>MLP Projector E_{vid} & E_{frm}</i>	
Number of Layers	2
Hidden Size	1536
Output Size	3072
<i>Large Language Model</i>	
LoRA r	128
LoRA α	256
LoRA Dropout Rate	0.05
LoRA Modules	QKVO Layers
<i>Model Training</i>	
Max Number of Tokens	2048
Number of Epochs	1
Batch Size	32
Learning Rate for LoRA	5e-5
Learning Rate for Other Parameters	2e-5
Weight Decay	0.0
Warmup Ratio	0.03
LR Scheduler Type	Cosine
Optimizer	AdamW [162]
AdamW β_1, β_2	(0.9, 0.999)

Table 4.8: Architectures of the models on E.T. Bench. Size means the LLM size.

Model	Size	Frame Resolution	Sampled Frames	Visual Encoder	Language Model
<i>Image-LLMs</i>					
LLaVA-1.5 [147]	7B	336 × 336	8	CLIP-ViT-L/14	Vicuna-1.5
LLaVA-InternLM2 [23]	7B	336 × 336	8	CLIP-ViT-L/14	InternLM2
mPLUG-Owl2 [308]	7B	448 × 448	8	CLIP-ViT-L/14	Llama-2
InternLM-XComposer [332]	7B	224 × 224	8	EVA-ViT-G/14	InternLM
Bunny-Llama3-V [85]	8B	384 × 384	8	SigLIP-ViT-L/14	Llama-3
MiniCPM-Llama3-V-2.5 [240]	8B	980 × 980	8	SigLIP-ViT-L/14	Llama-3
Qwen-VL-Chat [14]	7B	448 × 448	8	CLIP-ViT-bigG/14	Qwen
<i>Video-LLMs</i>					
Video-ChatGPT [171]	7B	224 × 224	100	CLIP-ViT-L/14	LLaMA
Video-LLaVA [135]	7B	224 × 224	8	LanguageBind-ViT-L/14	Vicuna-1.5
LLaMA-VID [131]	7B	224 × 224	1 FPS	EVA-ViT-G/14	Vicuna-1.5
Video-LLaMA-2 [328]	7B	224 × 224	8	EVA-ViT-G/14	Llama-2-Chat
PLLaVA [291]	7B	672 × 672	16	CLIP-ViT-L/14	Vicuna-1.5
VTimeLLM [89]	7B	224 × 224	100	CLIP-ViT-L/14	Vicuna-1.5
VTG-LLM [80]	7B	224 × 224	96	EVA-ViT-G/14	Llama-2
TimeChat [216]	7B	224 × 224	96	EVA-ViT-G/14	Llama-2
LITA [90]	13B	224 × 224	100	CLIP-ViT-L/14	Vicuna-1.3
E.T. Chat (Ours)	3.8B	224 × 224	1 FPS	EVA-ViT-G/14	Phi-3-Mini

hours. More detailed hyperparameters are listed in Table 4.7.

4.5.3 Baselines

We extensively evaluate 20 MLLMs on E.T. Bench, including 7 open-source Image-LLMs (LLaVA-1.5 [147], LLaVA-InternLM2 [23], mPLUG-Owl2 [308], InternLM-XComposer [332], Bunny-Llama3-V [85], MiniCPM-Llama3-V-2.5 [240], and Qwen-VL-Chat [14]), 9 open-source Video-LLMs (Video-ChatGPT [171], Video-LLaVA [135], LLaMA-VID [131], Video-LLaMA-2 [328], PLLaVA [291], VTimeLLM [89], VTG-LLM [80], TimeChat [216], and LITA [90]), and 4 proprietary MLLMs (GPT-4V [187], GPT-4o [188], Gemini-1.5-Flash [213], and Gemini-1.5-Pro [213]).

We compare the architectures of open-source MLLMs in Table 4.8. Optional visual encoders for these models are CLIP [205], EVA-CLIP [61], SigLIP [322], OpenCLIP [94], and LanguageBind [348], while the LLM backbones include LLaMA [245],

Llama-2 [246], Llama-3 [4], Vicuna [43], InternLM [332], InternLM2 [23], Qwen [13], and Phi-3-Mini [1].

4.5.4 Main Results

We extensively evaluate 7 open-source Image-LLMs, 9 open-source Video-LLMs, and 4 proprietary MLLMs on E.T. Bench. For Image-LLMs, we uniformly sample 8 frames and add an extra prompt indicating the video duration as a hint for timestamps. For Video-LLMs, we use their default number of frames as inputs. Proprietary MLLMs are evaluated by calling APIs on a subset with 470 samples. The inputs for GPT-4V and GPT-4o are aligned with Image-LLMs. For Gemini-1.5 models, raw videos are directly uploaded.

The evaluation results are presented in Table 4.9. We report the metrics averaged among sub-tasks due to space limit. The *Random* in the first row refers to random guessing. We also provide comparisons and ranking in Figure 4.9. Below we summarize our key findings from the results.

Performance gap between Image- and Video-LLMs. We observe that on *referring* tasks, most Image- and Video-LLMs perform at the same level. Some Image-LLMs such as XComposer and Qwen-VL-Chat can even beat most Video-LLMs. This is because the videos for *referring* are generally short, as compared in Figure 4.3 (right), such that the sampled 8 frames cover most information. The gap becomes larger on tasks with longer videos such as [TVG] and [TAL], demonstrating the importance of temporal modeling on E.T. Bench compared with other benchmarks.

Strong Video-LLMs on existing benchmarks struggle on E.T. Bench. Some state-of-the-art Video-LLMs on existing benchmarks, *e.g.*, Video-LLaMA-2 and PLLaVA, are less effective on our E.T. Bench, especially on *grounding* and *dense captioning* tasks. We attribute this to the single-frame bias caused by both model architecture and training data. It also motivate us to consider the balance between spatial and

Table 4.9: Performance of representative MLLMs on E.T. Bench. We evaluate both open-source and proprietary models. The best and second-best results are marked with **bold** and underline, respectively.

Method	Referring			Grounding					Dense Captioning				Complex	
	RAR _{Acc}	EVC _{Acc}	RVQ _{Acc}	TVG _{F1}	EPM _{F1}	TAL _{F1}	EVS _{F1}	VHD _{F1}	DVC _{F1}	DVC _{Sim}	SLC _{F1}	SLC _{Sim}	TEM _{Rec}	GVQ _{Rec}
<i>Random</i>	25.0	25.0	20.0	–	–	–	–	–	–	–	–	–	–	–
<i>Open-source Image-LLMs: All models use 8 uniformly sampled frames as inputs. Prompts have been added with hints about timestamps.</i>														
LLaVA-1.5 [147]	34.2	27.4	26.2	6.1	1.9	7.8	2.4	30.9	14.5	11.5	0.9	9.5	7.7	0.0
LLaVA-InternLM2 [23]	34.0	34.8	37.0	2.7	0.1	0.3	0.2	32.3	16.9	8.5	0.1	4.7	7.2	1.5
mPLUG-Owl2 [308]	<u>37.8</u>	26.4	34.6	1.1	0.2	3.0	4.1	36.8	0.1	8.1	0.1	7.7	6.2	0.0
XComposer [332]	33.0	19.6	40.2	4.9	1.5	9.9	2.8	28.9	5.4	5.9	2.7	9.0	10.5	0.0
Bunny-Llama3-V [85]	33.2	27.4	26.6	7.0	0.1	5.1	0.4	30.6	13.5	8.8	0.1	7.6	7.2	0.0
MiniCPM-V-2.5 [240]	37.6	28.0	<u>37.6</u>	2.0	0.1	4.4	13.4	18.7	6.2	11.8	1.4	9.7	0.7	0.0
Qwen-VL-Chat [14]	33.4	32.2	33.6	16.2	4.0	10.7	16.3	34.4	17.4	13.8	6.2	13.1	3.2	1.5
<i>Open-source Video-LLMs: All models use their default numbers of frames as inputs.</i>														
Video-ChatGPT [171]	22.6	24.2	23.0	7.0	1.3	15.1	8.4	28.8	8.8	11.3	5.7	10.2	15.9	0.0
Video-LLaVA [135]	33.6	33.0	22.6	7.0	1.9	15.0	0.3	28.9	28.0	15.0	0.9	8.3	7.5	0.1
LLaMA-VID [131]	30.4	38.4	28.8	5.5	1.2	8.0	1.4	30.0	27.1	12.6	5.2	11.1	7.0	0.9
Video-LLaMA-2 [328]	28.8	27.4	28.0	0.1	0.0	0.0	0.0	1.5	0.6	14.5	0.0	15.2	0.0	0.1
PLLaVA [291]	33.8	22.6	31.8	6.9	1.1	5.7	0.3	28.9	13.3	10.6	9.7	11.8	4.1	1.2
VTimeLLM [89]	28.4	31.0	29.2	7.6	1.9	<u>18.2</u>	15.9	28.9	12.4	13.1	8.7	6.4	6.8	1.9
VTG-LLM [80]	6.6	12.0	7.8	15.9	3.7	14.4	26.8	<u>48.2</u>	40.2	<u>18.6</u>	20.8	14.4	8.9	1.4
TimeChat [216]	30.8	27.6	24.6	<u>26.2</u>	3.9	10.1	<u>29.1</u>	40.5	16.6	12.5	5.6	9.2	18.0	1.5
LITA [90]	33.0	40.8	27.2	22.2	<u>4.6</u>	18.0	29.7	23.9	<u>39.7</u>	17.2	<u>21.0</u>	12.2	16.0	<u>2.2</u>
E.T. Chat (Ours)	44.6	37.0	33.6	38.6	10.2	30.8	25.4	62.5	38.4	19.7	24.4	<u>14.6</u>	<u>16.5</u>	3.7
<i>Proprietary MLLMs: Evaluated on a subset with 470 samples.</i>														
GPT-4V [187]	33.3	<u>40.9</u>	46.2	27.0	1.8	18.0	28.6	55.1	16.1	19.4	21.9	13.5	<u>23.9</u>	0.0
GPT-4o [188]	27.8	27.3	<u>57.7</u>	40.4	4.5	20.0	<u>17.6</u>	56.9	46.9	22.3	<u>23.1</u>	14.9	13.6	0.0
Gemini-1.5-Flash [213]	<u>38.9</u>	50.0	61.5	43.9	5.4	27.0	5.4	<u>60.8</u>	31.6	14.9	16.5	13.3	20.8	<u>1.0</u>
Gemini-1.5-Pro [213]	61.1	27.3	<u>57.7</u>	<u>43.1</u>	<u>6.2</u>	33.8	7.9	47.0	24.0	17.5	5.8	9.8	32.1	<u>1.0</u>
E.T. Chat[†] (Ours)	33.3	31.8	30.8	32.9	8.9	<u>28.1</u>	15.3	60.9	<u>39.8</u>	<u>19.5</u>	23.7	<u>14.8</u>	14.8	1.3

[†] Evaluated on the same subset as proprietary MLLMs

temporal modeling in Video-LLMs.

Some Video-LLMs fail to follow instructions. During evaluation, we also notice that some models, *e.g.*, Video-LLaVA and Video-LLaMA-2, fail to generate outputs in desired formats for some tasks even with carefully designed instructions and examples. For instance, Video-LLaVA can only generate repeated text outputs without any timestamps for [EVS], while Video-LLaMA-2 faces the similar problem on almost all *grounding* tasks. we claim that this is due to the severe overfitting on their

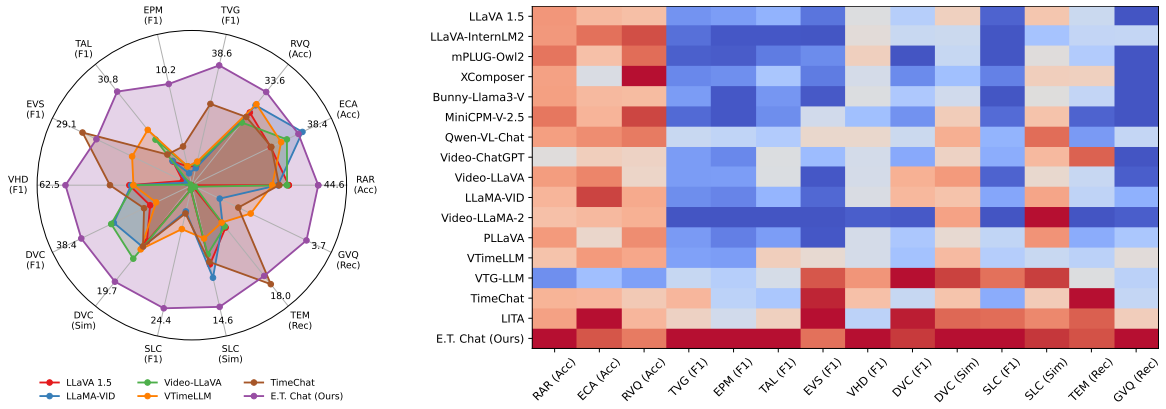


Figure 4.9: **Left:** Performance comparison between E.T. Chat and representative models. **Right:** Ranking of MLLMs on E.T. Bench, where red means higher ranks and blue represents lower ranks.

instruction-tuning data, which do not contain any timestamps outputs.

Some Image-LLMs performs exceptionally well. Qwen-VL-Chat and Bunny-Llama-3-8B-V are two models producing relatively good results compared with other Image-LLMs and even some Video-LLMs. On [TVG] and [DVC], Qwen-VL-Chat achieves 15.6 and 21.3 F1 scores, respectively, surpassing a number of Video-LLMs. But there is still a significant gap when compared with best-performing Video-LLMs such as TimeChat and LITA.

Time-sensitive Video-LLMs are the first-class models. VTimeLLM, TimeChat, and LITA are three Video-LLMs with explicit optimizations for timestamps modeling, such that they can persistently follow instructions and generate considerable responses when facing time-sensitive scenarios.

Proprietary MLLMs are still competitive. Even with 8-frame inputs, GPT-4V and GPT-4o show their significance compared with open-source models on some tasks such as [TVG], [TAL], and [DVC]. By supporting direct video inputs, Gemini-1.5 series achieve the strongest performance on a number of tasks in E.T. Bench, including [RAR], [EVC], [RVQ], [TVG], [TAL], [VHD], and [TEM].

Table 4.10: Comparison on architectural designs.

<vid> Token	Bi-directional	Smoothing	Acc _{ref}	F1 _{gnd}	F1 _{cap}	Sim _{cap}	Rec _{com}
			25.0	17.5	21.2	12.3	8.6
✓			34.0	25.2	26.4	15.4	9.2
✓	✓		33.7	<u>30.5</u>	<u>27.5</u>	<u>15.8</u>	<u>9.8</u>
✓		✓	<u>34.5</u>	25.8	26.4	13.6	9.5
✓	✓	✓	38.4	33.5	31.4	17.1	10.1

Table 4.11: Choices of frame compressor.

Method	Acc _{ref}	F1 _{gnd}	F1 _{cap}	Sim _{cap}	Rec _{com}
Pooling	29.6	25.5	21.1	11.3	9.1
Q-Former	38.4	33.5	31.4	17.1	10.1

E.T. Chat fills the gap between open-source and proprietary MLLMs. Benefit from the novel timestamps processing design and the multi-event instruction-tuning data, E.T. Chat achieves state-of-the-art performance among open-source MLLMs on most tasks, and obtain comparable results as proprietary MLLMs. Notably, significant improvements can be viewed on [EPM], [VHD], [TEM], and [GVQ].

4.5.5 Ablation Studies

Effect of Architectural Designs We verify the effectiveness of <vid> token, bi-directional attention across video tokens, and label smoothing during training. The results are compared in Table 4.10. Without introducing the <vid> token (first row), our model falls back to the *numerical expression* variant discussed in Section 4.4.1, which struggles in timestamp prediction even with instruction-tuning on E.T. Instruct 164K. By reformulating timestamp prediction as embedding matching (second row), our method significantly works better on all tasks on E.T. Bench. Extra modifications, *i.e.*, bi-directional attention (third row) and label smoothing (fourth row), further enhance the model to achieve better localization abilities,

Table 4.12: Choices of layers for matching. Layer_{vid} and Layer_{frm} denote the index of LLM layer utilized for matching. $\underline{-1}$ means using the final-layer hidden states.

Layer_{vid}	Layer_{frm}	Acc_{ref}	F1_{gnd}	F1_{cap}	Sim_{cap}	Rec_{com}
-1	-1	<u>38.2</u>	32.2	31.0	17.2	9.6
-1	-2	37.8	32.5	30.8	16.5	<u>9.9</u>
-2	-1	38.4	<u>33.5</u>	31.4	<u>17.1</u>	10.1
-2	-2	37.6	33.8	<u>31.2</u>	16.7	9.7

Table 4.13: Comparison on learnable modules. ATTN and FFN represent the attention and feed-forward layers in Q-Former, respectively.

Q-Former E_q		Aggregator	Projector	LLM	Acc_{ref}	F1_{gnd}	F1_{cap}	Sim_{cap}	Rec_{com}
ATTN	FFN	E_a	E_p	(LoRA)					
					36.1	29.2	27.1	14.1	8.7
					37.3	30.5	28.3	15.0	<u>9.6</u>
					<u>37.9</u>	<u>31.8</u>	29.6	15.7	9.5
					38.4	33.5	<u>31.4</u>	<u>17.1</u>	10.1
					37.5	29.1	32.2	17.5	9.3

demonstrated by the substantial increase in F1_{gnd} .

Choice of Frame Compressor Table 4.11 compares the performance of two design choices for frame compressor E_c , i.e., a naive spatial pooling among frame patches $\mathbf{P}_t$ and a query-guided compression based on Q-Former [124]. The results confirm that employing Q-Former for frame compression is a superior alternative, as it can adaptively determine the aggregation weight according to queries.

Choice of LLM Layer for Matching As articulated above, during matching, we utilize the second-last layer’s hidden states for the $\langle \text{vid} \rangle$ token, while the final-layer’s hidden states for frame tokens. This strategy take into consideration the small feature range of final-layer hidden states for the $\langle \text{vid} \rangle$ token [83]. We further verify its effectiveness in Table 4.12. While the results are essentially similar, current setting exhibits a marginally superior overall performance.

Table 4.14: Effect of α for label smoothing.

α	Acc_{ref}	F1_{gnd}	F1_{cap}	Sim_{cap}	Rec_{com}
1.0	37.2	30.8	26.3	14.8	9.6
1.5	37.9	31.6	28.6	15.4	10.3
2.0	38.4	<u>33.5</u>	31.4	17.1	<u>10.1</u>
2.5	<u>38.1</u>	33.0	<u>29.8</u>	<u>16.8</u>	9.2
3.0	37.5	33.7	28.4	16.0	9.8

Learnable Modules In Table 4.13, we justify the training strategy for instruction-tuning. Updating only the context aggregator E_a and projector E_p (first row) makes the training hard to converge with new tokens, and fine-tuning the LLM with LoRA (second row) brings better performance. We contend that the pre-trained Q-Former is suitable only for short & single event videos due to the constraints of pre-training data. Serving as the frame compressor, such limitation would hinder the model’s performance. Line 3 ~ 5 corroborate our hypothesis, as updating Q-Former on E.T. Instruct 164K brings notable performance improvements. Furthermore, we observe that fine-tuning the whole Q-Former makes the model slightly overfit to dense captioning tasks, and freezing its FFN layers could strike the balance between adapting to new data and retaining pre-trained capabilities.

Effect of α for label smoothing. We ablate the effect of different α values for label smoothing in Table 4.14. Smaller α values make the optimization goal of matching scores smoother. Generally, setting α to around 2.0 brings considerable results.

Joint effect of model and instruction-tuning dataset. We compare in Table 4.15 the joint effect of model design and instruction-tuning dataset collection. We choose two representative models (LLaMA-VID [131] and TimeChat [216]) as baselines and train them on E.T. Instruct 164K. Our E.T. Bench is also trained on TimeIT dataset [216] for in-depth comparison. The comparison results between line 1 & 2, 3 & 4, and 5 & 6 demonstrate the effectiveness of E.T. Instruct 164K. Results in line 2, 4, and 6 verify the significance of our model design.

Table 4.15: Joint effect of model and instruction-tuning (IT) dataset.

Model	IT Dataset	Acc _{ref}	F1 _{gnd}	F1 _{cap}	Sim _{cap}	Rec _{com}
LLaMA-VID [131]	723K Corpus [131]	32.5	9.2	16.2	11.9	4.0
LLaMA-VID [131]	E.T. Instruct 164K (Ours)	31.3	16.0	19.8	<u>14.9</u>	7.8
TimeChat [216]	TimeIT [216]	27.7	21.9	11.1	10.8	9.7
TimeChat [216]	E.T. Instruct 164K (Ours)	29.5	<u>24.3</u>	<u>21.5</u>	11.5	11.4
E.T. Chat (Ours)	TimeIT [216]	<u>34.9</u>	22.1	20.1	13.4	6.9
E.T. Chat (Ours)	E.T. Instruct 164K (Ours)	38.4	33.5	31.4	17.1	<u>10.1</u>

Table 4.16: Ablation study on instruction-tuning tasks.

RVC	TVG	TAL	EVS	VHD	DVC	SLC	GVQ	Acc _{ref}	F1 _{gnd}	F1 _{cap}	Sim _{cap}	Rec _{com}
✓								36.5	9.8	0.4	10.3	0.5
✓	✓							36.0	12.8	3.7	12.5	5.1
✓	✓	✓						35.4	31.9	10.3	11.5	<u>9.9</u>
✓	✓	✓	✓					35.6	32.5	9.5	11.8	9.5
✓	✓	✓	✓	✓				35.9	33.6	15.1	10.8	9.7
✓	✓	✓	✓	✓	✓			<u>37.4</u>	34.8	18.2	12.5	9.4
✓	✓	✓	✓	✓	✓	✓		34.2	33.7	<u>28.5</u>	<u>14.3</u>	9.2
✓	✓	✓	✓	✓	✓	✓	✓	38.4	<u>33.5</u>	31.4	17.1	10.1

Effect of instruction-tuning tasks. To study the effect of each task during instruction tuning, we provide detailed comparisons in Table 4.16. We observe that adding more tasks for instruction-tuning might slightly affect the performance on original tasks. This can be alleviated by carefully balancing the number of samples per task.

4.5.6 Visualizations

Figure 4.10~4.15 present task-specific qualitative comparisons among 5 representative open-source MLLMs, *i.e.*, LLaVA-1.5 [147], Video-ChatGPT [171], LLaMA-VID [131], TimeChat [216], and E.T. Chat. The correct model responses are marked green. We observe that the unsatisfactory performance of existing methods comes from (1) weak instruction-following abilities, (2) low temporal resolution, (3) lack of time-sensitive designs, and (4) lack of multi-event instruction-tuning data.

4.6 Summary

In this chapter, we systematically investigate the solution for integrating video temporal grounding with multi-modal large language models for time-sensitive video reasoning. A complete solution, *i.e.*, a benchmark, a model, and an instruction-tuning dataset is proposed to study this problem. To summarize, our contribution in this chapter is three-fold:

1. A comprehensive benchmark (E.T. Bench) focusing on event-level and time-sensitive video-language understanding for multi-modal large language models (MLLMs) evaluation. This benchmark encompasses 7.3K samples under 12 tasks with 7K videos (251.4h total length) under 8 domains. Currently, 8 Image-LLMs and 12 Video-LLMs have been evaluated on it.
2. A new Video-LLM (E.T. Chat) designed for event-level and time-sensitive video understanding. The proposed model can comprehend user referred moments and generate accurate timestamps in its responses. Strong fine-grained video understanding capabilities has achieved by this model.
3. A large-scale instruction-tuning dataset (E.T. Instruct 164K) to support the training of Video-LLMs. The dataset consists of high-quality samples curated for carefully designed temporal understanding tasks.

We also list our novelties in this chapter:

1. We identify the fundamental challenge of timestamp understanding and prediction in MLLM-based video temporal understanding. A complete solution including benchmark, model, and dataset is therefore proposed.
2. We present a novel reformulation of timestamp prediction in MLLMs as an embedding matching problem. This new formulation enables a unified framework that effectively integrates discrete next-token prediction with continuous timestamp regression.

3. We also introduce a new smoothed embedding matching loss to enhance the model’s temporal awareness. Comprehensive ablation studies further validate the effectiveness of the proposed scheme.
4. For the benchmark and instruction-tuning dataset, we design two controllable and scalable data engines to generate high-quality evaluation and training samples, respectively. These pipelines are easily extensible to more tasks.

4.7 Limitations & Future Work

Currently, the proposed E.T. Bench is based on `val` or `test` split of existing datasets, whose training split might be included for MLLM training. This could potentially result in data leakage, thereby compromising the integrity of the zero-shot evaluation framework and leading to unfair comparisons. Therefore, an important future work to overcome this limitation would be self-collecting new videos and provide manual annotations under each carefully designed task. More flexible input-output formats shall also be considered.

For E.T. Chat, even with advanced frame compression strategies, the low spatial resolution limits the model’s ability to perceive spatial details. Modern Image-LLMs can support extra-high-resolution image inputs, but this is not directly compatible to videos due to the large computational overhead. Our future work will focus on the balance between spatial and temporal resolution for Video-LLMs.

4.8 Licenses

The annotations of E.T. Bench are provided to the public under CC BY-NC-SA 4.0 license. A copy can be obtained at <https://creativecommons.org/licenses/by-nc-sa/4.0/>. By downloading our dataset from our website or other sources,

the user agree to adhere to the terms of CC BY-NC-SA 4.0 and licenses of the source datasets. Licenses of the source datasets are listed in Table 4.17.

Table 4.17: Licenses of source datasets in E.T. Bench.

Dataset	License	Link
Perception Test [194]	CC BY 4.0	https://creativecommons.org/licenses/by/4.0/
Charades-STA [70, 225]	Non-Commercial	https://prior.allenai.org/projects/data/charades/license.txt
STAR [273]	Apache 2.0	https://github.com/csbobby/STAR/blob/main/LICENSE
QVHighlights [120]	CC BY-NC-SA 4.0	https://creativecommons.org/licenses/by-nc-sa/4.0/
Ego4D-NLQ [76]	Custom	https://ego4d-data.org/pdfs/Ego4D-Licenses-Draft.pdf
THUMOS'14 [104]	Research Only	https://www.crcv.ucf.edu/THUMOS14
THUMOS'15 [75]	Research Only	http://www.thumos.info/
TVSum [230]	CC BY 4.0	https://creativecommons.org/licenses/by/4.0/
SumMe [81]	N/A	https://doi.org/10.1007/978-3-319-10584-0_33
YouTube Highlights [234]	N/A	https://doi.org/10.1007/978-3-319-10590-1_51
HiREST [321]	MIT License	https://opensource.org/license/mit
YouCook2 [346]	MIT License	https://opensource.org/license/mit
CrossTask [351]	BSD 3-Clause	https://opensource.org/license/bsd-3-clause
HT-Step [3]	CC BY-NC 4.0	https://creativecommons.org/licenses/by-nc/4.0/
QAEgo4D [20]	N/A	https://doi.org/10.1109/CVPRW56347.2022.00162

Chapter 4. Event-Level & Time-Sensitive Video-Language Understanding

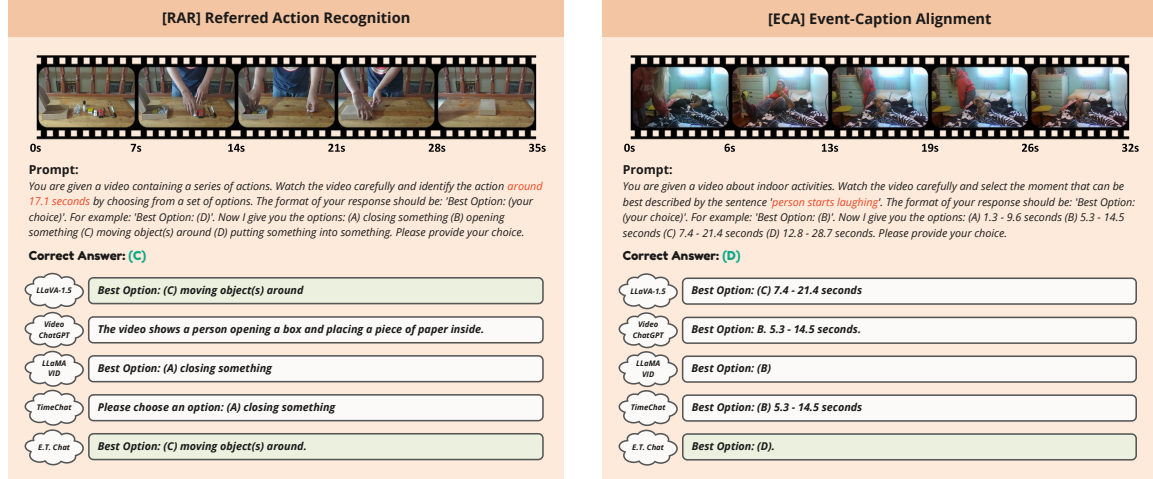


Figure 4.10: Qualitative comparison on [RAR] (left) and [ECA] (right).

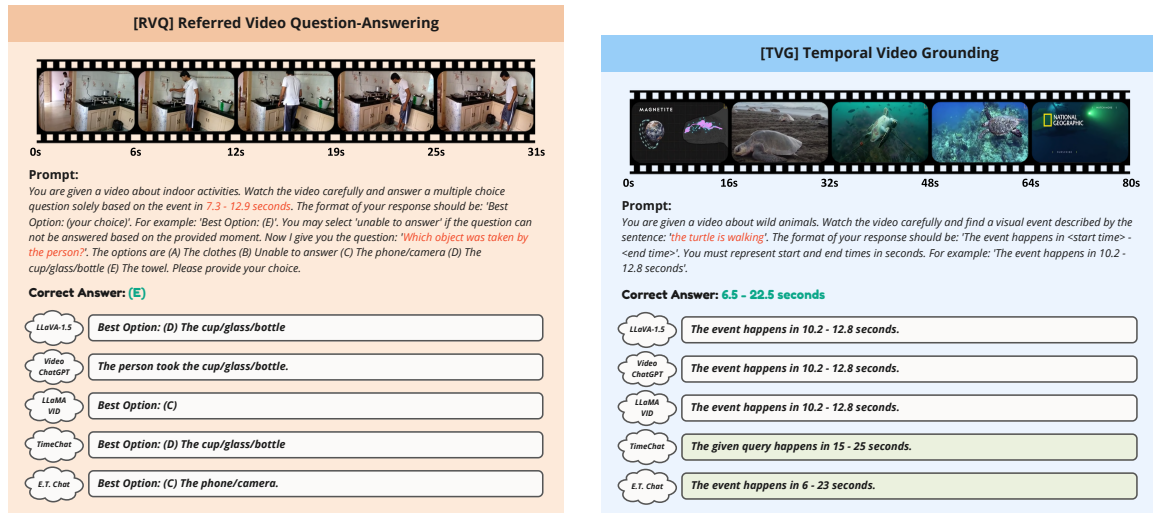


Figure 4.11: Qualitative comparison on [RVQ] (left) and [TVG] (right).

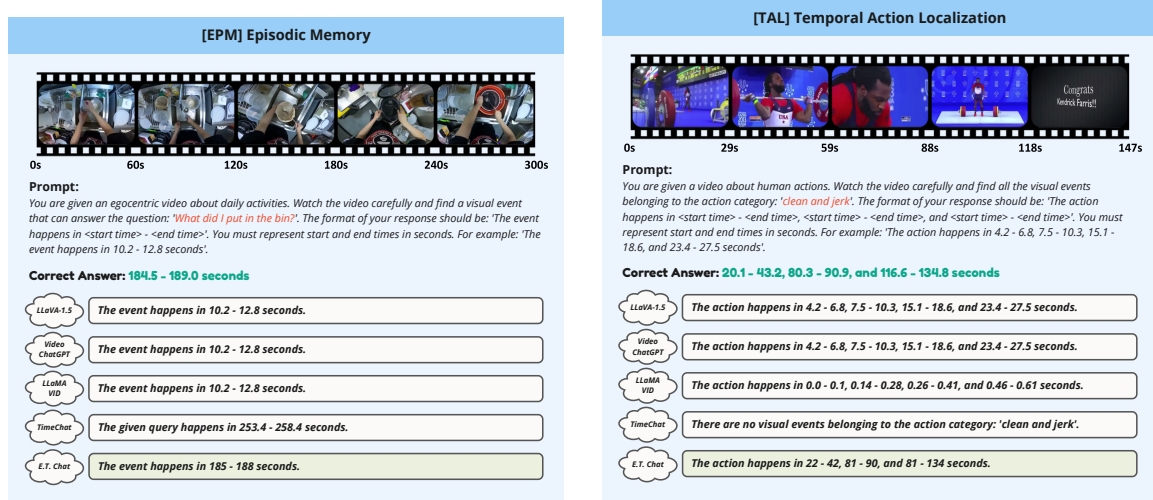


Figure 4.12: Qualitative comparison on [EPM] (left) and [TAL] (right).

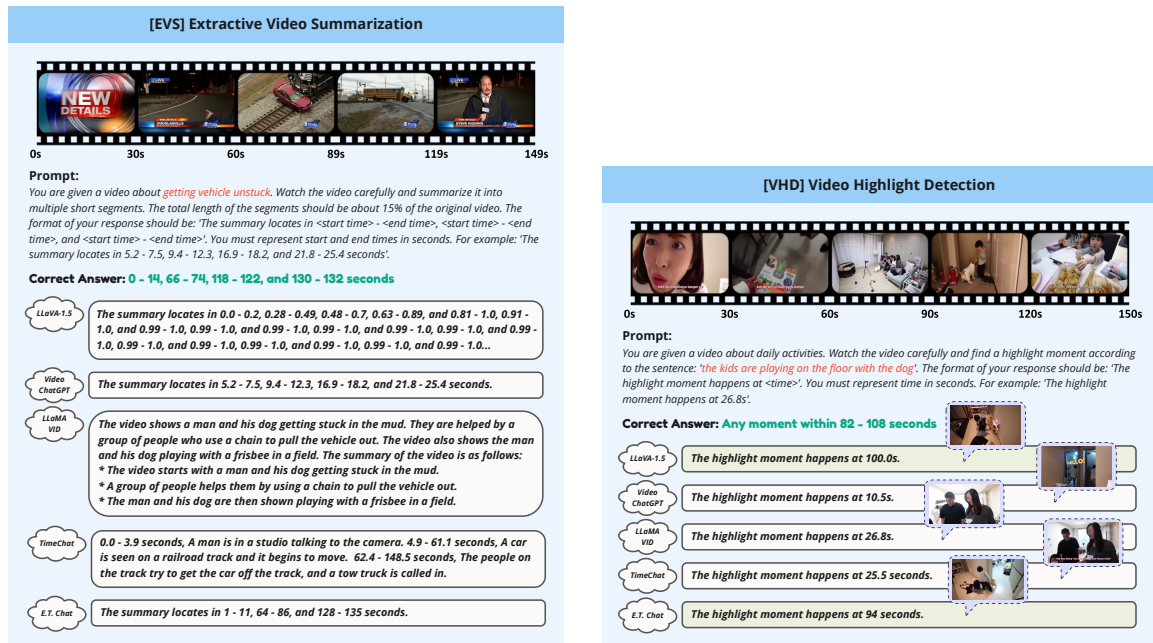


Figure 4.13: Qualitative comparison on [EVS] (left) and [VHD] (right).

Chapter 4. Event-Level & Time-Sensitive Video-Language Understanding

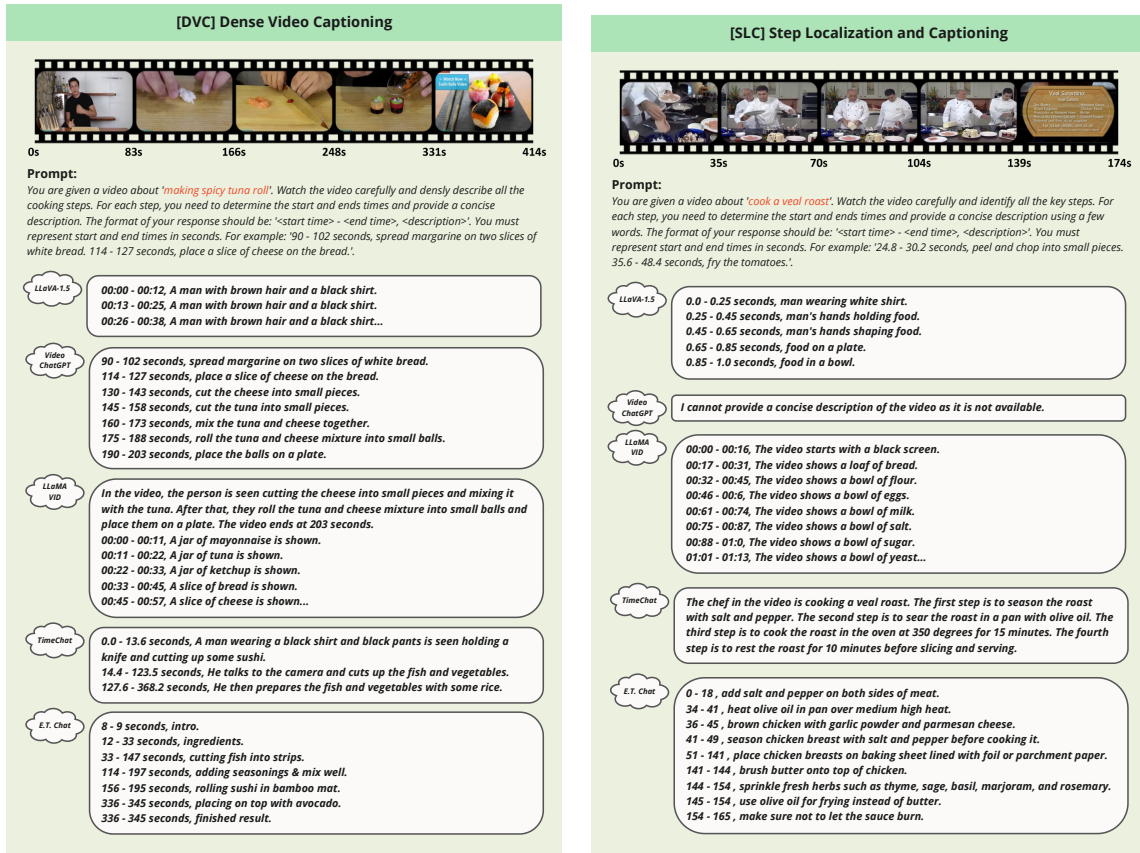


Figure 4.14: Qualitative comparison on [DVC] (left) and [SLC] (right).

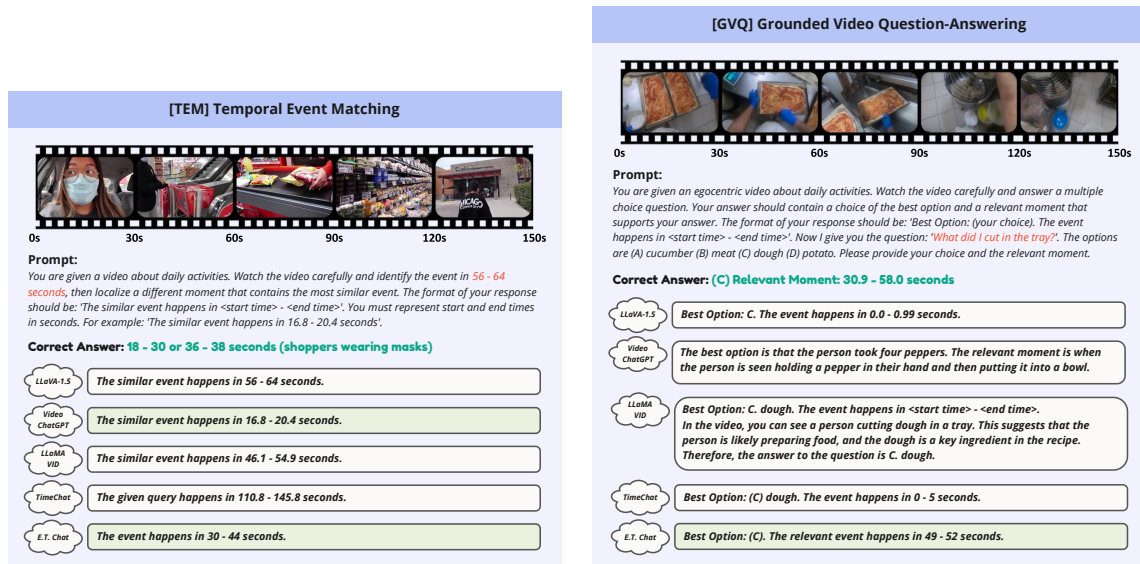


Figure 4.15: Qualitative comparison on [TEM] (left) and [GVQ] (right).

Chapter 5

Unified Referring and Segmentation for Pixel-Level Visual Reasoning

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 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.

5.1 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 [188, 50, 8, 148, 156], autonomous driving (AD) [47, 257, 330, 38], and Embodied AI [339, 57, 109, 312].

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 [144, 21, 346] and general question answering [126, 66, 173, 158]. 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 [39, 318, 319], referring expression segmentation [108, 174, 146, 54,



Figure 5.1: UniPixel flexibly supports a variety of fine-grained image and video understanding tasks, including referring/reasoning/interactive segmentation and referred video description & question answering. It can also handle a novel PixelQA task that jointly requires object-centric referring, segmentation, and QA in videos.

[221, 199], and reasoning segmentation [116, 97, 297, 16, 341]. 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 [116]), 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 [284, 283, 282, 286, 285] multi-modal reasoning abilities, resulting in degraded performance on general visual understanding benchmarks [314, 288, 99].

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 objects. 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 Figure 5.1, given a video¹, 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 information. All these operations are seamlessly conducted *within a single model*, eliminating the need for external frame samplers [297], mask generators [318, 319], or object trackers [16].

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 8 image/video referring/segmentation tasks, including ReVOS [297] for reasoning video object segmentation, MeViS [54], Ref-YouTube-VOS [221], Ref-DAVIS17 [199] for referring video object segmentation (RVOS), GroundMoRe [51] for motion-grounded video reasoning, RefCOCO+/g [108, 174] for referring expression segmentation (RES) and comprehension (REC), ReasonSeg [116] for reasoning segmentation, VideoRefer-Bench^D [319] for referred video description,

¹Images are treated as single-frame videos, thus we do not explicitly differentiate them in this work.

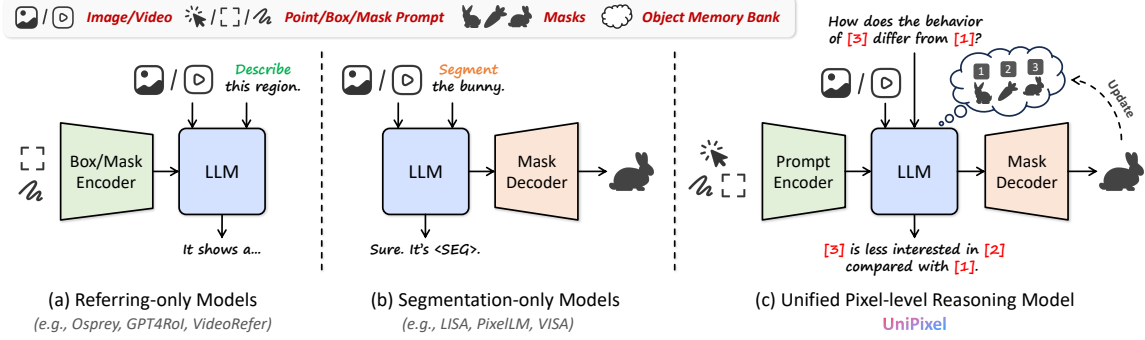


Figure 5.2: Schematic comparison between UniPixel and its counterparts. To the best of our knowledge, UniPixel is the first unified method supporting simultaneous object referring and segmentation.

and VideoRefer-Bench^Q [319] for referred video question answering. Our method achieves state-of-the-art performance in diverse scenarios. Notably, on video reasoning segmentation and referred video QA tasks, our 3B model obtains **62.1** $\mathcal{J}\&\mathcal{F}$ on ReVOS [297] and **72.8%** Acc on VideoRefer-Bench^Q [319], surpassing strong counterparts with 7B $\sim$ 13B parameters. Further ablation studies demonstrate the mutual reinforcement effect across tasks. 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.

5.2 Related Work

5.2.1 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 [148, 147, 48, 349] involve an MLP projector or Q-Former [124] 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 [14, 255, 15] and InternVL [41, 190, 40] series, have narrowed the gap with advanced proprietary models like the GPT [187, 188] and Gemini families [213, 49]. Recent studies [189, 79, 168, 133, 156] 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.

5.2.2 Visual Referring and Segmentation

To meet the growing demand for fine-grained visual understanding [159, 154, 155, 153, 276], recent efforts have focused on enhancing LMMs with object referring and segmentation capabilities, as compared in Figure 5.2. LISA [116] is a representative model that enables LMM-based segmentation by integrating SAM [111] 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 [323, 217, 336, 209, 97] have explored advanced mask decoders, more flexible tasks, and larger-scale datasets. Recent studies have also extended these capabilities to videos [16, 297, 315]. Additionally, some research has examined regional

understanding through boxes [39] and masks [318, 319]. While recent approaches attempt to unify these two capabilities, they either support only images [209] or rely on sub-optimal, tool-based pipelines [62]. To the best of knowledge, UniPixel is the first end-to-end method unifying object referring and mask prediction.

5.3 Method

5.3.1 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.

5.3.2 Overview

Figure 5.3 presents an overview of UniPixel. It is built upon the Qwen2.5-VL [15] 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* supporting three types of visual prompts, (2) an *object memory bank* for storing object information and injecting it into the response generation process, and (3) a *mask decoder* for generating spatial-temporal masks. We also extend the LLM’s vocabulary by adding <REF>, <MEM>, and

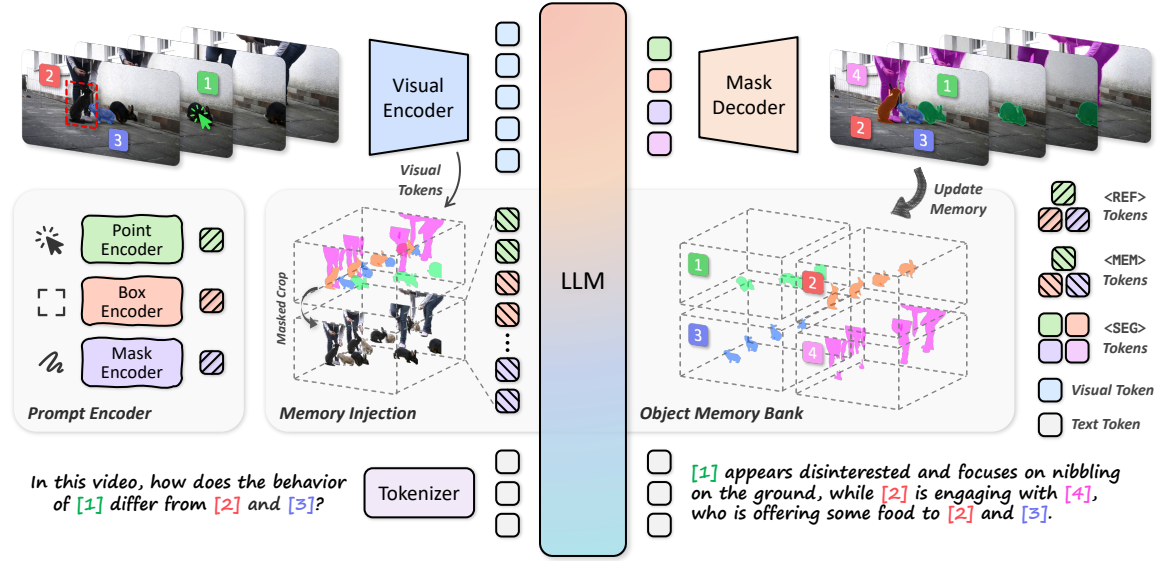


Figure 5.3: The overall architecture of UniPixel. Given a video, a question, and visual prompts, the model encodes them into tokens via the visual encoder, prompt encoder, and tokenizer, respectively, then predicts a spatial-temporal mask for each visual prompt via the mask decoder. The masks are updated into the object memory bank, and subsequently injected into the prompt for pixel-level reasoning.

<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.

5.3.3 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.

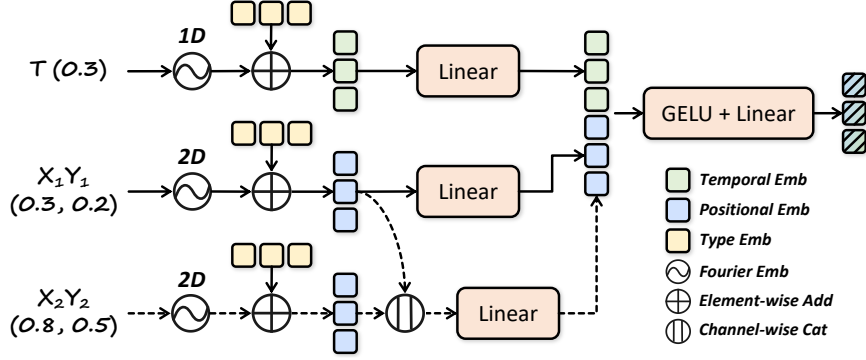


Figure 5.4: Illustration of joint positional & temporal encoding. The joint encoding approach supports both point (X_1Y_1T) and box ($X_1Y_1X_2Y_2T$) prompts.

For sparse prompts (points and boxes), as shown in Figure 5.4, we encode each position (x_i, y_i) as the sum of a 2D Fourier embedding [237] and a learnable type embedding. 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 $\rightarrow$ Linear block, such that the sparse coordinates in a point/box are encoded into a compact high-dimensional token. This design is inspired by [111, 210] 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 $M \rightarrow L$ projector (Linear $\rightarrow$ GELU $\rightarrow$ Linear) is leveraged to project the pooled visual features to the LLM’s embedding space.

5.3.4 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 [39, 318, 319] 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*. An example is shown below.

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 with masks added to the object memory bank)*

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 is not visible 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 Section 5.3.3, each frame-level object mask is downsampled to match the spatial 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 in LLMs, 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.

5.3.5 Mask Decoder

We adopt SAM 2.1 [210] 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 $L \rightarrow M$ projector (architecturally identical to the $M \rightarrow L$ 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

remaining frames by the same module.

5.3.6 Model Training

The training loss for UniPixel is a linear combination of language modeling loss and mask decoding losses [210], 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 Table 5.1. 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 [88] on the visual encoder and LLM. The model is jointly trained on a large-scale corpus with around 2M samples for diverse tasks.

5.4 Experiments

We evaluate the effectiveness of UniPixel by conducting extensive experiments across a diverse set of benchmarks and tasks. 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?

5.4.1 Implementation Details

We instantiate our base models with 3B and 7B versions of Qwen2.5-VL [15]. Both variants employ pre-trained SAM 2.1 [210] with Hiera Base+ [219] 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 [88] 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.

5.4.2 Training Datasets

The detailed distribution of training datasets for UniPixel is shown in Table 5.1. Within the three-stage training recipe, we first pre-train the sparse prompt encoder using short caption samples from Inst-IT [195] and VideoRefer [319]. 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 ob-

Table 5.1: 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 image & 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 [195]	✓	✓		✓	✓		✓		351K	1	41.2%
	VideoRefer-Short-Caption [319]	✓		✓	✓	✓		✓		500K	1	58.8%
2	RefCOCO [108]	✓	✓					✓	✓	17K	5	20.8%
	RefCOCO+ [108]	✓	✓					✓	✓	17K	5	20.8%
	RefCOCOG [174]	✓	✓					✓	✓	22K	5	26.8%
	RefClef [108]	✓	✓					✓	✓	18K	5	22.0%
	Ref-YouTube-VOS [221]	✓		✓				✓	✓	13K	3	9.5%
3	Osprey-Conversation [318]	✓	✓				✓	✓		1.4K	5	0.1%
	Osprey-Detail-Description [318]	✓	✓				✓	✓		29K	5	2.5%
	Osprey-Pos-Neg [318]	✓	✓				✓	✓		20K	5	1.7%
	VideoRefer-Detailed-Caption [319]	✓		✓			✓	✓		120K	5	10.1%
	VideoRefer-QA [319]	✓		✓			✓	✓		69K	5	5.8%
	Inst-IT-Video-QA [195]	✓		✓			✓	✓		159K	5	13.4%
	VideoRefer-QA-Memory [319]	✓		✓	✓	✓		✓	✓	69K	3	3.5%
	Inst-IT-QA-Memory [195]	✓		✓	✓	✓		✓	✓	158K	3	8.0%
	RefCOCO [108]	✓	✓					✓	✓	17K	10	2.9%
	RefCOCO+ [108]	✓	✓					✓	✓	17K	10	2.9%
	RefCOCOG [174]	✓	✓					✓	✓	22K	10	3.7%
	RefClef [108]	✓	✓					✓	✓	18K	10	3.0%
	ReasonSeg [116]	✓	✓					✓	✓	1.6K	10	0.3%
	ADE20K [342]	✓	✓					✓	✓	20K	3	1.0%
	COCOSuff [22]	✓	✓					✓	✓	118K	3	6.0%
	Mapillary Vistas [182]	✓	✓					✓	✓	18K	3	0.9%
	PACO-LVIS [207]	✓	✓					✓	✓	46K	3	2.3%
	PASCAL-Part [36]	✓	✓					✓	✓	4.4K	3	0.2%
	Ref-YouTube-VOS [221]	✓		✓				✓	✓	13K	5	1.1%
	Ref-DAVIS17 [199]	✓		✓				✓	✓	0.6K	10	0.1%
	Ref-SAV [315]	✓		✓				✓	✓	56K	3	2.8%
	MeVis [54]	✓		✓				✓	✓	23K	5	1.9%
	LV-VIS [252]	✓		✓				✓	✓	11K	3	0.6%
	ViCaS [10]	✓		✓				✓	✓	41K	3	2.1%
	ReVOS [297]	✓		✓				✓	✓	29K	5	2.5%
	GroundMoRe [51]	✓		✓				✓	✓	5.6K	3	0.3%
	LLaVA-1.5-Mix-665K [147]	✓	✓					✓		647K	1	10.9%
	VideoGPT+ Instruct [169]	✓		✓				✓		573K	1	9.7%

ject segmentation datasets [108, 174, 221]. 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 dataset called UniPixel-SFT-1M² to

²<https://huggingface.co/datasets/PolyU-ChenLab/UniPixel-SFT-1M>

Table 5.2: Comparison with state-of-the-art methods on ReVOS [297] val split.

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 [19]	–	29.8	30.2	30.0	20.4	21.5	21.0	25.1	25.9	25.5	5.6
LMPM [54]	–	29.0	39.1	34.1	13.3	24.3	18.8	21.2	31.7	26.4	3.2
ReferFormer [277]	–	31.2	34.3	32.7	21.3	25.6	23.4	26.2	29.9	28.1	8.8
LLM-based Generalists											
LISA [116]	13B	45.2	47.9	46.6	34.3	39.1	36.7	39.8	43.5	41.6	8.6
TrackGPT [231]	13B	48.3	50.6	49.5	38.1	42.9	40.5	43.2	46.8	45.0	12.8
VISA [297]	13B	55.6	59.1	57.4	42.0	46.7	44.3	48.8	52.9	50.9	14.5
HyperSeg [269]	3B	56.0	60.9	58.5	50.2	55.8	53.0	53.1	58.4	55.7	–
InstructSeg [270]	3B	54.8	59.2	57.0	49.2	54.7	51.9	52.0	56.9	54.5	–
GLUS [142]	7B	56.0	60.7	58.3	48.8	53.9	51.4	52.4	57.3	54.9	17.9
ViLLa [341]	6B	–	–	–	–	–	–	54.9	59.1	57.0	–
Sa2VA [315]	4B	–	–	–	–	–	–	–	–	53.2	–
UniPixel (Ours)	3B	<u>62.3</u>	<u>66.7</u>	<u>64.5</u>	<u>57.1</u>	<u>62.1</u>	<u>59.6</u>	<u>59.7</u>	<u>64.4</u>	<u>62.1</u>	<u>19.0</u>
UniPixel (Ours)	7B	63.9	67.8	65.8	59.4	63.7	61.5	61.7	65.7	63.7	19.4

jointly train the model on diverse pixel-level tasks. The original annotations have been rewritten using task-specific templates with instructions. General image and video understanding datasets [147, 169] are also incorporated, resulting around 2M samples in total.

5.4.3 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 [188] scores and MCQ accuracy, respectively. For referring expression comprehension, we leverage mean accuracies, where a predicted bounding box is considered correct when it has the IoU with the ground truth no less than 0.5.

Table 5.3: Comparison on referring video object segmentation (RVOS) and motion-grounded video reasoning datasets, including MeViS [54] (val), Ref-YouTube-VOS [221] (val), Ref-DAVIS17 [199] (val), and GroundMoRe [51] (test).

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													
URVOS [221]	–	25.7	29.9	27.8	45.3	49.2	47.2	47.3	56.0	51.6	–	–	–
MTTR [19]	–	28.8	31.2	30.0	54.0	56.6	55.3	–	–	–	–	–	–
ReferFormer [277]	–	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 [54]	–	34.2	40.2	37.2	–	–	–	–	–	–	12.7	14.0	13.3
OnlineRefer [274]	–	–	–	–	61.6	65.5	63.5	61.6	67.7	64.8	–	–	–
LLM-based Generalists													
PixelLM [217]	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 [116]	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 [297]	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 [16]	3.8B	41.3	47.6	44.4	61.7	65.7	63.7	64.9	72.7	68.8	–	–	–
VideoGLaMM [179]	3.8B	42.1	48.2	45.2	65.4	68.2	66.8	73.3	65.6	69.5	–	–	–
ViLLa [341]	6B	46.5	52.3	49.4	64.6	70.4	67.5	70.6	78.0	74.3	–	–	–
GLUS [142]	7B	48.5	54.2	51.3	65.5	69.0	67.3	–	–	–	–	–	–
Sa2VA [315]	4B	–	–	46.2	–	–	70.0	–	–	73.8	–	–	–
MoRA [51]	7B	–	–	–	–	–	–	–	–	–	27.4	26.9	27.2
UniPixel (Ours)	3B	<u>50.4</u>	<u>55.7</u>	<u>53.1</u>	<u>68.6</u>	<u>72.3</u>	<u>70.5</u>	<u>70.7</u>	<u>77.8</u>	<u>74.2</u>	<u>36.0</u>	<u>38.7</u>	<u>37.4</u>
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

5.4.4 Q1: Comparison with State-of-the-Arts

Reasoning Video Object Segmentation We begin with the most challenging ReVOS [297] 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 Table 5.2. Our 3B variant outperforms all existing methods with larger LLMs (including Sa2VA-4B [315] 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.

Table 5.4: Experimental results on MeViS [54] val^u set. Post means applying post optimization after mask prediction.

Method	Size	FT	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$
LMPM [54]	–	✗	36.5	43.9	40.2
LISA [116]	7B	✗	39.9	46.5	43.2
LISA [116] + XMem [42]	7B	✗	41.9	49.3	45.6
VideoLISA [16]	7B	✗	48.4	54.9	51.7
VideoLISA [16] + Post	7B	✗	50.9	58.1	54.5
Sa2VA [315]	4B	✗	–	–	52.1
Sa2VA [315]	8B	✗	–	–	57.0
UniPixel (Ours)	3B	✗	<u>56.1</u>	<u>63.2</u>	<u>59.7</u>
UniPixel (Ours)	7B	✗	58.4	65.0	61.7

Table 5.5: Performance comparison on Ref-SAV [315] val set. FT means fine-tuning after pre-/co-training.

Method	Size	FT	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$
UniRef++ [278]	–	✗	11.6	9.5	10.5
UNINEXT [296]	–	✗	8.8	6.4	7.6
LMPM [54]	–	✗	12.2	9.8	10.3
VISA [297]	7B	✗	13.2	11.3	11.8
Sa2VA [315]	8B	✗	39.6	43.0	41.3
UniRef++ [278]	–	✓	15.8	13.4	14.6
Sa2VA [315]	8B	✓	48.3	51.7	50.0
UniPixel (Ours)	3B	✗	<u>66.9</u>	<u>67.6</u>	<u>67.2</u>
UniPixel (Ours)	7B	✗	68.5	69.6	69.0

Referring Video Object Segmentation The performance comparisons on MeViS [54], Ref-YouTube-VOS [221], and Ref-DAVIS17 [199] datasets are presented in Table 5.3. 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 [142] by around 3.5%, as well as the similarly sized VideoGLaMM-3.8B [179] by 17%. More experimental results on MeViS [54] val^u set and Ref-SAV [315] val set are provided in Table 5.4 and Table 5.5, 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 [51] dataset (shown in Table 5.3), which highlights visual answer generation that requires joint spatial and temporal grounding. Note that we mainly compare the results with MoRA [51], 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.

Table 5.6: Performance comparison with representative methods on referring expression segmentation (RES) and reasoning segmentation datasets, including RefCOCO/+/g [108, 174] and ReasonSeg [116] (val).

Method	Size	RefCOCO			RefCOCO+			RefCOCOg		ReasonSeg	
		val	testA	testB	val	testA	testB	val(U)	test(U)	gIoU	cIoU
Non-LLM-based Specialists											
ReLA [146]	–	73.8	76.5	70.2	66.0	71.0	57.7	65.0	66.0	–	–
PolyFormer [149]	–	76.9	78.5	74.8	72.2	75.7	66.7	71.2	71.2	–	–
X-Decoder [352]	–	–	–	–	–	–	–	64.6	–	22.6	17.9
SEEM [353]	–	–	–	–	–	–	–	65.7	–	25.5	21.2
LLM-based Image Generalists											
NExT-Chat [323]	7B	74.7	78.9	69.5	65.1	71.9	56.7	67.0	67.0	–	–
LISA [116]	7B	74.9	79.1	72.3	65.1	70.8	58.1	67.9	70.6	61.3	62.9
Groundhog [336]	7B	78.5	79.9	75.7	70.5	75.0	64.9	74.1	74.6	56.2	–
LaSagna [268]	7B	76.8	78.7	73.8	66.4	70.6	60.1	70.6	71.9	48.8	47.2
M ² SA [97]	13B	74.6	77.6	71.0	64.0	68.1	57.6	69.0	69.3	–	–
LLM-based Video Generalists											
VideoLISA [16]	3.8B	73.8	76.6	68.8	63.4	68.8	56.2	68.3	68.8	61.4	67.1
VISA [297]	7B	72.4	75.5	68.1	59.8	64.8	53.1	65.5	66.4	52.7	57.8
Vitron [62]	7B	75.5	79.5	72.2	66.7	72.5	58.0	67.9	68.9	–	–
Sa2VA [315]	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

Referring Expression Segmentation and Reasoning Segmentation Table 5.6 compares the image segmentation capabilities using explicit and implicit queries. We evaluate our co-trained model on RefCOCO/+/g [108, 174] and ReasonSeg [116]. 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. Table 5.7 presents the RES performance after fine-tuning. We follow the common practice that jointly fine-tunes the model on RefCOCO/+/g datasets [108, 174], 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 ex-

Table 5.7: Fine-tuned performance on referring expression segmentation (RES) datasets, including RefCOCO/+g [108, 174].

Method	Size	RefCOCO			RefCOCO+			RefCOCOg	
		val	testA	testB	val	testA	testB	val(U)	test(U)
LISA [116]	7B	74.9	79.1	72.3	65.1	70.8	58.1	67.9	70.6
GSVA [279]	7B	77.2	78.9	73.5	65.9	69.6	59.8	72.7	73.3
OMG-LLaVA [335]	7B	78.0	80.3	74.1	69.1	73.1	63.0	72.9	72.9
GLaMM [209]	7B	79.5	83.2	76.9	72.6	78.7	64.6	74.2	74.9
Sa2VA [315]	4B	80.4	–	–	74.3	–	–	75.7	–
UniPixel (Ours)	3B	<u>81.9</u>	<u>83.5</u>	<u>78.6</u>	<u>75.3</u>	<u>80.3</u>	<u>70.6</u>	<u>77.2</u>	<u>78.5</u>
UniPixel (Ours)	7B	83.0	84.9	80.4	77.8	82.3	72.7	78.7	79.7

Table 5.8: Performance comparison on referring expression comprehension (REC) datasets, including RefCOCO/+g [108, 174].

Method	Size	RefCOCO			RefCOCO+			RefCOCOg	
		val	testA	testB	val	testA	testB	val(U)	test(U)
OFA [256]	–	80.0	83.7	76.4	68.3	76.0	61.8	67.6	67.6
Shikra [31]	7B	87.0	90.6	80.2	81.6	87.4	72.1	82.3	82.2
MiniGPT-v2 [30]	7B	88.7	91.6	85.3	79.9	85.1	74.4	84.4	84.6
Vitron [62]	7B	90.9	93.2	89.3	83.7	89.1	76.9	86.4	87.0
UniPixel (Ours)	3B	<u>91.8</u>	<u>93.8</u>	87.5	<u>86.3</u>	<u>90.8</u>	<u>80.3</u>	<u>88.0</u>	<u>88.2</u>
UniPixel (Ours)	7B	92.0	94.4	<u>88.1</u>	87.2	91.9	82.1	88.6	88.7

pression comprehension by inferring the bounding boxes from predicted masks. Its performance (accuracy with $\text{IoU} \geq 0.5$) is compared with representative methods in Table 5.8. 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 [319], which contains two subsets for description and question answering tasks. The comparisons are in Table 5.9 and Table 5.10. 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-

Table 5.9: Comparison with state-of-the-art methods on VideoRefer-Bench^D [319].

Method	Size	Single-Frame					Multi-Frame				
		SC	AD	TD	HD	Avg.	SC	AD	TD	HD	Avg.
General LMMs											
Qwen2-VL [255]	7B	2.97	2.24	2.03	2.31	2.39	3.30	2.54	2.22	2.12	2.55
InternVL2 [190]	26B	3.55	2.99	2.57	2.25	2.84	<u>4.08</u>	3.35	3.08	2.28	3.20
GPT-4o-mini [188]	–	3.56	2.85	2.87	2.38	2.92	<u>3.89</u>	3.18	2.62	2.50	3.05
GPT-4o [188]	–	3.34	2.96	3.01	2.50	2.95	4.15	<u>3.31</u>	<u>3.11</u>	2.43	3.25
Image Referring LMMs											
Ferret [311]	7B	3.08	2.01	1.54	2.14	2.19	3.20	2.38	1.97	1.38	2.23
Osprey [318]	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 [251]	7B	2.35	0.30	0.02	<u>3.59</u>	1.57	–	–	–	–	–
Artemis [203]	7B	–	–	–	–	–	3.42	1.34	1.39	2.90	2.26
VideoRefer [319]	7B	4.41	3.27	<u>3.03</u>	2.97	3.42	4.44	3.27	3.10	3.04	3.46
UniPixel (Ours)	3B	<u>4.04</u>	<u>3.15</u>	3.10	3.37	3.42	<u>4.08</u>	3.13	3.13	<u>3.42</u>	<u>3.44</u>
UniPixel (Ours)	7B	3.83	3.07	2.96	3.62	<u>3.37</u>	3.82	3.05	3.01	3.57	3.36

frame modes denote applying the masks only on a specific frame and on all frames, respectively. UniPixel can effectively comprehend both types of prompts, and accurately respond with object-centric descriptions or answers.

General Video Question Answering We also evaluate UniPixel on MVBench [126] in Table 5.12 to compare its general video understanding capabilities with existing methods. 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.

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

We design the new PixelQA task based on VideoRefer-Bench^Q [319], where the original mask prompts are replaced with point or box prompts. Given these ambiguous visual cues, models are expected to identify the target object according to

Table 5.10: Comparison with state-of-the-art methods on VideoRefer-Bench^Q [319] (*mask prompts*). MF denotes multi-frame mode.

Method	Size	MF	BQ	SQ	RQ	CQ	FP	Avg.
<i>General LMMs</i>								
Qwen2-VL [255]	7B	✗	62.0	69.6	54.9	87.3	74.6	66.0
InternVL2 [190]	26B	✗	58.5	63.5	53.4	88.0	78.9	65.0
GPT-4o-mini [188]	–	✗	57.6	67.1	56.5	85.9	75.4	65.8
GPT-4o [188]	–	✗	62.3	74.5	66.0	88.0	73.7	71.3
<i>Image Referring LMMs</i>								
Ferret [311]	7B	✗	35.2	44.7	41.9	70.4	74.6	48.8
Osprey [318]	7B	✗	45.9	47.1	30.0	48.6	23.7	39.9
<i>Video Referring LMMs</i>								
VideoRefer [319]	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 [319]	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 5.11: Evaluation results on PixelQA. All the visual prompts are applied in a single frame.

Method	Size	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	Acc
<i>Point Prompts</i>					
InternVL2 [190]	26B	–	–	–	60.8
Qwen2-VL [255]	72B	–	–	–	69.3
UniPixel (Ours)	2B	45.6	50.7	48.1	68.2
UniPixel (Ours)	7B	43.7	48.8	46.2	70.0
<i>Box Prompts</i>					
InternVL2 [190]	26B	–	–	–	61.3
Qwen2-VL [255]	72B	–	–	–	69.0
UniPixel (Ours)	2B	48.1	53.6	50.9	68.7
UniPixel (Ours)	7B	43.4	48.4	45.9	69.6
<i>Mixed (50% Points + 50% Boxes)</i>					
InternVL2 [190]	26B	–	–	–	60.9
Qwen2-VL [255]	72B	–	–	–	69.1
UniPixel (Ours)	2B	46.4	51.7	49.0	68.5
UniPixel (Ours)	7B	43.6	48.6	46.1	69.9

the question and the visual prompt, then respond with **both the textual answer and the object masks**. We report the mask prediction $\mathcal{J}\&\mathcal{F}$ and MCQ accuracy in Table 5.11. Note that none of the existing methods supports this scenario. Thus, we apply set-of-mark prompts [302] directly on video frames, and evaluate the QA accuracies of two strong LMMs [255, 190] 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.

5.4.6 Q3: Ablation Studies

Effect of Task Unification We study the effect of task unification in Table 5.13 (a). Unifying referring and segmentation capabilities into a single model and training

Table 5.12: Comparison with representative methods on MVBench [126]. 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 [187]	–	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 [171]	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 [328]	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 [125]	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 [135]	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 [216]	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 [291]	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 [151]	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+ [169]	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 [126]	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 5.13: Key ablation studies with UniPixel-2B on PixelQA (mixed).

(a) Task Unification					(b) Object Memory Bank					(c) Encoder & Decoder				
Refer	Segment	Memory	$\mathcal{J}\&\mathcal{F}$	Acc	Referring Method	$\mathcal{J}\&\mathcal{F}$	Acc	Encoder	Decoder	$\mathcal{J}\&\mathcal{F}$	Acc			
✓			–	64.6	① <REF>	46.8	64.5	w/o Time	–	44.3	63.7			
	✓		47.5	–	② <REF><SEG>	47.8	64.9	w/ Time	–	49.0	68.5			
✓	✓		48.2	67.4	③ <REF><SEG> + Pooling	47.5	66.3							
✓	✓	✓	49.0	68.5	④ Object Memory Bank	49.0	68.5	–	Independent	46.1	66.2			
								–	Propagation	49.0	68.5			

them jointly leads to better results 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 Table 5.13 (b) verifies the effectiveness of object memory bank. ① means using a single token for each referred object. ② means adding segmentation 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 further boosts the performance.

Design Space of Prompt Encoder & Mask Decoder We compare different

Table 5.14: Effectiveness justification of multi-stage training. The three-stage training recipe leads to the 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	<u>63.6</u>	61.0	54.8	61.9	58.4	71.1	71.5
		✓	59.0	63.4	61.2	55.2	62.1	58.7	<u>71.8</u>	<u>72.3</u>
	✓	✓	<u>59.6</u>	63.5	<u>61.6</u>	<u>55.7</u>	<u>62.5</u>	<u>59.1</u>	71.2	71.6
✓	✓	✓	59.7	64.4	62.1	56.1	63.2	59.7	72.2	72.8

Table 5.15: Ablation study on the number of hidden tokens for each <SEG> token. 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	<u>59.7</u>
4	<u>59.8</u>	63.9	<u>61.9</u>	56.8	<u>63.1</u>	59.9
8	59.5	<u>64.0</u>	61.8	<u>56.4</u>	<u>62.8</u>	<u>59.6</u>

Table 5.16: 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	<u>72.6</u>	<u>68.2</u>
✓	✓	<u>72.2</u>	72.8	68.5

prompt encoder and mask decoder designs in Table 5.13 (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.

Effect of Multi-stage Training We investigate the effectiveness of multi-stage training in Table 5.14. As shown in the first line, directly training the model using large-scale data only leads to sub-optimal performance, due to the unaligned

Table 5.17: Ablation study on training data used in stage 3. 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	<u>63.2</u>	59.6	–	–
✓	✓			59.2	63.7	61.5	55.8	63.1	59.5	<u>72.3</u>	<u>72.6</u>
✓	✓	✓		<u>59.6</u>	<u>64.5</u>	<u>62.1</u>	<u>56.3</u>	<u>63.5</u>	<u>59.9</u>	<u>72.4</u>	72.5
✓	✓	✓	✓	<u>59.7</u>	<u>64.4</u>	<u>62.1</u>	<u>56.1</u>	<u>63.2</u>	<u>59.7</u>	72.2	<u>72.8</u>

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 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 Table 5.15. 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 Table 5.16. We investigated two strategies: 1) re-using the weights and 2) adding

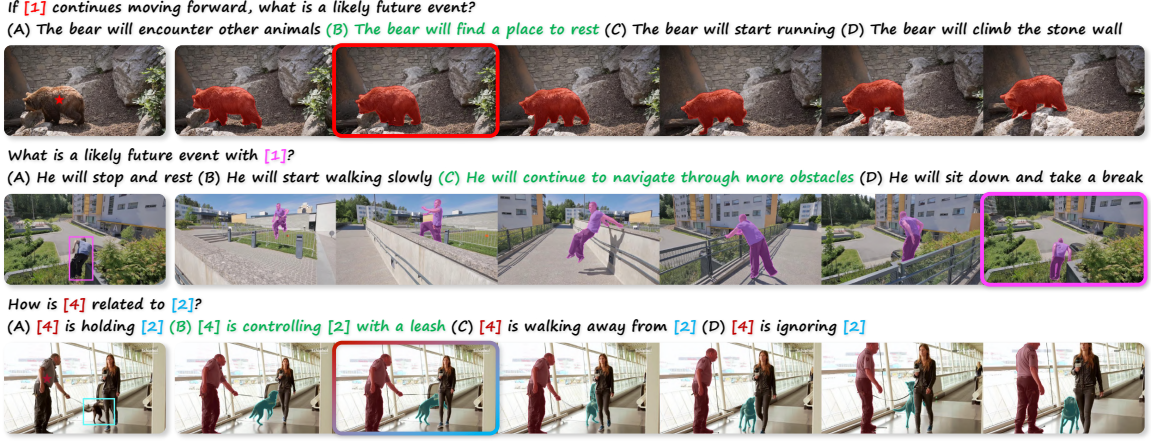


Figure 5.5: Visualization of UniPixel-2B’s outputs on PixelQA task. In the left-most frame, star markers and boxes represent point and box prompts, respectively. The frames with highlighted borders are those where visual prompts are applied.

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.

Combination of Training Data Table 5.17 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 base model, while it slightly affects the performance on pixel-level tasks.

5.4.7 Visualizations

Visualizations of outputs from UniPixel-2B on PixelQA are shown in Figure 5.5. Figure 5.6~5.11 present more qualitative results on different pixel-level understanding tasks. Our method can effectively handle flexible visual prompts [319], implicit queries [116, 297], long queries [315], and motion-grounded questions [51].

5.5 Summary

In this chapter, we further study the topic of fine-grained spatial-temporal reasoning via multi-modal large language models. Specifically, we address this problem from two aspects: model and evaluation. We have demonstrated that through appropriate model architectural design, fine-grained pixel-level perception and high-level reasoning could be effectively integrated into a unified framework. Moreover, we also present more flexible ways of interactions between human and machine – through explicit visual prompts such as points, boxes, and masks. Our detailed contributions in this chapter are summarized as follows:

1. A unified large multi-modal model (UniPixel) that supports flexible pixel-level object referring and segmentation in images and videos. It can also integrate such fine-grained grounding capabilities with its internal reasoning.
2. An object memory bank module storing the intermediate information of key regions or objects. This enables the model to re-examine the regions/objects after segmentation for further reasoning.
3. A new pixel-level video understanding task (PixelQA) that requires visual prompt understanding, object segmentation and tracking, and video question answering capabilities.

We also list the technical novelties in this chapter below:

1. To effectively unify object referring and segmentation, the novel design of object memory bank allows the model to crop and focus more on the user referred regions/objects, largely enhancing the model’s fine-grained video understanding capabilities.
2. In the visual prompt encoder, we carefully designed the joint spatial and temporal encoding scheme for visual prompts. This novelty ensures the model to accurately comprehend the spatial-temporal position of the reference.

3. The proposed PixelQA benchmark is a novel combination of object referring, object segmentation, and video question answering tasks, serving as the foundation for future pixel-level MLLM research.

5.6 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 [209] or videos [179], 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. To addresses these limitations, potential future works shall focus on exploring more pixel-level understanding tasks and more flexible mechanisms for the mask decoder.

Chapter 5. Unified Referring and Segmentation for Pixel-Level Visual Reasoning

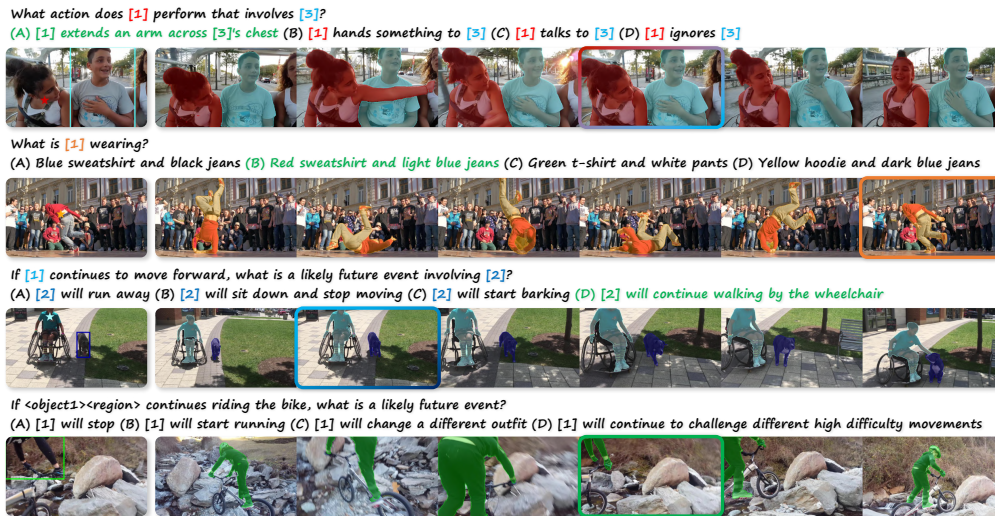


Figure 5.6: Visualization of the outputs from UniPixel on PixelQA.



Figure 5.7: Visualization of the outputs from UniPixel on ReVOS [297].



Figure 5.8: Visualization of the outputs from UniPixel on Ref-DAVIS17 [199].

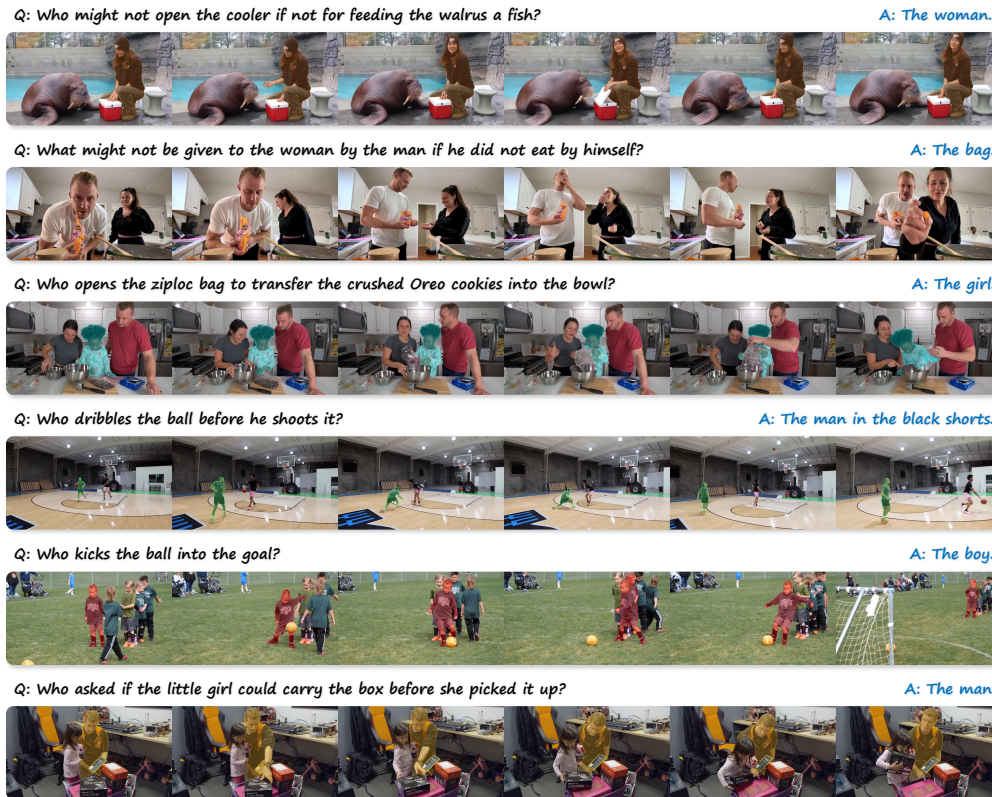
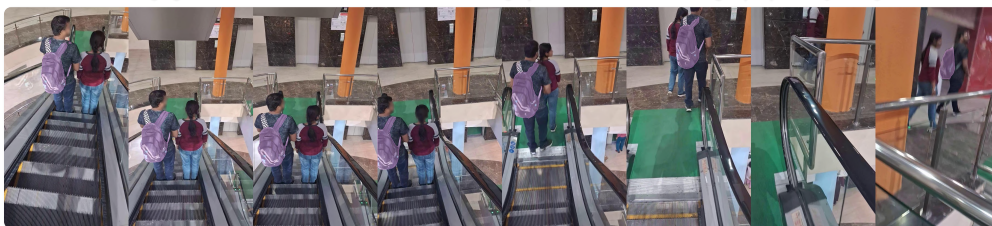


Figure 5.9: Visualization of the outputs from UniPixel on GroundMoRe [51].

Chapter 5. Unified Referring and Segmentation for Pixel-Level Visual Reasoning

Find the object according to the description: The object is a dark-colored backpack with light-colored accents, featuring multiple compartments and pockets, securely fastened to an individual's back. The person is dressed in dark clothing and ascending an escalator in a public setting, likely a mall or transportation hub. The backpack has adjustable straps and a top handle, appearing functional for carrying various items. The individual moves steadily up the escalator, indicating a purposeful journey.



Analyze the following sentences and provide the corresponding segmentation mask: The object is a dark-colored sedan, likely blue or black, parked on an unpaved surface, possibly a dirt road or an area with loose soil. It has four doors, a visible rear spoiler on the trunk, silver wheels, and tinted windows. The car is slightly tilted, suggesting it might be parked on uneven ground or experiencing some form of imbalance. Throughout the video, the sedan remains stationary, with no indication of movement or actions being performed by the vehicle.



Please segment the object according to the description: The object is a person with long dark hair, wearing a dark top and a patterned skirt with geometric designs. This individual is stationary or moving very slowly in the background of a retail store, possibly a furniture or home goods store. The person remains in close proximity to another shopper pushing a shopping cart, suggesting they might be together or interacting. The scene captures a typical shopping experience.



Figure 5.10: Visualization of the outputs from UniPixel on Ref-SAV [315].

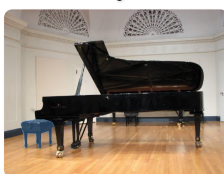
Find the lens that is more suitable for photographing nearby objects.



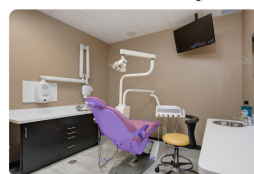
Where is the goat nearest to the bottom stone? Give me the segmentation mask.



Please localize the place where piano players should sit in this image.



Segment the place where the patient lies down to receive examination in this image.



Which part of the vehicle must be used to display identifying information as required by law? Segment the target directly.



What item in the picture can provide information to help guide travelers through this rugged terrain that can be challenging to navigate?



Where is the place where the garbage should be put? Please respond with the segmentation mask.



In some rural areas, horse-drawn carts are still used for transportation and carrying goods. What is the main source of power that drives the cart in the picture?



Figure 5.11: Visualization of the outputs from UniPixel on ReasonSeg [116].

Part III

Interaction

We must perceive in order to move, but we must also move in order to perceive.

— J. J. Gibson, *The Ecological Approach to Visual Perception*

Chapter 6

A Chain-of-LoRA Agent for Interactive Video Understanding

Videos, with their unique temporal dimension, demand precise grounded understanding, where answers are directly linked to visual, interpretable evidence. Despite significant breakthroughs in text-based reasoning with large language models, multi-modal reasoning – especially for videos – remains limited. In this chapter, we fill this gap by introducing **VideoMind**, a novel video-language agent for temporal-grounded video reasoning. Our method involves two key innovations: (1) We identify four essential capabilities for grounded video reasoning and propose a role-based agentic workflow, comprising a planner to coordinate roles, a grounder for temporal event localization, a verifier to assess event candidates, and an answerer for question answering. (2) To efficiently integrate these roles during inference, we propose a novel **Chain-of-LoRA** mechanism, where a unified base model with multiple LoRA adapters is leveraged to enable seamless role switching, balancing efficiency and flexibility. Extensive experiments on 14 benchmarks across 3 tasks demonstrate the effectiveness of the proposed scheme in advancing video agent, test-time scaling, and long-form video reasoning.

6.1 Introduction

Recent advancements in large language models (LLMs) have demonstrated remarkable success in text-based reasoning [272, 304, 222], significantly improving both accuracy and interpretability in complex problem-solving scenarios [305]. Following these breakthroughs, efforts have been devoted to extending these reasoning capabilities to multi-modal domains [337, 289, 243] such as vision-centric science [163] and math [168] understanding.

Among multi-modal signals, videos pose a unique challenge due to their temporal dimension, introducing complexities absent in images or text. Effective video reasoning requires not only recognizing visual appearances but also understanding how they evolve over time [281, 28, 158, 276]. While recent visual Chain-of-Thought (CoT) methods [337, 289, 243] excel at generating detailed thoughts for static images, they struggle with long videos as they cannot explicitly localize or revisit earlier parts of the sequence, as presented in Figure 6.1 (left). Humans, by contrast, can reason over long videos with ease [284, 283, 285]: they break down complex problems, identify relevant moments, revisit them to confirm details, and synthesize their observations into coherent answers. This natural proficiency motivates the development of an AI agent that emulates this process – flexibly coordinating multiple capabilities to achieve advanced, vision-centric reasoning.

In this work, we introduce **VideoMind**, a video-language agent with enhanced temporal-grounded reasoning capabilities. To meet the demands of diverse tasks, we define four essential roles for understanding complex long-form videos: (1) a **planner** to decompose tasks and coordinate other roles, (2) a **grounder** for precise moment localization, (3) a **verifier** for moment candidates assessment, and (4) an **answerer** for moment-aware response generation. Each role is carefully designed to deliver strong performance, for example, the grounder is equipped with a

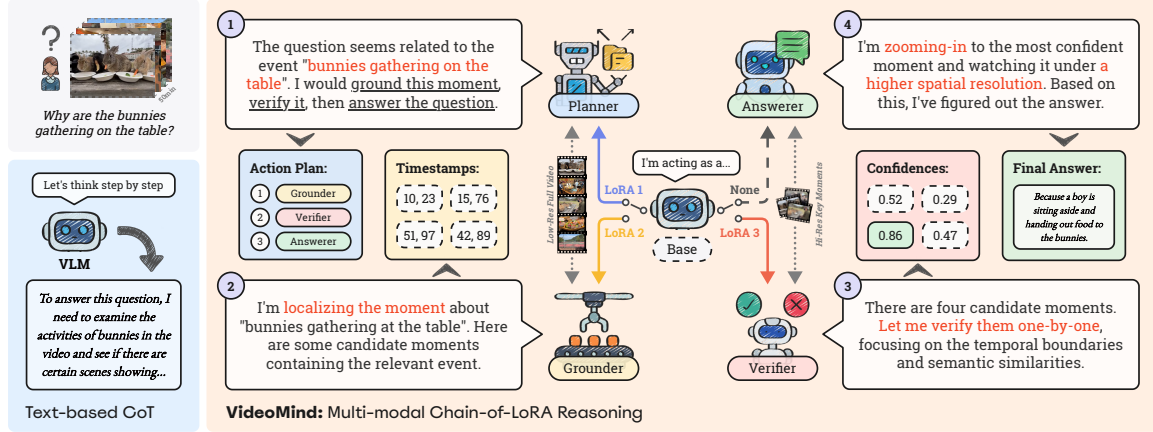


Figure 6.1: Illustration of VideoMind’s Chain-of-LoRA reasoning mechanism. The problem is decomposed by the planner and distributed to grounder, verifier, and answerer to systematically localize, verify, and interpret the relevant video moments.

timestamp decoder to ensure accurate temporal grounding. To enable efficient integration of these roles, we also propose a novel **Chain-of-LoRA** mechanism, where all the roles are implemented based on the same base LMM but with different LoRA adapters [88]. Therefore, role-specific capabilities can be trained separately on tailored datasets. During inference, all the LoRA parameters are cached into the memory, so that each role could be activated by simply switching to the corresponding LoRA, as shown in Figure 6.1 (right). This approach reflects a minimalist yet flexible design philosophy, facilitating seamless transitions and interactions among roles without incurring the memory overhead of maintaining multiple full models. As a result, VideoMind achieves both efficiency and flexibility on diverse tasks.

We conduct extensive experiments on 14 public benchmarks, including 3 on Grounded VideoQA, 6 on Video Temporal Grounding, and 5 on General VideoQA, to evaluate the effectiveness of our approach. VideoMind exhibits strong adaptability in addressing diverse reasoning tasks by jointly providing accurate responses and temporal-grounded evidence. Notably, our 2B model surpasses GPT-4o [188] and Gemini-1.5-Pro [213] on several long video benchmarks such as CG-Bench [28],

MILVU [343], and IVBench [258]. State-of-the-art performance is also achieved on temporal grounding datasets including QVHighlights [120] and Charades-STA [70]. We further conduct ablation studies to justify our design choices, particularly the Chain-of-LoRA mechanism for enhancing flexibility while preserving efficiency. Our contributions are summarized as follows:

1. We propose **VideoMind**, a multi-modal agentic framework that enhances video reasoning by emulating human cognitive processes, including task decomposition, moment localization and verification, and answer synthesis. It addresses the unique challenges of long video reasoning in a progressive and structured manner.
2. We also introduce **Chain-of-LoRA**, an efficient test-time scaling mechanism that enables a single model to seamlessly switch among multiple roles. This approach enhances VideoMind’s flexibility without incurring additional memory overhead.
3. Our method demonstrates strong performance across three scenarios, including Grounded VideoQA, Video Temporal Grounding, and General VideoQA. Notably, VideoMind-2B outperforms GPT-4o and Gemini-1.5-Pro on several long video benchmarks.

6.2 Related Work

6.2.1 Temporal-grounded Video Understanding

Significant advances in video understanding have propelled tasks such as video captioning [340, 141], video question answering [280, 324], and video-text retrieval [176, 139]. However, these models often lack *visually grounded correspondence* and interpretability, particularly for long-form videos. The task of Video Tempo-

ral Grounding [70, 114] tackles this issue by requiring precise temporal localization for diverse queries, though regression-based models [155, 153] excel at localization but fall short in providing textual interpretability. Recent benchmarks [281, 28, 158] intensify this challenge, demanding both reasoning for complex questions and fine-grained temporal correspondence. Previous baselines for these tasks typically rely on multi-task objectives or modular agents composed of distinct components [313, 261, 60], often yielding sub-optimal performance or overly complex systems, which constrain their efficiency and flexibility. Our VideoMind is an agentic workflow built upon a unified LMM, seamlessly integrating multiple functionalities while enhancing localization and interpretability, thus surpassing the limitations of prior methods.

6.2.2 Multi-modal Reasoning

Large Multi-modal Models [148, 122, 157] exhibit generalized capabilities such as free-form question answering. However, they fall short in addressing complex challenges that often require reasoning [272]. One approach to overcome this is to develop agent-based interfaces [324, 107], which integrates textual outputs from visual tools to enable reasoning via LLMs. Advanced methods [236, 303, 69] invoke visual APIs (e.g., detectors and captioners) through progressive execution and reasoning. Alternatively, pure text-based reasoning [189, 79] has been a dominant paradigm in LLMs, exemplified by training with long CoT processes using reinforcement learning, which provides detailed step-by-step reasoning, with some works [337, 289] extending this mechanism to the visual domain. Despite these advances, extending reasoning to videos remains an open challenge. Given the long-context nature of informative videos, we believe that a *vision-centric* CoT should incorporate human-like re-watching, self-validation, and state-switching strategies [282, 286, 285], leading us to introduce the novel Chain-of-LoRA framework.

6.2.3 Inference-time Searching

Inference-time searching has emerged as a critical technique for tackling complex reasoning and planning challenges in domains like robotics [264], games [226], and navigation [242]. The advent of OpenAI o1 [189] has advanced these inference-time techniques within LLMs by integrating sampling strategies such as controlled decoding [25, 295], Best-of-N sampling [134], and Monte Carlo Tree Search (MCTS) [263, 327, 250], allowing LLMs to iteratively refine outputs and achieve superior performance without altering their underlying weights. However, the potential of inference-time searching remains largely untapped in video understanding, where temporal reasoning introduces unique challenges. In our framework, we explore how such a strategy can be tailored for video temporal reasoning, observing that models are highly sensitive to the selection of temporal segments, often producing unreliable predictions when segment choices are sub-optimal. To address this, we propose a *moment-level* searching approach where a grounder generates multiple candidates, followed by a verifier that evaluates and determines the correct correspondence. This strategy significantly enhances temporal grounding accuracy and robustness across diverse scenarios.

6.3 Method

6.3.1 Overview

Figure 6.2 provides an overview of VideoMind. Our model derives from the Qwen2-VL [255] architecture, consisting of an LLM backbone and a ViT-based visual encoder support dynamic resolution inputs. Given a video input $\mathcal{V}$ and a text query $\mathcal{Q}$, the model performs step-by-step reasoning by adaptively calling different roles: (1) **Planner**: Dynamically coordinates the following roles based on the query. (2)

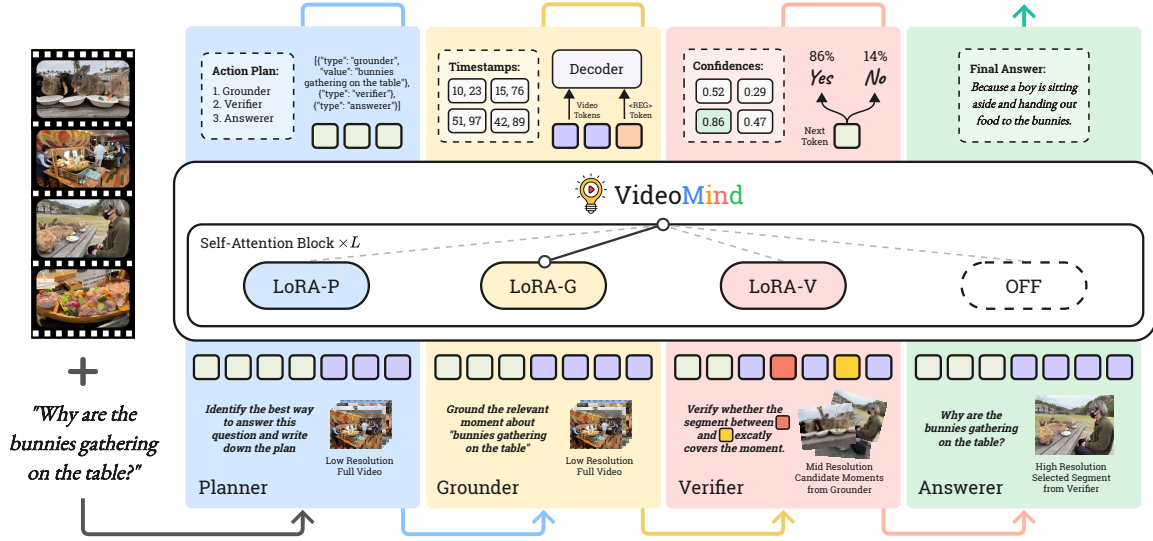


Figure 6.2: The overall workflow of VideoMind. Given a video and a query, it adaptively activates different roles (e.g., Planner → Grounder → Verifier → Answerer in this case) and performs step-by-step reasoning by calling individual modules.

Grounder: Identifies and localizes relevant video moments. (3) **Verifier:** Evaluates the validity of the moments identified by the grounder, refining them through a zoom-in process with boolean outputs. (4) **Answerer:** Generates the final response in natural language. This mechanism enables the models to **revisit the videos several times** (with varying temporal segments & spatial resolutions) to derive the final response.

6.3.2 Planner

An agent must be flexible enough to handle diverse tasks and efficiently determine which functions (roles) to call. To achieve this, we design the planner, which dynamically coordinates all the other roles for each query. It decides the sequence of function calls based on the multi-modal context. We utilize a JSON-style object `{"type": "<role>", "value": "<argument>"}` to denote a function call. In this way, a sequence of roles can be succinctly represented as a list of such objects. Three

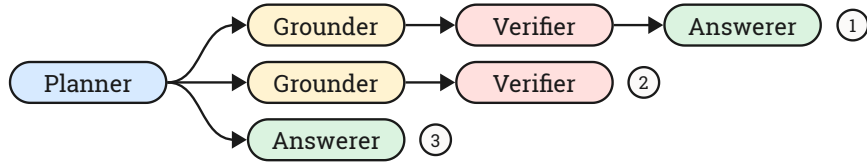


Figure 6.3: Planner coordinates all the other roles based on the video and query context, offering three reasoning plans and a query rephrasing mechanism to address diverse demands.

reasoning plans for different tasks are pre-defined and illustrated in Figure 6.3.

(1) Grounding & Verifying & Answering: This plan requires the agent to generate both a textual response and a corresponding temporal moment. For example, in Grounded VideoQA scenarios [280], to answer the question “*What is the boy doing when the baby is crying?*”, the agent should identify the moment of “*baby is crying*”, and then investigate the boy’s activity.

(2) Grounding & Verifying: This plan is designed for grounding-only tasks such as moment retrieval [120, 70]. For questions like “*When does the woman go downstairs?*”), the model should provide precise timestamps directly as the answer. Since the grounding results could potentially be unreliable, an extra zoom-in verification step is necessary.

(3) Answering Only: If the question is straightforward (e.g., “*Summarize this video*”) or the video is very short, it could be unnecessary to perform grounding. Instead, the model should watch the entire video and answer directly.

Query Rephrasing When the user query lacks sufficient detail for accurate moment localization, the planner is allowed to **rephrase** the question into a more descriptive version. For instance, the question “*What is the person sitting on the bed doing as the baby plays?*” may confuse the grounder as it contains multiple events (“*person sitting on the bed*” and “*baby plays*”). It can be rephrased to “*the baby is playing*” as an accurate scene description.

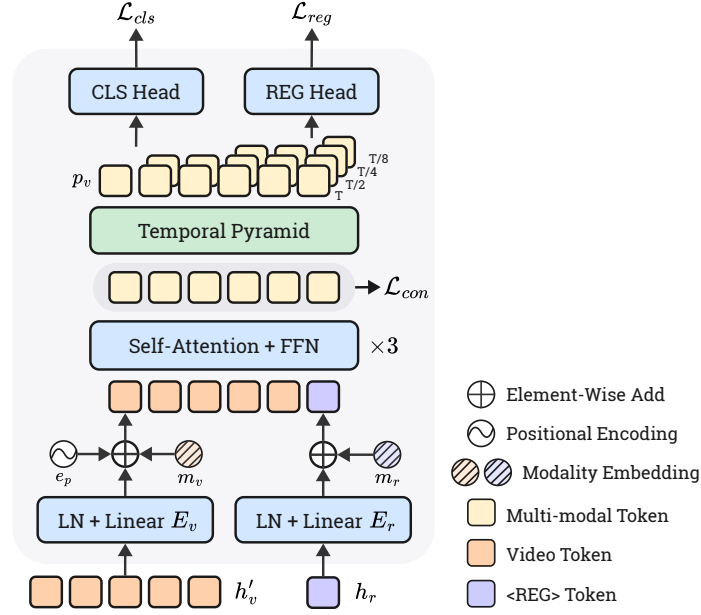


Figure 6.4: Detailed architecture of the timestamp decoder. This module accepts hidden states of both the frame tokens and the <REG> token, and decodes them into the corresponding start and end timestamps.

To train the planning and query rephrasing capabilities, we curated a dataset of 39K samples (shown in Table 6.1) from public benchmarks. For planning, we aligned each reasoning plan with corresponding question types: *temporal* questions from NExT-QA [280] are assigned to Plan-1, moment queries from QVHighlights [120] are for Plan-2, and *causal* & *descriptive* questions from NExT-QA [280] are for Plan-3. For query rephrasing, we leverage GPT-4o mini [188] to generate synthetic `video + question` $\rightarrow$ `query` samples.

6.3.3 Grounder

The grounder aims to localize relevant moments (*i.e.*, predicting start and end timestamps) based on text queries, thereby supporting the reasoning process by identifying visual cues. This requirement calls for the development of an LMM with robust temporal grounding capabilities.

Timestamp Decoder Instead of directly predicting timestamps through language modeling [216] or special tokens [89, 158], we develop a timestamp decoder to maximize the LMM-based grounding performance. Specifically, we introduce a $\langle \text{REG} \rangle$ token to facilitate this process. When the $\langle \text{REG} \rangle$ token is generated, the last-layer hidden states of it and all the visual tokens will be sent into the decoder for timestamp prediction, obtaining a tuple $[t_{start}, t_{end}]$ representing the normalized start and end timestamps.

As shown in Figure 6.4, the decoder accepts the hidden states of the visual tokens $\mathbf{h}_v \in \mathbb{R}^{(T \times H \times W) \times D_L}$ and the $\langle \text{REG} \rangle$ token $\mathbf{h}_r \in \mathbb{R}^{1 \times D_L}$ as inputs, where T, H, W, D_L are the downsampled number of frames, height, width, and hidden dimensions of the LLM, respectively. We apply a 1D average pooling with kernel size and stride equal to $H \times W$ to compress the visual tokens to one token per frame.

$$\mathbf{h}'_v = \text{AvgPool}(\mathbf{h}_v) \in \mathbb{R}^{T \times D_L} \quad (6.1)$$

Then, $\mathbf{h}'_v$ and $\mathbf{h}_r$ are projected by two linear layers E_v and E_r to reduce the hidden dimension to D .

$$\mathbf{e}_v = E_v(\mathbf{h}'_v) \in \mathbb{R}^{T \times D}, \quad \mathbf{e}_r = E_r(\mathbf{h}_r) \in \mathbb{R}^{1 \times D} \quad (6.2)$$

The resulting $\mathbf{e}_v$ and $\mathbf{e}_r$ serve as consolidated representations of the video frames and the query¹, respectively. To effectively integrate their information, we concatenate them along the sequence dimension and send them into a three-layer transformer encoder [248].

$$[\mathbf{e}'_v; \mathbf{e}'_r] = \text{Transformer}([\mathbf{e}_v + \mathbf{m}_v + \mathbf{e}_p; \mathbf{h}_r + \mathbf{m}_r]) \quad (6.3)$$

Here, modality indicators $m_v \in \mathbb{R}^{1 \times D}$ and $m_r \in \mathbb{R}^{1 \times D}$ are randomly initialized learnable embeddings. m_v is expanded to $T \times D$ before being added with e_v . e_p is a normalized sinusoidal positional encoding [248] for preserving temporal aware-

¹We use the term “query” to denote the features of $\langle \text{REG} \rangle$ token.

ness. The output sequence is split back into e'_v and e'_r , indicating the contextualized frame and query embeddings, respectively.

Temporal Feature Pyramid To improve the model’s adaptability to videos and moments of varying lengths, we map e'_v into a four-level temporal feature pyramid [153, 325]. Each level is produced by a $\text{Conv1D} \rightarrow \text{LayerNorm} \rightarrow \text{SiLU}$ block, where the Conv1D employs a kernel size and stride of 2. Therefore, the resulting four levels retain 1, 1/2, 1/4, and 1/8 of the original sequence length, respectively. To accelerate the prediction, we concatenate the sequences from all pyramid levels along the temporal dimension to form p_v with length $L = T + T/2 + T/4 + T/8$, allowing parallelized prediction across temporal resolutions.

Prediction Heads We introduce two heads for timestamps prediction: **(1) A classification head** is designed for frame-level foreground-background classification. This is instantiated by a two-layer Conv1D module with kernel size 3 and padding 1, followed by a Sigmoid activation. The outputs are frame-level confidence scores $\{\hat{c}_i\}_{i=0}^L$ indicating whether each frame falls inside the desired moment. A binary focal loss [143] is utilized to optimize these scores.

$$\mathcal{L}_{cls} = -\lambda_{cls}\alpha(1 - \hat{c}_i)^\gamma \log(\hat{c}_i) \quad (6.4)$$

Here, $\alpha = 0.9$ and $\gamma = 2.0$ are hyperparameters of the focal loss, and λ_{cls} is the loss reweighing term. **(2) A boundary regression head** is adopted to predict the frame-level temporal offsets for start and end boundaries $\{[\hat{b}_i^s, \hat{b}_i^e]\}_{i=0}^L$. This is also a two-layer Conv1D block (with 2 output channels), followed by an exponential activation. Predictions from different pyramid levels are further modulated by different learnable scaling factors. These outputs are supervised by an $L1$ loss.

$$\mathcal{L}_{reg} = \lambda_{reg}(|b_i^s - \hat{b}_i^s| + |b_i^e - \hat{b}_i^e|) \quad (6.5)$$

In order to realize better alignment between e'_v and e'_r , we incorporate an additional contrastive loss to encourage learning more discriminative representations. Specif-

Table 6.1: Training datasets for different roles. Source datasets were repurposed for training planner and verifier. *mr* and *step* denote the moment retrieval and step localization subsets, respectively.

Role	#Samples	Datasets
Planner	39K	NeXT-QA-Plan (34K), QVHighlights-Plan (5K)
Grounder	210K	QVHighlights (5K), DiDeMo (33K), TACoS (9K), InternVid-VTime (54K), CosMo-Cap (87K), QuerYD (19K), HiREST _{mr} (8K), HiREST _{step} (4K)
Verifier	232K	DiDeMo-Verify (165K), TACoS-Verify (43K), QVHighlights-Verify (24K)

ically, we calculate the cosine similarities among all frame-query pairs (denoted as $\{s_i\}_{i=0}^L$), then sample a positive frame (falling within the ground truth boundary) and apply the following optimization objective:

$$\mathcal{L}_{con} = -\lambda_{con} \log \frac{\exp(s_p/\tau)}{\exp(s_p/\tau) + \sum_{i \in \Theta} \exp(s_i/\tau)} \quad (6.6)$$

Here, Θ is the set of frame indices with $s_p > s_i$, and $\tau = 0.07$ is the temperature parameter. The final loss for the timestamp decoder is the sum of these losses at all layers with $\lambda_{cls} = 5.0$, $\lambda_{reg} = 1.0$, and $\lambda_{con} = 0.05$. The training datasets for the grounder are listed in Table 6.1.

6.3.4 Verifier

Key moments are crucial for providing visual cues, yet they might be unreliable due to grounding errors. Thus, further verifications are necessary. We let the grounder generate top-5 predictions, then employ the verifier to select the most reliable one. This process is presented in Figure 6.5.

Recap by Zooming-in For each candidate moment, we apply a zoom-in strategy by expanding the boundaries by 50% on both sides and temporally cropping the enlarged segment. The resulting segment and the original text query are sent to the verifier to assess whether the queried event exactly occurs within the tem-

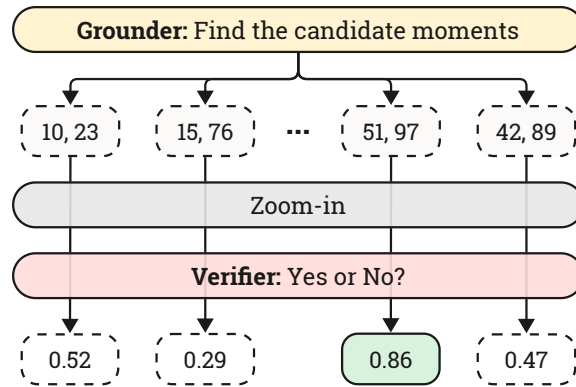


Figure 6.5: The grounder generates multiple candidate moments, which are then refined by the verifier via **zooming-in** to select the best moment.

poral boundaries. To enhance boundary awareness, we adopt two special tokens, <SEG-START> and <SEG-END>, to explicitly mark the beginning and end of the moment. These tokens are inserted among the visual tokens at the corresponding frames, effectively guiding the model in recognizing moment boundaries.

Boolean Judgement The verifier’s responses are binary, *i.e.*, either “Yes” or “No”. To train this role, we sample predictions from the grounder and assign binary labels based on an IoU threshold of 0.5. The model is then fine-tuned via SFT to predict these labels. During inference, for each candidate moment, we employ teacher forcing to obtain the likelihoods of the <Yes> and <No> tokens, denoted as L_y and L_n , respectively. The confidence score is then computed as $\text{Sigmoid}(L_y - L_n)$. The moment with the highest score is selected and passed to the answerer.

6.3.5 Answerer

The answerer responds to the given question based on the cropped video segment (w/ grounder) or the whole video (w/o grounder). Since the objective of this role is strictly aligned with existing LMMs, we employ the original model directly *without fine-tuning or architectural modifications*.

6.3.6 Chain-of-LoRA

The four roles introduced above demonstrate distinct yet complementary capabilities, collaborating to achieve advanced vision-centric reasoning. However, simply integrating these roles into a single model poses challenges, as their core functionalities can interfere with one another. To avoid inefficiently implementing them as multiple models while still accommodating diverse demands, we propose a novel **Chain-of-LoRA** mechanism to enable flexible and efficient role switching.

In greater detail, all roles are based on a shared LMM backbone and are augmented with different LoRA adapters [88]. Note that an additional timestamp decoder is used exclusively by the grounder. During inference, the framework dynamically activates role-specific LoRA adapters according to the planner, thereby maximizing the strengths of each role while minimizing the memory consumption and architectural modifications to the base model.

6.4 Experiments

We evaluate the effectiveness of VideoMind through extensive experiments across 14 public benchmarks. Specifically, we study the following research questions.

- Q1.** Whether VideoMind is flexible and effective on diverse video understanding tasks compared to the corresponding baselines with task-specific designs?
- Q2.** Compared with (1) training a single agent on multiple tasks or (2) distributing all roles to different models, what advantages does Chain-of-LoRA offer?
- Q3.** What effects does each individual design contribute? More importantly, whether each role is necessary for building such a video reasoning system?

Table 6.2: Statistics of evaluation benchmarks. The datasets encompass three representative tasks – Grounded VideoQA, Video Temporal Grounding, and General VideoQA, with video durations ranging from several seconds to more than 1 hour.

Dataset	Duration	Domain	Main Metrics
<i>Grounded Video Question Answering (Grounding + QA)</i>			
CG-Bench [28]	1624.4s	Diverse	rec.@IoU, acc.@IoU
ReXTime [29]	141.1s	Vlog, News, Activity	mIoU, Acc (IoU $\geq$ 0.5)
NExT-GQA [281]	39.5s	Reasoning	mIoP, Acc@GQA
<i>Video Temporal Grounding (Grounding only)</i>			
Charades-STA [70]	30.1s	Indoor	R@{0.3 ~ 0.7}, mIoU
ActivityNet-Captions [114]	111.4s	Activity	R@{0.3 ~ 0.7}, mIoU
QVHighlights [120]	150s	Vlog, News	R@{0.5, 0.7}, mAP
TACoS [212]	358.2s	Cooking	R@{0.3 ~ 0.7}, mIoU
Ego4D-NLQ [76]	379.0s	Egocentric	R@{0.3 ~ 0.7}, mIoU
ActivityNet-RTL [90]	111.4s	Reasoning	P@0.5, mIoU
<i>General Video Question Answering (QA only)</i>			
Video-MME [66]	1017.9s	Diverse	Acc (w/o subs)
MLVU [343]	930s	Diverse	Acc
LVBench [258]	4101s	Diverse	Acc
MVBench [126]	15s	Diverse	Acc
LongVideoBench [275]	473s	Diverse	Acc

6.4.1 Benchmarks and Settings

The experiments are extensively designed across 14 diverse benchmarks. The statistics are listed in Table 6.2. We mainly evaluate VideoMind on grounded VideoQA and video temporal grounding scenarios, and also study its generalizability on general VideoQA benchmarks. The major benchmarks are introduced below.

CG-Bench [28] is designed for long video grounded question answering, featuring a diverse domain and various evaluation metrics. It includes 1.2K manually curated videos, ranging from 10 to 80 minutes, with a total of 12K QA pairs. The dataset is categorized into perception, reasoning, and hallucination question types, and introduces clue-based evaluation methods like white box and black box assessments to ensure models provide answers based on accurate video reasoning.

ReXTime [29] tests models on complex temporal reasoning, using an automated pipeline for QA pair generation, significantly reducing manual effort. It includes 921 validation and 2.1K test samples, each manually curated for accuracy, and highlights a 14.3% accuracy gap between SoTA models and human performance. This benchmark is crucial for evaluating models on cause-and-effect relationships across video segments.

NExT-GQA [281] aims to challenge models to reason about causal and temporal actions, supporting both multi-choice and open-ended tasks. This is an extension of NExT-QA [280] comprising 10.5K manually labeled video QA pairs with temporal segments. The samples in this benchmark are from “causal” and “temporal” classes, while the “descriptive” questions in NExT-QA are discarded.

Charades-STA [70] contains 10K in-door videos, averaging 30.1 seconds each, with 16K temporal annotations spanning daily activity, alongside free-text descriptions. These rich annotations make Charades-STA particularly suitable for evaluating temporal grounding models under indoor environments.

ActivityNet-Captions [114] is a large-scale benchmark with 20K untrimmed YouTube videos totaling 849 hours, covering diverse activities from personal care to sports. This dataset contains high-quality dense video captioning annotations (3.65 temporally localized sentences per video), which we use as queries for video temporal grounding. Each query has an average length of 13.5 words.

6.4.2 Implementation Details

We leverage the 2B and 7B versions of Qwen2-VL [255] as our base models, and apply LoRA adapters with $\text{rank} = 64$ and $\alpha = 64$ to the planner, grounder, and verifier. The hidden size of the timestamp decoder is set to 256. The maximum number of tokens per frame and maximum number of frames for the planner, grounder, verifier, and answerer are set as [64, 100], [64, 150], [64, 64], and [256,

Table 6.3: Performance comparison on Grounded VideoQA on CG-Bench [28]. Despite its smaller size, VideoMind surpasses GPT-4o [188] and open-source baselines on this challenging long video benchmark (avg. duration: 27 min).

Method	Size	long-acc.	mIoU	rec.@IoU	acc.@IoU
GPT-4o [188]	–	45.2	5.62	8.30	<u>4.38</u>
Gemini-1.5-Pro [213]	–	37.2	3.95	5.81	2.53
Claude-3.5-Sonnet [7]	–	40.5	3.99	5.67	2.79
Video-LLaVA [135]	7B	16.2	1.13	1.96	0.59
VideoLLaMA [328]	7B	18.4	1.21	1.87	0.84
VideoChat2 [126]	7B	19.3	1.28	1.98	0.94
ST-LLM [151]	7B	23.8	2.23	2.86	1.13
ShareGPT4Video [33]	16B	26.7	1.85	2.65	1.01
Chat-UniVi-v1.5 [105]	13B	25.9	2.07	2.53	1.21
VILA [137]	8B	28.7	1.56	2.89	1.35
MiniCPM-v2.6 [240]	8B	30.1	2.35	2.61	1.04
LongVA [333]	7B	28.7	2.94	3.86	1.78
LLaVA-OneVision [122]	7B	31.1	1.63	1.78	1.08
VITA [67]	8×7B	33.3	3.06	3.53	2.06
Qwen2-VL [255]	72B	41.3	3.58	5.32	3.31
InternVL2 [190]	78B	<u>42.2</u>	3.91	5.05	2.64
VideoMind (Ours)	2B	31.0	<u>5.94</u>	<u>8.50</u>	4.02
VideoMind (Ours)	7B	38.4	7.10	9.93	4.67

32], respectively. We train different roles separately and load them together during inference, so that the model can efficiently switch roles by activating different LoRAs. During training, we set the global batch size to 32, and utilize the AdamW optimizer [162] with learning rates of 2e-5, 1e-4, and 5e-5 for planner, grounder, and verifier, respectively. All the roles were trained for 1 epoch on their specific datasets, with a linear warmup in the first 3% steps. During inference, we apply an NMS with $\text{IoU} = 0.75$ to reduce duplicated moments from the grounder.

6.4.3 Q1: Comparison with State-of-the-Arts

Grounded Video Question Answering Table 6.3 compares the performance on Grounded VideoQA on CG-Bench [28], a challenging video benchmark with

Table 6.4: Performance comparison on Grounded VideoQA on ReXTime [29]. FT indicates whether fine-tuned on the downstream training set.

Method	Size	FT	R@0.3	R@0.5	mIoU	Acc	Acc@IoU
VTimeLLM [89]	7B	✗	28.84	17.41	20.14	36.16	–
TimeChat [216]	7B	✗	14.42	7.61	11.65	40.04	–
LITA [90]	13B	✗	29.49	16.29	21.49	34.44	–
VTimeLLM [89]	7B	✓	43.69	26.13	29.92	57.58	17.13
TimeChat [216]	7B	✓	40.13	21.42	26.29	49.46	10.92
VideoMind (Ours)	2B	✗	34.31	22.69	24.83	69.06	17.26
VideoMind (Ours)	7B	✗	38.22	25.52	27.61	74.59	20.20

an average duration of 27 minutes. On temporal grounding metrics (mIoU and rec.@IoU), our lightweight 2B model outperforms all the baselines, including GPT-4o [188] and Gemini 1.5 Pro [213]. Our 7B model further setups a new state-of-the-art on clue-grounded QA (acc.@IoU). In Table 6.4 and Table 6.5, we further present the comparison results on ReXTime [29] and NExT-GQA [281]. Despite the challenges posed by the causal event relationships on RexTime, our model can successfully identify the correct moment, resulting in significant performance boosts compared with zero-shot baselines. On NExT-GQA, compared to agent-based solutions such as LLoVi [324] and LangRepo [107] and end-to-end methods like VideoChat-TPO [299], VideoMind demonstrates its effectiveness on both key event grounding and question answering.

Video Temporal Grounding Since the performance of the grounder and verifier is essential for VideoMind, we evaluate these modules on temporal grounding datasets. In Table 6.6 and Table 6.7, we validate the zero-shot grounding capabilities of VideoMind. Benefiting from (1) the timestamp decoder design of the grounder, and (2) a verifier that refines the results by focusing on critical segments, our model surpasses all LLM-based temporal grounding methods and yields competitive results compared to fine-tuned temporal grounding experts.

Table 6.5: Performance comparison on Grounded VideoQA on NExT-GQA [281].

Method	Size	IoU			IoP			Acc@GQA
		R@0.3	R@0.5	mIoU	R@0.3	R@0.5	mIoP	
FrozenBiLM NG+ [300]	890M	13.5	6.1	9.6	28.5	23.7	24.2	17.5
VIOLETV2 [68]	–	4.3	1.3	3.1	25.1	23.3	23.6	12.8
SeViLA [313]	4B	29.2	13.8	21.7	34.7	22.9	29.5	16.6
LangRepo [107]	8×7B	–	12.2	18.5	–	28.7	31.3	17.1
VideoStreaming [201]	8.3B	–	13.3	19.3	–	31.0	32.2	17.8
LLOVi [324]	1.8T	–	15.3	20.0	–	36.9	<u>37.3</u>	24.3
HawkEye [266]	7B	37.0	19.5	25.7	–	–	–	–
VideoChat-TPO [299]	7B	41.2	<u>23.4</u>	27.7	47.5	32.8	35.6	<u>25.5</u>
VideoMind (Ours)	2B	<u>45.2</u>	23.2	<u>28.6</u>	<u>51.3</u>	32.6	36.4	25.2
VideoMind (Ours)	7B	50.2	25.8	31.4	56.0	<u>35.3</u>	39.0	28.2

We additionally compare VideoMind with representative methods on the challenging TACoS [212], Ego4D-NLQ [76], and QVHighlights [120] datasets in Table 6.8, Table 6.9, and Table 6.10, respectively. Our model performs better than the strong task-specific baseline UniVTG [140] on TACoS but slightly worse than it on Ego4D-NLQ, because UniVTG was originally pre-trained on egocentric videos. Even without egocentric pre-training, VideoMind can still produce comparable results on Ego4D-NLQ. To our best knowledge, VideoMind is **the first LLM-based grounding model that supports multi-moment outputs**, thereby being able to be evaluated on QVHighlights. Compared with task-specific experts, our VideoMind-2B significantly outperforms all previous methods on this challenging dataset, setting a new state-of-the-art.

Reasoning Temporal Localization We also evaluate the generalization ability of grounder and verifier on the more challenging reasoning temporal localization [90] task, which is similar to video temporal grounding, but the queries are not directly describing the moment. The models are required to infer the actual event using their world knowledge. The results in Table 6.11 show that VideoMind can successfully generalize its strong zero-shot grounding capability to complex scenarios.

Table 6.6: Performance of video temporal grounding on Charades-STA [70].

Method	Size	FT	R@0.3	R@0.5	R@0.7	mIoU
<i>Non-LLM-based Specialists</i>						
Moment-DETR [120]	–	✓	65.8	52.1	30.6	45.5
UniVTG [140]	–	✓	70.8	58.1	35.6	50.1
R ² -Tuning [153]	–	✓	70.9	59.8	37.0	50.9
<i>LLM-based Generalists</i>						
VTimeLLM [89]	13B	✗	55.3	34.3	14.7	34.6
TimeChat [216]	7B	✗	51.5	32.2	13.4	–
Momentor [200]	7B	✗	42.6	26.6	11.6	28.5
HawkEye [266]	7B	✗	50.6	31.4	14.5	33.7
ChatVTG [204]	7B	✗	52.7	33.0	15.9	34.9
VideoChat-TPO [299]	7B	✗	58.3	40.2	18.4	38.1
E.T. Chat [158]	4B	✗	65.7	45.9	20.0	42.3
VideoMind (Ours)	2B	✗	<u>67.6</u>	<u>51.1</u>	<u>26.0</u>	<u>45.2</u>
VideoMind (Ours)	7B	✗	73.5	59.1	31.2	50.2

Table 6.8: Video temporal grounding on TACoS [212]. Note that our method was co-trained on this dataset.

Method	Size	FT	R@0.3	R@0.5	R@0.7	mIoU
<i>Non-LLM-based Specialists</i>						
2D-TAN [334]	–	✓	40.0	28.0	12.9	27.2
Moment-DETR [120]	–	✓	38.0	24.7	12.0	25.5
UniVTG [140]	–	✓	51.4	35.0	17.4	33.6
R ² -Tuning [153]	–	✓	49.7	38.7	25.1	35.9
<i>LLM-based Generalists</i>						
VideoMind (Ours)	2B	✓	<u>38.6</u>	<u>26.9</u>	<u>15.5</u>	<u>27.4</u>
VideoMind (Ours)	7B	✓	49.5	36.2	21.4	34.4

Table 6.7: Performance of video temporal grounding on ANet-Captions [114].

Method	Size	FT	R@0.3	R@0.5	R@0.7	mIoU
<i>Non-LLM-based Specialists</i>						
2D-TAN [334]	–	✓	60.4	43.4	25.0	42.5
MMN [267]	–	✓	64.5	48.2	29.4	46.6
VDI [164]	–	✓	–	48.1	28.8	–
<i>LLM-based Generalists</i>						
VideoChat [125]	7B	✗	8.8	3.7	1.5	7.2
Video-LLaMA [328]	7B	✗	6.9	2.1	0.8	6.5
Video-ChatGPT [171]	7B	✗	26.4	13.6	6.1	18.9
Valley [166]	7B	✗	30.6	13.7	8.1	21.9
ChatVTG [204]	7B	✗	40.7	22.5	9.4	27.2
Momentor [200]	7B	✗	42.9	23.0	12.4	29.3
E.T. Chat [158]	4B	✗	24.1	12.8	6.1	18.9
VideoMind (Ours)	2B	✗	<u>44.0</u>	<u>26.5</u>	<u>12.6</u>	<u>30.1</u>
VideoMind (Ours)	7B	✗	48.4	30.3	15.7	33.3

Table 6.9: Performance of video temporal grounding on Ego4D-NLQ [76].

Method	Size	FT	R@0.3	R@0.5	R@0.7	mIoU
<i>Non-LLM-based Specialists</i>						
2D-TAN [334]	–	✓	4.3	1.8	0.6	3.4
VSLNet [329]	–	✓	4.5	2.4	1.0	3.5
Moment-DETR [120]	–	✓	4.3	1.8	0.7	3.5
UniVTG [140]	–	✓	7.3	4.0	1.3	4.9
R ² -Tuning [153]	–	✓	7.2	4.5	2.1	4.9
UniVTG [140]	–	✗	6.5	3.5	1.2	4.6
<i>LLM-based Generalists</i>						
VideoMind (Ours)	2B	✗	<u>5.9</u>	<u>2.9</u>	<u>1.2</u>	<u>4.7</u>
VideoMind (Ours)	7B	✗	7.2	3.7	1.7	5.4

General Video Question Answering We are also interested in whether our temporally augmented design can improve general VideoQA tasks. In Table 6.12, we evaluate our model on three long video benchmarks to determine if the Chain-of-LoRA design generalizes to common settings. Our designs effectively help the model localize cue segments before answering the question. For long VideoQA, we also provide evaluations on LongVideoBench [275] in Table 6.13, which further ver-

Table 6.10: Fine-tuned video temporal grounding results on QVHighlights [120].

Method	Size	R1		mAP		
		@0.5	@0.7	@0.5	@0.75	Avg.
Non-LLM-based Specialists						
XML [121]	–	41.83	30.35	44.63	31.73	32.14
XML+ [120]	–	46.69	33.46	47.89	34.67	34.90
Moment-DETR [120]	–	59.78	40.33	60.51	35.36	36.14
UMT [155]	–	60.83	43.26	57.33	39.12	38.08
MomentDiff [128]	–	58.21	41.48	54.57	37.21	36.84
QD-DETR [178]	–	62.40	44.98	62.52	39.88	39.86
UniVTG [140]	–	65.43	50.06	64.06	45.02	43.63
R ² -Tuning [153]	–	68.03	49.35	69.04	47.56	46.17
LLM-based Generalists						
VideoMind (Ours)	2B	<u>75.42</u>	<u>59.35</u>	<u>74.11</u>	<u>55.15</u>	<u>51.60</u>
VideoMind (Ours)	7B	78.53	61.09	76.07	58.17	54.19

ifies the effectiveness of VideoMind on videos scaling to one-hour long. Table 6.14 presents more results of VideoMind on MVBench [126], which is a benchmark with very short videos (around 15s). Our model can still achieve good performance.

6.4.4 Q2: The Advantages of Chain-of-LoRA

Table 6.15 studies the effect of role integration on VideoMind-2B. First, text-based CoT does not improve the base model, highlighting the need for a vision-centric reasoning strategy. Second, although the all-distributed approach achieves the best performance, it requires multiple copies (4×) of the model weights. In contrast, Chain-of-LoRA maintains top performance while being efficient.

6.4.5 Q3: Ablation Studies

Effect of Individual Roles The contributions of different roles are studied in Table 6.16. Our observations are as follows: (1) **Grounder**: By identifying visual

Table 6.11: Performance comparison on reasoning temporal localization on ActivityNet-RTL [90]. Our zero-shot method outperforms the fine-tuned baseline LITA by a considerable margin.

Method	Size	FT	P@0.5	mIoU
LITA [90]	7B	✓	21.2	24.1
LITA [90]	13B	✓	25.9	28.6
VideoMind (Ours)	2B	✗	20.1	22.7
VideoMind (Ours)	7B	✗	28.0	31.3

Table 6.12: VideoQA performance on Video-MME [66], MLVU [343], and IVBench [258].

Method	Size	Video-MME		MLVU	IVBench
		All	Long	M-Avg	Overall
Gemini-1.5-Pro [213]	–	75.0	67.4	–	33.1
GPT-4o [188]	–	71.9	65.3	54.5	30.8
Video-LLaVA [135]	7B	41.1	37.8	29.3	–
TimeChat [216]	7B	34.3	32.1	30.9	22.3
MovieChat [228]	7B	38.2	33.4	25.8	22.5
PLLaVA [291]	34B	40.0	34.7	53.6	26.1
VideoChat-TPO [299]	7B	48.8	41.0	54.7	–
LongVA [333]	7B	52.6	46.2	56.3	–
VideoMind (Ours)	2B	55.4	46.3	58.7	35.4
VideoMind (Ours)	7B	58.2	49.2	64.4	40.8

Table 6.13: Performance of VideoQA on LongVideoBench [275] val split.

Method	Size	Acc	Acc @ Duration Groups			
			(8, 15]	(15, 60]	(180, 600]	(900, 3600]
GPT-4o [188]	–	66.7	71.4	76.7	69.1	60.9
GPT-4 Turbo [2]	–	59.0	65.2	68.2	62.4	50.5
Gemini-1.5-Pro [213]	–	64.0	67.4	75.1	65.3	58.6
Gemini-1.5-Flash [213]	–	61.6	68.3	76.2	62.6	54.0
Idefics2 [118]	8B	49.7	59.8	65.7	47.8	42.7
Phi-3-Vision [11]	4B	49.6	59.3	61.6	46.8	44.7
Mantis-Idefics2 [102]	8B	47.0	56.6	55.8	45.6	42.2
Mantis-BakLLaVA [102]	7B	43.7	53.4	57.6	40.3	38.7
VideoMind (Ours)	2B	48.8	59.3	59.3	49.3	41.7
VideoMind (Ours)	7B	56.3	67.7	67.4	56.8	48.6

cues, the grounder can slightly improve QA accuracy, indicating that the grounder is especially effective on long videos. (2) Verifier: Selecting the best candidate through the verifier improves grounding performance, yielding a consistent gain of 3.2 mIoU on Charades-STA. (3) Planner: Coordinating roles via the planner – even when performing grounding on only 40% samples – boosts the accuracy from 69.2 to 70.0. This highlights the model’s flexibility to adaptively determine whether to perform grounding under different temporal contexts.

Table 6.14: Performance comparison on general VideoQA on MVBench [126].

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 [187]	–	55.5	63.5	72.0	46.5	<u>73.5</u>	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 [171]	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 [328]	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 [125]	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 [135]	7B	46.0	42.5	56.5	39.0	53.5	53.0	48.0	<u>41.0</u>	29.0	31.5	82.5	<u>45.0</u>	26.0	53.0	41.5	33.5	41.5	27.5	38.5	31.5	43.0
TimeChat [216]	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 [291]	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
ShareGPT4Video [33]	7B	49.5	39.5	79.5	40.0	54.5	82.5	54.5	32.5	50.5	41.5	84.5	35.5	62.5	75.0	51.0	25.5	46.5	28.5	39.0	51.5	51.2
ST-LLM [151]	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+ [170]	3.8B	69.0	60.0	83.0	48.5	66.5	85.5	75.5	36.0	44.0	34.0	<u>89.5</u>	39.5	71.0	<u>90.5</u>	45.0	53.0	50.0	29.5	44.0	60.0	58.7
VideoChat2 [126]	7B	<u>75.5</u>	58.0	<u>83.5</u>	50.5	60.5	<u>87.5</u>	<u>74.5</u>	45.0	47.5	<u>44.0</u>	82.5	37.0	64.5	87.5	51.0	66.5	47.0	35.0	37.0	72.5	60.4
VideoMind (Ours)	2B	77.0	78.0	77.0	46.5	70.5	87.0	71.5	33.0	<u>48.0</u>	39.5	91.0	53.0	<u>78.0</u>	89.0	43.5	53.5	<u>61.5</u>	<u>37.5</u>	49.5	53.0	<u>61.9</u>
VideoMind (Ours)	7B	74.0	<u>71.5</u>	81.0	<u>50.0</u>	<u>77.0</u>	93.0	75.0	38.0	48.5	46.0	91.0	39.0	80.0	94.5	<u>49.5</u>	<u>55.5</u>	70.0	40.5	<u>57.0</u>	<u>61.0</u>	64.6

Table 6.15: Comparison with different test-time scaling strategies.

Method	Memory	NExT-GQA		Charades-STA		Video-MME	
		mIoU	Acc	R@0.5	mIoU	All	Long
Qwen2-VL-2B	4.1G	–	69.6	–	–	53.0	43.1
+ CoT [272]	4.1G	–	69.7	–	–	52.8	43.3
+ All-in-One	4.2G	<u>28.0</u>	<u>70.5</u>	<u>47.8</u>	<u>42.1</u>	<u>53.6</u>	<u>43.6</u>
+ All-Distributed	16.6G	28.6	71.4	51.1	45.2	55.4	46.3
+ Chain-of-LoRA	4.2G	28.6	71.4	51.1	45.2	55.4	46.3

Effect of the Temporal Feature Pyramid Table 6.17 studies the effectiveness of the temporal feature pyramid. Our baseline model directly makes predictions on the last-layer transformer outputs. When adding more pyramid levels, the performance of video temporal grounding consistently improves under all metrics on the Charades-STA [70] dataset under zero-shot setting, suggesting the effectiveness of improving the robustness of the model when facing moments with different lengths.

Design Choices of Verifier In Table 6.18, we examine various design choices for the verifier. The term “Direct” refers to the method where the grounded moment is directly sent into the model without any expansion. “Expand” denotes expanding the temporal boundaries by 50%, while “Textual” involves adding supplementary textual information to indicate the length of the target event. “Special Token” rep-

Table 6.16: Effects of individual roles. The top half evaluates the full system on grounded VideoQA, while the bottom half details the impact of the grounder and verifier on video temporal grounding. Ground% indicates the percentage of samples processed with the grounder.

VideoMind Roles					ReXTime		Charades-STA		
Answerer	Grounder	Verifier	Planner	Ground%	mIoU	Acc	R@0.5	R@0.7	mIoU
✓				0%	–	68.0	–	–	–
✓	✓			100%	24.5	68.8	–	–	–
✓	✓	✓		100%	<u>24.8</u>	69.1	–	–	–
✓	✓	✓	✓	100%	24.7	<u>69.2</u>	–	–	–
✓	✓	✓	✓	40%	26.7	70.0	–	–	–
	✓				–	–	<u>47.2</u>	<u>21.7</u>	<u>42.0</u>
	✓	✓			–	–	51.1	26.0	45.2

Table 6.17: Effect of the temporal feature pyramid on Charades-STA [70].

#Pyramid Levels	Charades-STA			
	R@0.3	R@0.5	R@0.7	mIoU
1	60.55	44.57	15.82	38.13
2	61.51	46.90	19.36	40.43
3	<u>62.62</u>	<u>47.02</u>	<u>20.08</u>	<u>41.27</u>
4	63.55	47.23	21.69	42.02

Table 6.18: Effect of different verifier styles on Charades-STA [70].

Verifier Type	Charades-STA			
	R@0.3	R@0.5	R@0.7	mIoU
Direct	60.42	45.28	19.32	39.84
Expand	65.10	48.70	23.15	43.57
Textual	<u>65.24</u>	<u>49.33</u>	<u>23.89</u>	<u>44.01</u>
Special Token	67.63	51.05	25.99	45.22

resents our final approach, utilizing special tokens to denote the grounded start and end timestamps. The comparison demonstrates that expanding the temporal boundaries effectively broadens the verifier’s perceptual range, and the use of special tokens enhances the model’s awareness of precise moment boundaries.

6.4.6 Visualization

In Figure 6.6, we illustrate how VideoMind applies all roles to progressively derive the correct answer while avoiding potential mistakes. The planner determines what roles are needed, then calls the grounder to generate candidate moments. The

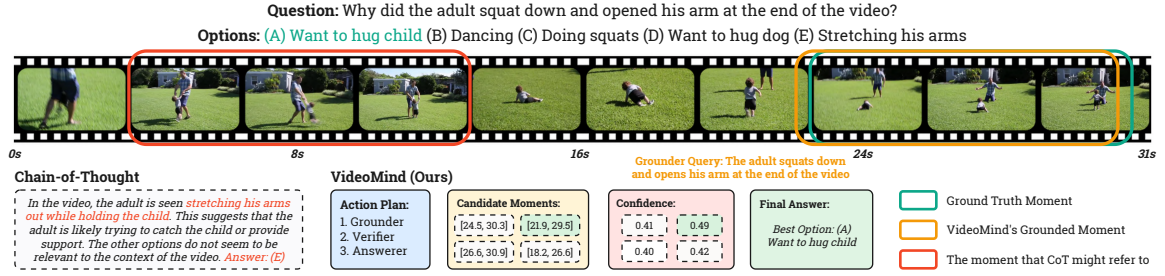


Figure 6.6: Visualization of VideoMind’s reasoning process. Through chaining the grounder, verifier, and answerer, our model accurately localizes the critical moment and selects the correct answer, avoiding confusion from incorrect segments.

verifier selects the most relevant segment (highlighted in yellow), which is then zoomed-in and passed to the answerer for further reasoning.

6.5 Summary

In this chapter, we finally combine all the relevant techniques introduced in the previous chapters, and propose VideoMind, a novel visual-language agent designed for temporal grounded video reasoning. Our approach employs an agentic workflow consisting of several roles, along with a Chain-of-LoRA strategy to efficiently switch among these roles. Extensive experiments in grounded video question answering, video temporal grounding, and general video question answering demonstrate its effectiveness and significance in providing precise, evidence-grounded answers. The contributions in this chapter are summarized below:

1. An explicit vision-centric formulation of multi-step long video reasoning, including the steps of planning, grounding, verification, and question answering. This formulation is in contrast to conventional pure-text based reasoning and can better handle vision-centric tasks.
2. A flexible, learnable, and scalable multi-agent framework (VideoMind) for

interactive image and video understanding. This framework explicitly defines four roles to handle different essential steps for long video reasoning.

3. A Chain-of-LoRA strategy for efficient role-switching and effective inference acceleration. This design combines the advantages of all-in-one learning and multi-agent frameworks.

The novelties behind our contributions in this chapter are:

1. We, for the first time, propose and demonstrate the significance and effectiveness of vision-centric multi-step video reasoning. Compared to pure-text-based reasoning like OpenAI o1 [189] and DeepSeek R1 [79], our new formulation is more suitable for vision-centric tasks.
2. The proposed VideoMind framework offers a simple yet scalable implementation of our formulation. Its detailed role-specific technical designs are novel and hold potential to inspire future research in related areas.
3. Our Chain-of-LoRA technology introduces a novel approach that combines the strong performance of multi-agent systems with the efficiency of an all-in-one joint training strategy. To the best of our knowledge, this is also the first solution to employ LoRA adapters dynamically during model inference, rather than solely in training.

6.6 Limitations & Future Work

The proposed framework represents our initial attempt at a multi-agent-based solution for interactive video understanding. In this chapter, we identify two limitations that warrant deeper exploration. First, the current planner design and role distribution are based on heuristic decisions, which might be sub-optimal and can potentially be more flexible. For example, a possible enhancement of planner would be allowing it to call the answerer for an initial answer generation, then generate

the detailed plan based on the preliminary answer. Second, the entire inference pipeline lacks an explicit reflection design. Therefore, the overall performance of each role is largely affected by the previous role. Enabling the reflection capability is essential for building a real interactive content understanding agent. To address these limitations, a promising direction for future work is to integrate the planner and answerer into a unified agent that possesses reflection capabilities. This would allow the model, upon detecting that a predicted answer may be incorrect, to re-trigger the grounder and/or verifier dynamically for correction and improvement. Modern reinforcement learning training techniques such as PPO [220] and GRPO [79] might also be utilized.

Chapter 7

Conclusion

This thesis presents a systematic approach towards building an interactive multi-modal content understanding system. We identify the fundamental limitation of contemporary AI systems: the reliance on the single-pass processing paradigm, which is sub-optimal for comprehensively encoding all information for diverse downstream tasks. Our solution to this limitation is a new paradigm that treats multi-modal content as an environment, and develops an agent to actively explore this environment during reasoning.

This agentic formulation introduces three core challenges: task-driven perception, open-ended reasoning, and proactive interaction. Chapter 2 addresses the integration of visual, audio, and textual modalities for video moment retrieval and highlight detection, particularly under varying modality reliability. Chapter 3 further proposes a simple yet effective solution for video temporal grounding through parameter- and memory-efficient transfer learning. These techniques are further extended to LLM-based architectures. Chapter 4 introduces a complete solution for event-level and time-sensitive video-language understanding, while Chapter 5 focuses on the more fine-grained spatial-temporal region/object referring and segmentation. Finally, Chapter 6 brings all these relevant techniques together into a

multi-agent system capable of planning, grounding, verification, and question answering. We have demonstrated through extensive experiments that this agentic solution for interactive multi-modal content understanding brings significant performance gains compared with conventional single-pass approaches, especially in complex, long-video scenarios.

In summary, we introduce a new paradigm of multi-modal content understanding, and provide innovative solutions in terms of problem formulations, model architectural designs, training objectives, high-quality data, and extensive evaluation. We hope that this thesis could inspire future research in the area of multi-modal understanding, large language models, and agents.

7.1 Potential Future Impact

While we have demonstrated significant performance gains across diverse public benchmarks, the broader impact of the innovations in this thesis extends far beyond the immediate tasks explored. Here, we outline some potential future impact of our research across multiple research directions and real-world application scenarios, then articulate open challenges and promising future directions.

7.1.1 Impact on Research Directions

Multi-modal Content Understanding The proposed new paradigm of treating multi-modal content as an interactive environment introduces a foundation for the next-generation content understanding system – from passive encoder-decoder architectures to proactive agentic solutions. We believe that this perspective opens up new avenues for advancing image/video/audio understanding.

Agentic AI The core conceptual innovation of our multi-agent system in Chapter [6](#)

– roles capable of planning, grounding, verification, and reasoning – can inform the design of future multi-agent systems. Our Chain-of-LoRA technique also naturally complements ongoing efforts in LLM-as-agent research and contributes to the development of more general-purpose agents.

MLLMs for Fine-grained Image/Video Understanding Our technical contributions in Chapter 4 and Chapter 5 outline a practical roadmap for MLLM-based fine-grained spatial-temporal image/video understanding. The proposed solutions have demonstrated robustness and scalability, and are strong candidates to become standard architectural components in future MLLM designs.

Efficient Fine-tuning and Transfer Learning The strategies and theoretical foundations for parameter- and memory-efficient learning introduced in Chapter 3 provide a robust framework for the scalable deployment of multi-modal models. These approaches are well-suited for real-world applications on resource-constrained platforms, thereby advancing the democratization of multi-modal intelligence.

7.1.2 Impact on Real-World Applications

Multimedia Analytics and Content Production Our moment retrieval and highlight detection techniques introduced in Chapter 2~4 could power automated multimedia curation, production, and management. These capabilities allow platforms to generate trailers, detect highlights, and segment long-form content into searchable and browsable units to enhance user engagement.

Education and E-Learning The relevant techniques in video spatial/temporal understanding and agentic exploration enable educational assistants capable of understanding and summarizing video lectures, identifying key moments, and answering complex questions from learners. These AI-driven systems have the potential to

revolutionize personalized learning and assessment.

Robotics and Embodied AI Multi-modal agents proposed in Chapter 6 that actively perceive environments through planning, grounding, verification, reasoning, and reflection could form the foundation of embodied AI systems for applications such as household robotics, autonomous navigation, and industrial automation.

Healthcare and Assistive Technology Our technical innovations in video understanding can also support medical video analysis, such as in surgery recordings or patient monitoring systems. They can help detect critical events, assist practitioners in reviewing long procedures, or aid in training simulations.

7.2 Future Work

Although the proposed paradigm demonstrates significant potential, many challenges remain unresolved. One of the most promising directions for future research is the explicit incorporation of a vision-centric reflection mechanism. Specifically, this would allow the model to adaptively revisit previous reasoning steps upon detecting inconsistencies or errors in the current inference process. One potential approach to implementing such a mechanism involves integrating Monte Carlo Tree Search (MCTS) [46] with vision-centric designs. This would enable models to actively explore visual content and iteratively refine their reasoning paths. Another direction worth investigating is the integration of reinforcement learning (RL) [220, 79] with visual reasoning tasks. With appropriately designed rewards, RL provides greater potential to learn more robust reasoning strategies while mitigating the risk of overfitting. Together, these approaches could play a pivotal role in developing the next generation of interactive content understanding systems. We hope that this thesis can provide new insights into interactive content understanding and inspire more research in this promising but challenging area.

References

- [1] Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv:2404.14219*, 2024.
- [2] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Floren-
cia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anad-
kat, et al. Gpt-4 technical report. *arXiv:2303.08774*, 2023.
- [3] Triantafyllos Afouras, Effrosyni Mavroudi, Tushar Nagarajan, Huiyu Wang, and
Lorenzo Torresani. Ht-step: Aligning instructional articles with how-to videos. In
NeurIPS, 2024.
- [4] Meta AI. Llama 3 model card, 2024.
- [5] John R Anderson. *The adaptive character of thought*. Psychology Press, 2013.
- [6] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and
Bryan Russell. Localizing moments in video with natural language. In *ICCV*, 2017.
- [7] Anthropic. Claude 3.5 sonnet model card, 2025.
- [8] Anthropic. Claude 3.7 sonnet system card, 2025.
- [9] Evlampios Apostolidis, Eleni Adamantidou, Alexandros I Metsai, Vasileios Mezaris,
and Ioannis Patras. Video summarization using deep neural networks: A survey.
Proceedings of the IEEE, 109(11):1838–1863, 2021.

References

- [10] 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.
- [11] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. In *NeurIPS*, 2016.
- [12] Taivanbat Badamdorj, Mrigank Rochan, Yang Wang, and Li Cheng. Joint visual and audio learning for video highlight detection. In *ICCV*, 2021.
- [13] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv:2309.16609*, 2023.
- [14] 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.
- [15] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2.5-vl technical report. *arXiv:2502.13923*, 2025.
- [16] 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*, 2024.
- [17] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. A clip-hitchhiker’s guide to long video retrieval. *arXiv:2205.08508*, 2022.
- [18] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *ACL Workshop*, 2005.
- [19] Adam Botach, Evgenii Zheltonozhskii, and Chaim Baskin. End-to-end referring video object segmentation with multimodal transformers. In *CVPR*, 2022.
- [20] Leonard Bärman and Alex Waibel. Where did i leave my keys?-episodic-memory-based question answering on egocentric videos. In *CVPR*, 2022.

-
- [21] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015.
- [22] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *CVPR*, 2018.
- [23] Zheng Cai, Maosong Cao, Haojiong Chen, Kai Chen, Keyu Chen, Xin Chen, Xun Chen, Zehui Chen, Zhi Chen, Pei Chu, et al. Internlm2 technical report. *arXiv:2403.17297*, 2024.
- [24] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017.
- [25] Souradip Chakraborty, Soumya Suvra Ghosal, Ming Yin, Dinesh Manocha, Mengdi Wang, Amrit Singh Bedi, and Furong Huang. Transfer q star: Principled decoding for llm alignment. *arXiv:2405.20495*, 2024.
- [26] Kelvin CK Chan, Xintao Wang, Ke Yu, Chao Dong, and Chen Change Loy. Basicvsr: The search for essential components in video super-resolution and beyond. In *CVPR*, 2021.
- [27] Keshigeyan Chandrasegaran, Agrim Gupta, Lea M Hadzic, Taran Kota, Jimming He, Cristóbal Eyzaguirre, Zane Durante, Manling Li, Jiajun Wu, and Li Fei-Fei. Hourvideo: 1-hour video-language understanding. In *NeurIPS*, 2024.
- [28] Guo Chen, Yicheng Liu, Yifei Huang, Yuping He, Baoqi Pei, Jilan Xu, Yali Wang, Tong Lu, and Limin Wang. Cg-bench: Clue-grounded question answering benchmark for long video understanding. *arXiv:2412.12075*, 2024.
- [29] Jr-Jen Chen, Yu-Chien Liao, Hsi-Che Lin, Yu-Chu Yu, Yen-Chun Chen, and Frank Wang. Rextime: A benchmark suite for reasoning-across-time in videos. In *NeurIPS*, 2024.
- [30] Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechun Liu, Pengchuan Zhang, Raghuraman Krishnamoorthi, Vikas Chandra, Yunyang Xiong, and Mohamed Elho-

References

- seiny. Minigpt-v2: large language model as a unified interface for vision-language multi-task learning. *arXiv:2310.09478*, 2023.
- [31] 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.
- [32] Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, et al. Are we on the right way for evaluating large vision-language models? *arXiv:2403.20330*, 2024.
- [33] Lin Chen, Xilin Wei, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Zhenyu Tang, Li Yuan, et al. Sharegpt4video: Improving video understanding and generation with better captions. In *NeurIPS*, 2024.
- [34] Shaoxiang Chen and Yu-Gang Jiang. Semantic proposal for activity localization in videos via sentence query. In *AAAI*, 2019.
- [35] Ting Chen, Saurabh Saxena, Lala Li, David J Fleet, and Geoffrey Hinton. Pix2seq: A language modeling framework for object detection. *arXiv:2109.10852*, 2021.
- [36] 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*, 2014.
- [37] Xiuyuan Chen, Yuan Lin, Yuchen Zhang, and Weiran Huang. Autoeval-video: An automatic benchmark for assessing large vision language models in open-ended video question answering. *arXiv:2311.14906*, 2023.
- [38] 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*, 2024.
- [39] Zewen Chen, Juan Wang, Wen Wang, Sunhan Xu, Hang Xiong, Yun Zeng, Jian Guo, Shuxun Wang, Chunfeng Yuan, Bing Li, others Shilong Zhang, Peize Sun, Shoufa Chen, Min Xiao, Wenqi Shao, Wenwei Zhang, Yu Liu, Kai Chen, and

- Ping Luo. Gpt4roi: Instruction tuning large language model on region-of-interest. *arXiv:2307.03601*, 2023.
- [40] 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.
- [41] 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.
- [42] Ho Kei Cheng and Alexander G Schwing. Xmem: Long-term video object segmentation with an atkinson-shiffrin memory model. In *ECCV*, 2022.
- [43] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90% chatgpt quality, 2023.
- [44] Andy Clark. *Being there: Putting brain, body, and world together again*. MIT Press, 1998.
- [45] Andy Clark. Whatever next? predictive brains, situated agents, and the future of cognitive science. *Behavioral and brain sciences*, 36(3):181–204, 2013.
- [46] Remi Coulom. Efficient selectivity and backup operators in monte-carlo tree search. In *CGAMES*, 2006.
- [47] 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*, 2024.
- [48] 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.

References

- [49] Google DeepMind. Introducing gemini 2.0: our new ai model for the agentic era, 2024.
- [50] Google DeepMind. Gemini 2.5: Our most intelligent ai model, 2025.
- [51] 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.
- [52] Shangzhe Di and Weidi Xie. Grounded question-answering in long egocentric videos. *arXiv:2312.06505*, 2023.
- [53] Haiwen Diao, Bo Wan, Ying Zhang, Xu Jia, Huchuan Lu, and Long Chen. Unipt: Universal parallel tuning for transfer learning with efficient parameter and memory. *arXiv:2308.14316*, 2023.
- [54] Henghui Ding, Chang Liu, Shuting He, Xudong Jiang, and Chen Change Loy. Mevis: A large-scale benchmark for video segmentation with motion expressions. In *ICCV*, 2023.
- [55] Chao Dong, Chen Change Loy, and Xiaoou Tang. Accelerating the super-resolution convolutional neural network. In *ECCV*, 2016.
- [56] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2020.
- [57] 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.
- [58] Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Sean Welleck, Peter West, Chandra Bhagavatula, Ronan Le Bras, et al. Faith and fate: Limits of transformers on compositionality. In *NeurIPS*, 2024.

-
- [59] Victor Escorcia, Mattia Soldan, Josef Sivic, Bernard Ghanem, and Bryan Russell. Temporal localization of moments in video collections with natural language. *arXiv:1907.12763*, 2019.
- [60] Yue Fan, Xiaojian Ma, Rujie Wu, Yuntao Du, Jiaqi Li, Zhi Gao, and Qing Li. Videoagent: A memory-augmented multimodal agent for video understanding. *arXiv:2403.11481*, 2024.
- [61] Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva: Exploring the limits of masked visual representation learning at scale. In *CVPR*, 2023.
- [62] 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.
- [63] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *ICCV*, 2019.
- [64] Simon Frieder, Luca Pinchetti, Ryan-Rhys Griffiths, Tommaso Salvatori, Thomas Lukasiewicz, Philipp Petersen, and Julius Berner. Mathematical capabilities of chatgpt. In *NeurIPS*, 2024.
- [65] Karl Friston. The free-energy principle: a unified brain theory. *Nature reviews neuroscience*, 11(2):127–138, 2010.
- [66] 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.
- [67] Chaoyou Fu, Haojia Lin, Zuwei Long, Yunhang Shen, Yuhang Dai, Meng Zhao, Yi-Fan Zhang, Shaoqi Dong, Yangze Li, Xiong Wang, et al. Vita: Towards open-source interactive omni multimodal llm. *arXiv:2408.05211*, 2024.

References

- [68] Tsu-Jui Fu, Linjie Li, Zhe Gan, Kevin Lin, William Yang Wang, Lijuan Wang, and Zicheng Liu. An empirical study of end-to-end video-language transformers with masked visual modeling. In *CVPR*, 2023.
- [69] Difei Gao, Lei Ji, Luowei Zhou, Kevin Qinghong Lin, Joya Chen, Zihan Fan, and Mike Zheng Shou. Assistgpt: A general multi-modal assistant that can plan, execute, inspect, and learn. *arXiv:2306.08640*, 2023.
- [70] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. Tall: Temporal activity localization via language query. In *ICCV*, 2017.
- [71] Junyu Gao and Changsheng Xu. Fast video moment retrieval. In *ICCV*, 2021.
- [72] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *ICASSP*, 2017.
- [73] James J Gibson. *The ecological approach to visual perception*. Taylor and Francis Group, 1979.
- [74] Yuan Gong, Yu-An Chung, and James Glass. Ast: Audio spectrogram transformer. *arXiv:2104.01778*, 2021.
- [75] Alex Gorban, Haroon Idrees, Yu-Gang Jiang, A Roshan Zamir, Ivan Laptev, Mubarak Shah, and Rahul Sukthankar. Thumos challenge: Action recognition with a large number of classes, 2015.
- [76] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *CVPR*, 2022.
- [77] Richard Langton Gregory. Perceptions as hypotheses. *Philosophical Transactions of the Royal Society of London. B, Biological Sciences*, 290(1038):181–197, 1980.
- [78] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv:2312.00752*, 2023.

-
- [79] 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.
- [80] Yongxin Guo, Jingyu Liu, Mingda Li, Xiaoying Tang, Xi Chen, and Bo Zhao. Vtg-llm: Integrating timestamp knowledge into video llms for enhanced video temporal grounding. *arXiv:2405.13382*, 2024.
- [81] Michael Gygli, Helmut Grabner, Hayko Riemenschneider, and Luc Van Gool. Creating summaries from user videos. In *ECCV*, 2014.
- [82] Michael Gygli, Yale Song, and Liangliang Cao. Video2gif: Automatic generation of animated gifs from video. In *CVPR*, 2016.
- [83] Junwen He, Yifan Wang, Lijun Wang, Huchuan Lu, Jun-Yan He, Jin-Peng Lan, Bin Luo, and Xuansong Xie. Multi-modal instruction tuned llms with fine-grained visual perception. *arXiv:2403.02969*, 2024.
- [84] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.
- [85] Muyang He, Yexin Liu, Boya Wu, Jianhao Yuan, Yueze Wang, Tiejun Huang, and Bo Zhao. Efficient multimodal learning from data-centric perspective. *arXiv:2402.11530*, 2024.
- [86] Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, R Channing Moore, Manoj Plakal, Devin Platt, Rif A Saurous, Bryan Seybold, et al. Cnn architectures for large-scale audio classification. In *ICASSP*, 2017.
- [87] Fa-Ting Hong, Xuanteng Huang, Wei-Hong Li, and Wei-Shi Zheng. Mini-net: Multiple instance rank- ing network for video highlight detection. In *ECCV*, 2020.
- [88] 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.

References

- [89] Bin Huang, Xin Wang, Hong Chen, Zihan Song, and Wenwu Zhu. Vtimellm: Empower llm to grasp video moments. In *CVPR*, 2024.
- [90] De-An Huang, Shijia Liao, Subhashree Radhakrishnan, Hongxu Yin, Pavlo Molchanov, Zhiding Yu, and Jan Kautz. Lita: Language instructed temporal-localization assistant. *arXiv:2403.19046*, 2024.
- [91] Gabriel Huang, Bo Pang, Zhenhai Zhu, Clara Rivera, and Radu Soricut. Multimodal pretraining for dense video captioning. *arXiv:2011.11760*, 2020.
- [92] Siteng Huang, Biao Gong, Yulin Pan, Jianwen Jiang, Yiliang Lv, Yuyuan Li, and Donglin Wang. Vop: Text-video co-operative prompt tuning for cross-modal retrieval. In *CVPR*, 2023.
- [93] Weizhong Huang, Yuxin Zhang, Xiawu Zheng, Yang Liu, Jing Lin, Yiwu Yao, and Rongrong Ji. Dynamic low-rank sparse adaptation for large language models. *arXiv:2502.14816*, 2025.
- [94] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, 2021.
- [95] Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *ICML*, 2021.
- [96] William James, Frederick Burkhardt, Fredson Bowers, and Kestutis Skrupskelis. *The principles of psychology*. Macmillan London, 1890.
- [97] 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.
- [98] Jinhyun Jang, Jungin Park, Jin Kim, Hyeongjun Kwon, and Kwanghoon Sohn. Knowing where to focus: Event-aware transformer for video grounding. In *ICCV*, 2023.
- [99] Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. Tgif-qa: Toward spatio-temporal reasoning in visual question answering. In *CVPR*, 2017.

-
- [100] Samy Jelassi, David Brandfonbrener, Sham M Kakade, and Eran Malach. Repeat after me: Transformers are better than state space models at copying. *arXiv:2402.01032*, 2024.
- [101] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *ECCV*, 2022.
- [102] Dongfu Jiang, Xuan He, Huaye Zeng, Cong Wei, Max Ku, Qian Liu, and Wenhui Chen. Mantis: Interleaved multi-image instruction tuning. *arXiv:2405.01483*, 2024.
- [103] Ruixiang Jiang, Lingbo Liu, and Changwen Chen. Clip-count: Towards text-guided zero-shot object counting. In *ACM MM*, 2023.
- [104] Yu-Gang Jiang, Jingen Liu, A Roshan Zamir, George Toderici, Ivan Laptev, Mubarak Shah, and Rahul Sukthankar. Thumos challenge: Action recognition with a large number of classes, 2014.
- [105] Peng Jin, Ryuichi Takanobu, Caiwan Zhang, Xiaochun Cao, and Li Yuan. Chat-univi: Unified visual representation empowers large language models with image and video understanding. *arXiv:2311.08046*, 2023.
- [106] Chen Ju, Tengda Han, Kunhao Zheng, Ya Zhang, and Weidi Xie. Prompting visual-language models for efficient video understanding. In *ECCV*, 2022.
- [107] Kumara Kahatapitiya, Kanchana Ranasinghe, Jongwoo Park, and Michael S Ryoo. Language repository for long video understanding. *arXiv:2403.14622*, 2024.
- [108] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. Referitgame: Referring to objects in photographs of natural scenes. In *EMNLP*, 2014.
- [109] 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.
- [110] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

References

- [111] 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*, 2023.
- [112] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.
- [113] Qiuqiang Kong, Yin Cao, Turab Iqbal, Yuxuan Wang, Wenwu Wang, and Mark D Plumbley. Panns: Large-scale pretrained audio neural networks for audio pattern recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:2880–2894, 2020.
- [114] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *ICCV*, 2017.
- [115] Sukanya Kudi and Anoop M Namboodiri. Words speak for actions: Using text to find video highlights. In *ACPR*, 2017.
- [116] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. In *CVPR*, 2024.
- [117] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Fractalnet: Ultra-deep neural networks without residuals. *arXiv:1605.07648*, 2016.
- [118] Hugo Laurencon, Leo Tronchon, Matthieu Cord, and Victor Sanh. What matters when building vision-language models? In *NeurIPS*, 2024.
- [119] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *ECCV*, 2018.
- [120] Jie Lei, Tamara L Berg, and Mohit Bansal. Qvhighlights: Detecting moments and highlights in videos via natural language queries. In *NeurIPS*, 2021.
- [121] Jie Lei, Licheng Yu, Tamara L Berg, and Mohit Bansal. Tvr: A large-scale dataset for video-subtitle moment retrieval. In *ECCV*, 2020.

-
- [122] 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.
- [123] Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. Seed-bench: Benchmarking multimodal llms with generative comprehension. *arXiv:2307.16125*, 2023.
- [124] 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*, 2023.
- [125] Kunchang Li, Yanan 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.
- [126] Kunchang Li, Yali Wang, Yanan 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*, 2024.
- [127] Lei Li, Yuwei Yin, Shicheng Li, Liang Chen, Peiyi Wang, Shuhuai Ren, Mukai Li, Yazheng Yang, Jingjing Xu, Xu Sun, et al. M³it: A large-scale dataset towards multi-modal multilingual instruction tuning. *arXiv:2306.04387*, 2023.
- [128] Pandeng Li, Chen-Wei Xie, Hongtao Xie, Liming Zhao, Lei Zhang, Yun Zheng, Deli Zhao, and Yongdong Zhang. Momentdiff: Generative video moment retrieval from random to real. In *NeurIPS*, 2023.
- [129] Shuaicheng Li, Feng Zhang, Kunlin Yang, Lingbo Liu, Shinan Liu, Jun Hou, and Shuai Yi. Probing visual-audio representation for video highlight detection via hard-pairs guided contrastive learning. *arXiv:2206.10157*, 2022.
- [130] Shuailin Li, Yuang Zhang, Yucheng Zhao, Qiuyue Wang, Fan Jia, Yingfei Liu, and Tiancai Wang. Vlm-eval: A general evaluation on video large language models. *arXiv:2311.11865*, 2023.

References

- [131] Yanwei Li, Chengyao Wang, and Jiaya Jia. Llama-vid: An image is worth 2 tokens in large language models. *arXiv:2311.17043*, 2023.
- [132] Zeqian Li, Qirui Chen, Tengda Han, Ya Zhang, Yanfeng Wang, and Weidi Xie. A strong baseline for temporal video-text alignment. *arXiv:2312.14055*, 2023.
- [133] 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.
- [134] Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let’s verify step by step. In *ICLR*, 2023.
- [135] 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.
- [136] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, 2004.
- [137] Ji Lin, Hongxu Yin, Wei Ping, Pavlo Molchanov, Mohammad Shoeybi, and Song Han. Vila: On pre-training for visual language models. In *CVPR*, 2024.
- [138] Jingyang Lin, Hang Hua, Ming Chen, Yikang Li, Jenhao Hsiao, Chiuman Ho, and Jiebo Luo. Videoxum: Cross-modal visual and textural summarization of videos. *IEEE Transactions on Multimedia*, 2023.
- [139] Kevin Qinghong Lin, Jinpeng Wang, Mattia Soldan, Michael Wray, Rui Yan, Eric Z Xu, Difei Gao, Rong-Cheng Tu, Wenzhe Zhao, Weijie Kong, et al. Egocentric video-language pretraining. In *NeurIPS*, 2022.
- [140] Kevin Qinghong Lin, Pengchuan Zhang, Joya Chen, Shraman Pramanick, Difei Gao, Alex Jinpeng Wang, Rui Yan, and Mike Zheng Shou. Univtg: Towards unified video-language temporal grounding. In *CVPR*, 2023.

-
- [141] Kevin Qinghong Lin, Pengchuan Zhang, Difei Gao, Xide Xia, Joya Chen, Ziteng Gao, Jinheng Xie, Xuhong Xiao, and Mike Zheng Shou. Learning video context as interleaved multimodal sequences. In *ECCV*, 2024.
- [142] 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.
- [143] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017.
- [144] 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*, 2014.
- [145] Ziyi Lin, Shijie Geng, Renrui Zhang, Peng Gao, Gerard de Melo, Xiaogang Wang, Jifeng Dai, Yu Qiao, and Hongsheng Li. Frozen clip models are efficient video learners. In *ECCV*, 2022.
- [146] Chang Liu, Henghui Ding, and Xudong Jiang. Gres: Generalized referring expression segmentation. In *CVPR*, 2023.
- [147] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *CVPR*, 2024.
- [148] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023.
- [149] Jiang Liu, Hui Ding, Zhaowei Cai, Yuting Zhang, Ravi Kumar Satzoda, Vijay Mahadevan, and R Manmatha. Polyformer: Referring image segmentation as sequential polygon generation. In *CVPR*, 2023.
- [150] Lingbo Liu, Bruce XB Yu, Jianlong Chang, Qi Tian, and Chang Wen Chen. Prompt-matched semantic segmentation. *arXiv:2208.10159*, 2022.
- [151] Ruyang Liu, Chen Li, Haoran Tang, Yixiao Ge, Ying Shan, and Ge Li. St-llm: Large language models are effective temporal learners. In *ECCV*, 2024.

References

- [152] Wu Liu, Tao Mei, Yongdong Zhang, Cherry Che, and Jiebo Luo. Multi-task deep visual-semantic embedding for video thumbnail selection. In *CVPR*, 2015.
- [153] 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.
- [154] 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.
- [155] 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*, 2022.
- [156] 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.
- [157] Ye Liu, Zongyang Ma, Junfu Pu, Zhongang Qi, Yang Wu, Shan Ying, and Chang Wen Chen. Unipixel: Unified object referring and segmentation for pixel-level visual reasoning. In *NeurIPS*, 2025.
- [158] 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*, 2024.
- [159] Ye Liu, Junsong Yuan, and Chang Wen Chen. Consnet: Learning consistency graph for zero-shot human-object interaction detection. In *ACM MM*, 2020.
- [160] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? *arXiv:2307.06281*, 2023.
- [161] Yuanxin Liu, Shicheng Li, Yi Liu, Yuxiang Wang, Shuhuai Ren, Lei Li, Sishuo Chen, Xu Sun, and Lu Hou. Tempcompass: Do video llms really understand videos? *arXiv:2403.00476*, 2024.

-
- [162] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.
- [163] Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *NeurIPS*, 2022.
- [164] Dezhao Luo, Jiabo Huang, Shaogang Gong, Hailin Jin, and Yang Liu. Towards generalisable video moment retrieval: Visual-dynamic injection to image-text pre-training. In *CVPR*, 2023.
- [165] Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. Clip4clip: An empirical study of clip for end to end video clip retrieval. *Neurocomputing*, 508:293–304, 2022.
- [166] Ruipu Luo, Ziwang Zhao, Min Yang, Junwei Dong, Da Li, Pengcheng Lu, Tao Wang, Linmei Hu, Minghui Qiu, and Zhongyu Wei. Valley: Video assistant with large language model enhanced ability. *arXiv:2306.07207*, 2023.
- [167] Chuofan Ma, Yi Jiang, Jiannan Wu, Zehuan Yuan, and Xiaojuan Qi. Groma: Localized visual tokenization for grounding multimodal large language models. *arXiv:2404.13013*, 2024.
- [168] 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.
- [169] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Khan. Videogpt+: Integrating image and video encoders for enhanced video understanding. *arXiv:2406.09418*, 2024.
- [170] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Khan. Videogpt+: Integrating image and video encoders for enhanced video understanding. *arXiv:2406.09418*, 2024.

References

- [171] 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.
- [172] Behrooz Mahasseni, Michael Lam, and Sinisa Todorovic. Unsupervised video summarization with adversarial lstm networks. In *CVPR*, 2017.
- [173] Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. Egoschema: A diagnostic benchmark for very long-form video language understanding. In *NeurIPS*, 2024.
- [174] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *CVPR*, 2016.
- [175] Maurice Merleau-Ponty. *Phenomenology of perception*. Gallimard, 1945.
- [176] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *ICCV*, 2019.
- [177] WonJun Moon, Sangeek Hyun, SuBeen Lee, and Jae-Pil Heo. Correlation-guided query-dependency calibration in video representation learning for temporal grounding. *arXiv:2311.08835*, 2023.
- [178] WonJun Moon, Sangeek Hyun, SangUk Park, Dongchan Park, and Jae-Pil Heo. Query-dependent video representation for moment retrieval and highlight detection. In *CVPR*, 2023.
- [179] 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.
- [180] Arsha Nagrani, Shan Yang, Anurag Arnab, Aren Jansen, Cordelia Schmid, and Chen Sun. Attention bottlenecks for multimodal fusion. In *NeurIPS*, 2021.

-
- [181] Ulric Neisser. *Cognition and reality: principles and implications of cognitive psychology*. W. H. Freeman and Company, 1976.
- [182] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Buló, and Peter Kotschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *ICCV*, 2017.
- [183] Bolin Ni, Houwen Peng, Minghao Chen, Songyang Zhang, Gaofeng Meng, Jianlong Fu, Shiming Xiang, and Haibin Ling. Expanding language-image pretrained models for general video recognition. In *ECCV*, 2022.
- [184] Munan Ning, Bin Zhu, Yujia Xie, Bin Lin, Jiaxi Cui, Lu Yuan, Dongdong Chen, and Li Yuan. Video-bench: A comprehensive benchmark and toolkit for evaluating video-based large language models. *arXiv:2311.16103*, 2023.
- [185] Andreea-Maria Oncescu, Joao F Henriques, Yang Liu, Andrew Zisserman, and Samuel Albanie. Queryd: A video dataset with high-quality text and audio narrations. In *ICASSP*, 2021.
- [186] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. In *NeurIPS*, 2018.
- [187] OpenAI. Gpt-4v(ision) system card, 2023.
- [188] OpenAI. Gpt-4o system card, 2024.
- [189] OpenAI. Openai o1 system card, 2024.
- [190] OpenGVLab. Internvl2: Better than the best—expanding performance boundaries of open-source multimodal models with the progressive scaling strategy, 2024.
- [191] Mayu Otani, Yuta Nakashima, Esa Rahtu, Janne Heikkilä, and Naokazu Yokoya. Learning joint representations of videos and sentences with web image search. In *ECCV*, 2016.
- [192] Junting Pan, Ziyi Lin, Xiatian Zhu, Jing Shao, and Hongsheng Li. St-adapter: Parameter-efficient image-to-video transfer learning. In *NeurIPS*, 2022.

References

- [193] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002.
- [194] Viorica Patraucean, Lucas Smaira, Ankush Gupta, Adria Recasens, Larisa Markeeva, Dylan Banarse, Skanda Koppula, Mateusz Malinowski, Yi Yang, Carl Doersch, et al. Perception test: A diagnostic benchmark for multimodal video models. In *NeurIPS*, 2024.
- [195] 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.
- [196] Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, and Furu Wei. Kosmos-2: Grounding multimodal large language models to the world. *arXiv:2306.14824*, 2023.
- [197] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014.
- [198] Aleksandar Petrov, Philip HS Torr, and Adel Bibi. When do prompting and prefix-tuning work? a theory of capabilities and limitations. *arXiv:2310.19698*, 2023.
- [199] 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.
- [200] Long Qian, Juncheng Li, Yu Wu, Yaobo Ye, Hao Fei, Tat-Seng Chua, Yueting Zhuang, and Siliang Tang. Momentor: Advancing video large language model with fine-grained temporal reasoning. *arXiv:2402.11435*, 2024.
- [201] Rui Qian, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Shuangrui Ding, Dahua Lin, and Jiaqi Wang. Streaming long video understanding with large language models. In *NeurIPS*, 2024.

-
- [202] Zhiwu Qing, Shiwei Zhang, Ziyuan Huang, Yingya Zhang, Changxin Gao, Deli Zhao, and Nong Sang. Disentangling spatial and temporal learning for efficient image-to-video transfer learning. In *CVPR*, 2023.
- [203] 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.
- [204] Mengxue Qu, Xiaodong Chen, Wu Liu, Alicia Li, and Yao Zhao. Chatvtg: Video temporal grounding via chat with video dialogue large language models. In *CVPR*, 2024.
- [205] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.
- [206] Santhosh Kumar Ramakrishnan, Ziad Al-Halah, and Kristen Grauman. Naq: Leveraging narrations as queries to supervise episodic memory. In *CVPR*, 2023.
- [207] 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*, 2023.
- [208] Hanoona Rasheed, Muhammad Uzair Khattak, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Fine-tuned clip models are efficient video learners. In *CVPR*, 2023.
- [209] Hanoona Rasheed, Muhammad Maaz, Sahal Shaji, Abdelrahman Shaker, Salman Khan, Hisham Cholakkal, Rao M Anwer, Eric Xing, Ming-Hsuan Yang, and Fahad S Khan. Glamm: Pixel grounding large multimodal model. In *CVPR*, 2024.
- [210] 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.

References

- [211] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016.
- [212] Michaela Regneri, Marcus Rohrbach, Dominikus Wetzels, Stefan Thater, Bernt Schiele, and Manfred Pinkal. Grounding action descriptions in videos. *Transactions of the Association for Computational Linguistics*, 1:25–36, 2013.
- [213] 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.
- [214] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv:1908.10084*, 2019.
- [215] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015.
- [216] 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*, 2024.
- [217] Zhongwei Ren, Zhicheng Huang, Yunchao Wei, Yao Zhao, Dongmei Fu, Jiashi Feng, and Xiaojie Jin. Pixellm: Pixel reasoning with large multimodal model. In *CVPR*, 2024.
- [218] Mrigank Rochan, Mahesh Kumar Krishna Reddy, and Yang Wang. Sentence guided temporal modulation for dynamic video thumbnail generation. In *BMVC*, 2020.
- [219] 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*, 2023.
- [220] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv:1707.06347*, 2017.

-
- [221] Seonguk Seo, Joon-Young Lee, and Bohyung Han. Urvos: Unified referring video object segmentation network with a large-scale benchmark. In *ECCV*, 2020.
- [222] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *NeurIPS*, 2023.
- [223] Zheng Shou, Dongang Wang, and Shih-Fu Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *CVPR*, 2016.
- [224] Nina Shvetsova, Anna Kukleva, Xudong Hong, Christian Rupprecht, Bernt Schiele, and Hilde Kuehne. Howtocation: Prompting llms to transform video annotations at scale. *arXiv:2310.04900*, 2023.
- [225] Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *ECCV*, 2016.
- [226] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.
- [227] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.
- [228] Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Xun Guo, Tian Ye, Yan Lu, Jenq-Neng Hwang, et al. Moviechat: From dense token to sparse memory for long video understanding. *arXiv:2307.16449*, 2023.
- [229] Yale Song, Miriam Redi, Jordi Vallmitjana, and Alejandro Jaimes. To click or not to click: Automatic selection of beautiful thumbnails from videos. In *CIKM*, 2016.
- [230] Yale Song, Jordi Vallmitjana, Amanda Stent, and Alejandro Jaimes. Tvsum: Summarizing web videos using titles. In *CVPR*, 2015.

References

- [231] Nicholas Stroh. Trackgpt—a generative pre-trained transformer for cross-domain entity trajectory forecasting. *arXiv:2402.00066*, 2024.
- [232] Jinhwan Sul, Jihoon Han, and Joonseok Lee. Mr. hisum: A large-scale dataset for video highlight detection and summarization. In *NeurIPS*, 2024.
- [233] Hao Sun, Mingyao Zhou, Wenjing Chen, and Wei Xie. Tr-detr: Task-reciprocal transformer for joint moment retrieval and highlight detection. In *AAAI*, 2024.
- [234] Min Sun, Ali Farhadi, and Steve Seitz. Ranking domain-specific highlights by analyzing edited videos. In *ECCV*, 2014.
- [235] Yi-Lin Sung, Jaemin Cho, and Mohit Bansal. Lst: Ladder side-tuning for parameter and memory efficient transfer learning. In *NeurIPS*, 2022.
- [236] Didac Suris, Sachit Menon, and Carl Vondrick. Vipergpt: Visual inference via python execution for reasoning. In *ICCV*, 2023.
- [237] 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*, 2020.
- [238] Hao Tang, Vivek Kwatra, Mehmet Emre Sargin, and Ullas Gargi. Detecting highlights in sports videos: Cricket as a test case. In *ICME*, 2011.
- [239] Yansong Tang, Dajun Ding, Yongming Rao, Yu Zheng, Danyang Zhang, Lili Zhao, Jiwen Lu, and Jie Zhou. Coin: A large-scale dataset for comprehensive instructional video analysis. In *CVPR*, 2019.
- [240] MiniCPM-V Team. Minicpm-llama3-v 2.5: A gpt-4v level multimodal llm on your phone, 2024.
- [241] Stefanie Tellex and Deb Roy. Towards surveillance video search by natural language query. In *CIVR*, 2009.

-
- [242] Siyu Teng, Xuemin Hu, Peng Deng, Bai Li, Yuchen Li, Yunfeng Ai, Dongsheng Yang, Lingxi Li, Zhe Xuanyuan, Fenghua Zhu, et al. Motion planning for autonomous driving: The state of the art and future perspectives. *IEEE Transactions on Intelligent Vehicles*, 8(6):3692–3711, 2023.
- [243] Omkar Thawakar, Dinura Dissanayake, Ketan More, Ritesh Thawkar, Ahmed Heakl, Noor Ahsan, Yuhao Li, Mohammed Zumri, Jean Lahoud, Rao Muhammad Anwer, et al. Llamav-01: Rethinking step-by-step visual reasoning in llms. *arXiv:2501.06186*, 2025.
- [244] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *ICCV*, 2019.
- [245] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv:2302.13971*, 2023.
- [246] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv:2307.09288*, 2023.
- [247] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015.
- [248] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.
- [249] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *CVPR*, 2015.
- [250] Ante Wang, Linfeng Song, Ye Tian, Baolin Peng, Dian Yu, Haitao Mi, Jinsong Su, and Dong Yu. Litesearch: Efficacious tree search for llm. *arXiv:2407.00320*, 2024.

References

- [251] Han Wang, Yongjie Ye, Yanjie Wang, Yuxiang Nie, and Can Huang. Elysium: Exploring object-level perception in videos via mllm. In *ECCV*, 2024.
- [252] Haochen Wang, Cilin Yan, Shuai Wang, Xiaolong Jiang, Xu Tang, Yao Hu, Weidi Xie, and Efstratios Gavves. Towards open-vocabulary video instance segmentation. In *ICCV*, 2023.
- [253] Jinjun Wang, Changsheng Xu, Engsiong Chng, and Qi Tian. Sports highlight detection from keyword sequences using hmm. In *ICME*, 2004.
- [254] Lezi Wang, Dong Liu, Rohit Puri, and Dimitris N. Metaxas. Learning trailer moments in full-length movies. In *ECCV*, 2020.
- [255] 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.
- [256] 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*, 2022.
- [257] 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.
- [258] Weihang Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Xiaotao Gu, Shiyu Huang, Bin Xu, Yuxiao Dong, et al. Lvbench: An extreme long video understanding benchmark. *arXiv:2406.08035*, 2024.
- [259] Weining Wang, Yan Huang, and Liang Wang. Language-driven temporal activity localization: A semantic matching reinforcement learning model. In *CVPR*, 2019.

-
- [260] Wenhai Wang, Zhe Chen, Xiaokang Chen, Jiannan Wu, Xizhou Zhu, Gang Zeng, Ping Luo, Tong Lu, Jie Zhou, Yu Qiao, et al. Visionllm: Large language model is also an open-ended decoder for vision-centric tasks. In *NeurIPS*, 2024.
- [261] Xiaohan Wang, Yuhui Zhang, Orr Zohar, and Serena Yeung-Levy. Videoagent: Long-form video understanding with large language model as agent. *arXiv:2403.10517*, 2024.
- [262] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *ECCV Workshop*, 2018.
- [263] Xiyao Wang, Linfeng Song, Ye Tian, Dian Yu, Baolin Peng, Haitao Mi, Furong Huang, and Dong Yu. Towards self-improvement of llms via mcts: Leveraging stepwise knowledge with curriculum preference learning. *arXiv:2410.06508*, 2024.
- [264] Xiyao Wang, Ruijie Zheng, Yanchao Sun, Ruonan Jia, Wichayaporn Wongkamjan, Huazhe Xu, and Furong Huang. Coplanner: Plan to roll out conservatively but to explore optimistically for model-based rl. *arXiv:2310.07220*, 2023.
- [265] Yi Wang, Yinan He, Yizhuo Li, Kunchang Li, Jiashuo Yu, Xin Ma, Xinhao Li, Guo Chen, Xinyuan Chen, Yaohui Wang, et al. Internvid: A large-scale video-text dataset for multimodal understanding and generation. *arXiv:2307.06942*, 2023.
- [266] Yueqian Wang, Xiaojun Meng, Jianxin Liang, Yuxuan Wang, Qun Liu, and Dongyan Zhao. Hawkeye: Training video-text llms for grounding text in videos. *arXiv:2403.10228*, 2024.
- [267] Zhenzhi Wang, Limin Wang, Tao Wu, Tianhao Li, and Gangshan Wu. Negative sample matters: A renaissance of metric learning for temporal grounding. In *AAAI*, 2022.
- [268] Cong Wei, Haoxian Tan, Yujie Zhong, Yujiu Yang, and Lin Ma. Lasagna: Language-based segmentation assistant for complex queries. *arXiv:2404.08506*, 2024.

References

- [269] 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.
- [270] 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.
- [271] Fanyue Wei, Biao Wang, Tiezheng Ge, Yuning Jiang, Wen Li, and Lixin Duan. Learning pixel-level distinctions for video highlight detection. In *CVPR*, 2022.
- [272] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*, 2022.
- [273] Bo Wu, Shoubin Yu, Zhenfang Chen, Joshua B Tenenbaum, and Chuang Gan. Star: A benchmark for situated reasoning in real-world videos. *arXiv:2405.09711*, 2024.
- [274] Dongming Wu, Tiancai Wang, Yuang Zhang, Xiangyu Zhang, and Jianbing Shen. Onlinerefer: A simple online baseline for referring video object segmentation. In *ICCV*, 2023.
- [275] Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. Longvideobench: A benchmark for long-context interleaved video-language understanding. In *NeurIPS*, 2024.
- [276] 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.
- [277] Jiannan Wu, Yi Jiang, Peize Sun, Zehuan Yuan, and Ping Luo. Language as queries for referring video object segmentation. In *CVPR*, 2022.
- [278] Jiannan Wu, Yi Jiang, Bin Yan, Huchuan Lu, Zehuan Yuan, and Ping Luo. Segment every reference object in spatial and temporal spaces. In *ICCV*, 2023.

-
- [279] Zhuofan Xia, Dongchen Han, Yizeng Han, Xuran Pan, Shiji Song, and Gao Huang. Gsva: Generalized segmentation via multimodal large language models. In *CVPR*, 2024.
- [280] Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. Next-qa: Next phase of question-answering to explaining temporal actions. In *CVPR*, 2021.
- [281] Junbin Xiao, Angela Yao, Yicong Li, and Tat-Seng Chua. Can i trust your answer? visually grounded video question answering. In *CVPR*, 2024.
- [282] 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.
- [283] 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.
- [284] 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.
- [285] 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.
- [286] 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.
- [287] Bo Xiong, Yannis Kalantidis, Deepti Ghadiyaram, and Kristen Grauman. Less is more: Learning highlight detection from video duration. In *CVPR*, 2019.

References

- [288] 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*, 2017.
- [289] Guowei Xu, Peng Jin, Li Hao, Yibing Song, Lichao Sun, and Li Yuan. Llava-cot: Let vision language models reason step-by-step. In *ICCV*, 2025.
- [290] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *CVPR*, 2016.
- [291] 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.
- [292] Minghao Xu, Hang Wang, Bingbing Ni, Riheng Zhu, Zhenbang Sun, and Changhu Wang. Cross-category video highlight detection via set-based learning. In *ICCV*, 2021.
- [293] Yi Xu, Fan Bai, Yingxuan Shi, Qiuyu Chen, Longwen Gao, Kai Tian, Shuigeng Zhou, and Huyang Sun. Gif thumbnails: Attract more clicks to your videos. In *AAAI*, 2021.
- [294] Yifang Xu, Yunzhuo Sun, Benxiang Zhai, Youyao Jia, and Sidan Du. Mh-detr: Video moment and highlight detection with cross-modal transformer. In *IJCNN*, 2024.
- [295] Yuancheng Xu, Udari Madhushani Sehwag, Alec Koppel, Sicheng Zhu, Bang An, Furong Huang, and Sumitra Ganesh. Genarm: Reward guided generation with autoregressive reward model for test-time alignment. *arXiv:2410.08193*, 2024.
- [296] 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*, 2023.
- [297] 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*, 2024.

-
- [298] Shen Yan, Xuehan Xiong, Arsha Nagrani, Anurag Arnab, Zhonghao Wang, Weina Ge, David Ross, and Cordelia Schmid. Unloc: A unified framework for video localization tasks. In *ICCV*, 2023.
- [299] Ziang Yan, Zhilin Li, Yinan He, Chenting Wang, Kunchang Li, Xinhao Li, Xiangyu Zeng, Zilei Wang, Yali Wang, Yu Qiao, et al. Task preference optimization: Improving multimodal large language models with vision task alignment. *arXiv:2412.19326*, 2024.
- [300] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Zero-shot video question answering via frozen bidirectional language models. *Advances in Neural Information Processing Systems*, 35:124–141, 2022.
- [301] Huan Yang, Baoyuan Wang, Stephen Lin, David Wipf, Minyi Guo, and Baining Guo. Unsupervised extraction of video highlights via robust recurrent auto-encoders. In *ICCV*, 2015.
- [302] 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.
- [303] Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Ehsan Azarnasab, Faisal Ahmed, Zicheng Liu, Ce Liu, Michael Zeng, and Lijuan Wang. Mm-react: Prompting chatgpt for multimodal reasoning and action. *arXiv:2303.11381*, 2023.
- [304] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *NeurIPS*, 2023.
- [305] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *ICLR*, 2023.
- [306] Ting Yao, Tao Mei, and Yong Rui. Highlight detection with pairwise deep ranking for first-person video summarization. In *CVPR*, 2016.

References

- [307] Qinghao Ye, Xiyue Shen, Yuan Gao, Zirui Wang, Qi Bi, Ping Li, and Guang Yang. Temporal cue guided video highlight detection with low-rank audio-visual fusion. In *ICCV*, 2021.
- [308] Qinghao Ye, Haiyang Xu, Jiabo Ye, Ming Yan, Haowei Liu, Qi Qian, Ji Zhang, Fei Huang, and Jingren Zhou. mplug-owl2: Revolutionizing multi-modal large language model with modality collaboration. *arXiv:2311.04257*, 2023.
- [309] Dongshuo Yin, Xueting Han, Bin Li, Hao Feng, and Jing Bai. Parameter-efficient is not sufficient: Exploring parameter, memory, and time efficient adapter tuning for dense predictions. In *ACM MM*, 2024.
- [310] Kaining Ying, Fanqing Meng, Jin Wang, Zhiqian Li, Han Lin, Yue Yang, Hao Zhang, Wenbo Zhang, Yuqi Lin, Shuo Liu, et al. Mmt-bench: A comprehensive multi-modal benchmark for evaluating large vision-language models towards multitask agi. *arXiv:2404.16006*, 2024.
- [311] 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.
- [312] Samson Yu, Kelvin Lin, Anxing Xiao, Jiafei Duan, and Harold Soh. Octopi: Object property reasoning with large tactile-language models. *arXiv:2405.02794*, 2024.
- [313] Shoubin Yu, Jaemin Cho, Prateek Yadav, and Mohit Bansal. Self-chained image-language model for video localization and question answering. In *NeurIPS*, 2023.
- [314] 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*, 2019.
- [315] 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.

-
- [316] Tongtong Yuan, Xuange Zhang, Kun Liu, Bo Liu, Jian Jin, and Zhenzhen Jiao. Ucf-crime annotation: A benchmark for surveillance video-and-language understanding. *arXiv:2309.13925*, 2023.
- [317] Yitian Yuan, Lin Ma, and Wenwu Zhu. Sentence specified dynamic video thumbnail generation. In *ACM MM*, 2019.
- [318] 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*, 2024.
- [319] 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.
- [320] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. *arXiv:2311.16502*, 2023.
- [321] Abhay Zala, Jaemin Cho, Satwik Kottur, Xilun Chen, Barlas Oguz, Yashar Mehdad, and Mohit Bansal. Hierarchical video-moment retrieval and step-captioning. In *CVPR*, 2023.
- [322] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *ICCV*, 2023.
- [323] 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.
- [324] Ce Zhang, Taixi Lu, Md Mohaiminul Islam, Ziyang Wang, Shoubin Yu, Mohit Bansal, and Gedas Bertasius. A simple llm framework for long-range video question-answering. *arXiv:2312.17235*, 2023.
- [325] Chen-Lin Zhang, Jianxin Wu, and Yin Li. Actionformer: Localizing moments of actions with transformers. In *ECCV*, 2022.

References

- [326] Da Zhang, Xiyang Dai, Xin Wang, Yuan-Fang Wang, and Larry S. Davis. Man: Moment alignment network for natural language moment retrieval via iterative graph adjustment. In *CVPR*, 2019.
- [327] Dan Zhang, Sining Zhoubian, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. Restmcts*: Llm self-training via process reward guided tree search. *arXiv:2406.03816*, 2024.
- [328] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv:2306.02858*, 2023.
- [329] Hao Zhang, Aixin Sun, Wei Jing, and Joey Tianyi Zhou. Span-based localizing network for natural language video localization. *arXiv:2004.13931*, 2020.
- [330] Jiawei Zhang, Chejian Xu, and Bo Li. Chatscene: Knowledge-enabled safety-critical scenario generation for autonomous vehicles. In *CVPR*, 2024.
- [331] Ke Zhang, Wei-Lun Chao, Fei Sha, and Kristen Grauman. Video summarization with long short-term memory. In *ECCV*, 2016.
- [332] Pan Zhang, Xiaoyi Dong Bin Wang, Yuhang Cao, Chao Xu, Linke Ouyang, Zhiyuan Zhao, Shuangrui Ding, Songyang Zhang, Haodong Duan, Hang Yan, et al. Internlm-xcomposer: A vision-language large model for advanced text-image comprehension and composition. *arXiv:2309.15112*, 2023.
- [333] Peiyuan Zhang, Kaichen Zhang, Bo Li, Guangtao Zeng, Jingkan Yang, Yuanhan Zhang, Ziyue Wang, Haoran Tan, Chunyuan Li, and Ziwei Liu. Long context transfer from language to vision. *arXiv:2406.16852*, 2024.
- [334] Songyang Zhang, Houwen Peng, Jianlong Fu, and Jiebo Luo. Learning 2d temporal adjacent networks for moment localization with natural language. In *AAAI*, 2020.
- [335] 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*, 2024.

-
- [336] Yichi Zhang, Ziqiao Ma, Xiaofeng Gao, Suhaila Shakiah, Qiaozi Gao, and Joyce Chai. Groundhog: Grounding large language models to holistic segmentation. In *CVPR*, 2024.
- [337] Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. Multimodal chain-of-thought reasoning in language models. *arXiv:2302.00923*, 2023.
- [338] Hang Zhao, Antonio Torralba, Lorenzo Torresani, and Zhicheng Yan. Hacs: Human action clips and segments dataset for recognition and temporal localization. In *ICCV*, 2019.
- [339] 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*, 2023.
- [340] Yue Zhao, Ishan Misra, Philipp Krahenbuhl, and Rohit Girdhar. Learning video representations from large language models. In *CVPR*, 2023.
- [341] 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.
- [342] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, 2017.
- [343] Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Shitao Xiao, Xi Yang, Yongping Xiong, Bo Zhang, Tiejun Huang, and Zheng Liu. Mlvu: A comprehensive benchmark for multi-task long video understanding. *arXiv:2406.04264*, 2024.
- [344] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *CVPR*, 2022.
- [345] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022.

References

- [346] Luowei Zhou, Chenliang Xu, and Jason Corso. Towards automatic learning of procedures from web instructional videos. In *AAAI*, 2018.
- [347] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. In *CVPR*, 2019.
- [348] Bin Zhu, Bin Lin, Munan Ning, Yang Yan, Jiayi Cui, HongFa Wang, Yatian Pang, Wenhao Jiang, Junwu Zhang, Zongwei Li, et al. Languagebind: Extending video-language pretraining to n-modality by language-based semantic alignment. *arXiv:2310.01852*, 2023.
- [349] 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.
- [350] Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Yuchen Duan, Hao Tian, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv:2504.10479*, 2025.
- [351] Dimitri Zhukov, Jean-Baptiste Alayrac, Ramazan Gokberk Cinbis, David Fouhey, Ivan Laptev, and Josef Sivic. Cross-task weakly supervised learning from instructional videos. In *CVPR*, 2019.
- [352] 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*, 2023.
- [353] 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*, 2023.