Surgical Video Temporal Grounding

Qixuan Liu[1,2], Shi Qiu[1,2] (✉), Xiwen Wu[3], Binzhu Xie[1,2], Chi-Wing Fu[1,2], and Pheng-Ann Heng[1,2]

[1] Dept. of Computer Science and Engineering, The Chinese University of Hong Kong
[2] Institute of Medical Intelligence and XR, The Chinese University of Hong Kong
[3] Zhujiang Hospital, Southern Medical University
{qxliu, shiqiu}@cse.cuhk.edu.hk

Abstract. Temporal grounding in surgical videos is essential for clinical applications, yet this area remains significantly underexplored. Furthermore, existing methods designed for general-domain videos struggle to generalize to the surgical domain due to two unique challenges. First, surgical procedures exhibit a complex inherent hierarchy, ranging from coarse phases to fine-grained actions. Second, processing long surgical videos in discrete chunks disrupts continuous context, leading to temporal isolation. To address these issues, we formally introduce the research problem of Surgical Video Temporal Grounding (SurgVTG) and formulate a novel task of Hierarchical Multi-Segment SurgVTG (HMS-SurgVTG), accompanied by a comprehensive benchmark. To tackle this task, we propose a Memory-Augmented Vision-Language Model framework that mitigates cross-chunk isolation through an explicit read-write memory loop. By dynamically updating and injecting structured procedural memories via lightweight heads, our framework seamlessly preserves historical context during scalable inference. Extensive experiments demonstrate that our approach enhances long-range temporal grounding quality, establishing a new benchmark for surgical video temporal grounding. Code is available at `https://github.com/Mr-Kill/SurgicalVideoTemporalGrounding`.

Keywords: Surgical Video · Temporal Grounding · VLMs.

1 Introduction

Surgical videos are a valuable resource for clinical practice, education, and research [14]. Clinical tasks depend on localizing temporal segments: generating structured surgical reports requires identifying each workflow phase and its duration [5]; reviewing complications involves pinpointing abnormally prolonged phases or unexpected procedural deviations [1]; and surgical trainees benefit from rapidly retrieving technique-specific segments for self-directed learning [13, 8]. Despite their importance, these tasks still rely heavily on manual navigation. In practice, clinicians and researchers need to review hours of footage to find the segments of interest, which is often prohibitively time-consuming. This bottleneck motivates us to investigate *Surgical Video Temporal Grounding* (SurgVTG),

a novel research problem of automatically localizing temporal segments in a surgical video that match a natural-language query.

Compared with general-domain videos, surgical videos exhibit substantial differences in workflow organization, event semantics, and long-range temporal dependencies, making existing temporal grounding approaches [9, 10] difficult to apply directly. One core difference is the inherent compositional hierarchy of surgical workflows. Rather than a flat collection of actions, a surgical workflow is a structured process in which coarse-grained surgical phases (or steps) organize instrument usage [3, 4, 7], which in turn gives rise to fine-grained action triplets, e.g., ⟨Instrument, Verb, Target⟩ [15, 2]. This hierarchy is important both clinically and semantically: the same instrument action can correspond to different intents, risks, and procedural meanings under different phase contexts. As a result, a flat temporal grounding formulation is insufficient, as it tends to conflate events that are visually similar but procedurally distinct. These observations motivate a hierarchical formulation of surgical temporal grounding that comprehensively represents procedural structure.

Another issue lies in the long-form nature of the surgical videos, which are typically long but require the precise localization of fine-grained action segment boundaries. Although recent VTG methods introduce token compression and efficient context modeling to accommodate longer sequences [16, 21], applying them to hour-long surgical procedures still necessitates extensive spatio-temporal pooling or sparse sampling. Such heavy compression inevitably distorts high-frequency temporal dynamics and obscures the fine-grained visual cues essential for precise surgical event grounding. Moreover, chunk-based video processing is more scalable in practice, but introduces a key limitation: each chunk is often processed independently, preventing the model from reasoning over cross-chunk continuity [20], e.g., whether an observed instrument action is a continuation of a previous event or the start of a new occurrence. This temporal isolation issue is particularly severe in surgery, where procedural interpretation often depends on accumulated workflow context rather than a single local observation.

To deeply investigate the issues and address them, we present a comprehensive solution set that formulates the problem, establishes a benchmark, and designs a dedicated framework. Our contributions are threefold: **(i)** We define *Surgical Video Temporal Grounding* (SurgVTG) as a novel research problem, identifying the unique requirements that set it apart from general-domain video temporal grounding. **(ii)** We introduce *Hierarchical Multi-Segment Surgical Video Temporal Grounding* (HMS-SurgVTG), a task benchmark that instantiates SurgVTG with three hierarchically nested grounding levels, phase (L_1), instrument (L_2), and action triplet (L_3), together with curated annotations and evaluation protocols derived from CholecT50 [15]. **(iii)** We propose a memory-augmented Vision-Language Model (VLM) framework that processes long surgical videos in chunks for scalability, while maintaining cross-chunk procedural context through an explicit read–write memory loop.

2 Method

2.1 HMS-SurgVTG Formulation

Task Objective. We formulate the *Hierarchical Multi-Segment Surgical Video Temporal Grounding* (HMS-SurgVTG) task as follows. Given an untrimmed surgical video V of duration T and a natural-language query q, the objective is to predict all non-overlapping temporal moments $\mathcal{Y} = \{(s_k, e_k)\}_{k=1}^{K}$, where each (s_k, e_k) pair denotes a start and end timestamp, such that each segment localises one occurrence of the event described by q.

Hierarchical Grounding Levels. Surgical procedures exhibit an inherent compositional structure in which a *phase* (or *step*) decomposes into instrument activations that in turn give rise to fine-grained action triplets ⟨Instrument, Verb, Target⟩. We represent this hierarchy in three grounding levels, where the natural-language query progressively incorporates stronger hierarchical context constraints: **(i)** *Phase-level* (L_1): the query $q^{(1)}$ describes a surgical phase (e.g., *"The surgery is in the clipping-and-cutting phase."*) and the search scope is full video. **(ii)** *Instrument-level* (L_2): the query $q^{(2)}$ asks when a specific instrument is active (e.g., *"During the gallbladder-packaging phase, the grasper is being used."*) *within* the parent phase's predicted segments $\hat{\mathcal{Y}}^{(1)}$. **(iii)** *Triplet-level* (L_3): the query $q^{(3)}$ targets a fine-grained action triplet (e.g., *"During the calot-triangle-dissection phase, the grasper is used to retract the omentum."*) *within* the parent instrument's predicted segments $\hat{\mathcal{Y}}^{(2)}$. Formally, the search scope at level $\ell > 1$ is constrained to the predictions of the parent level $\hat{\mathcal{Y}}^{(\ell-1)}$, yielding a coarse-to-fine grounding cascade that mirrors a surgeon's reasoning during an operation.

2.2 Memory-Augmented VLM Framework

We process surgical videos in non-overlapping chunks to keep memory bounded for hour-long procedures, while decoding all timestamps on the original video timeline. As shown in Fig. 1, our framework builds on a Vision Language Model (VLM) comprising a vision encoder $\mathcal{V}$ and an auto-regressive Large Language Model (LLM) $\mathcal{G}_\theta$, where θ denotes trainable LoRA [6] modules.

Timestamp-Interleaved Visual Sequence. Following [10], we encode temporal information by interleaving timestamp (e.g., $1.0s$) text tokens with visual frame tokens. Given chunk t containing N sampled frames at timestamps $(t_1, \ldots, t_N)$, each frame f_i is encoded by the vision encoder $\mathcal{V}$ and paired with its text-tokenised timestamp τ_i, forming a timestamp-interleaved visual sequence $\mathbf{s}_t = [\tau_1, \mathcal{V}(f_1);\ \tau_2, \mathcal{V}(f_2);\ \ldots;\ \tau_N, \mathcal{V}(f_N)]$. This interleaving enables the LLM to directly associate visual content with its temporal position and to retrieve relevant timestamps for grounding.

Input Sequence. To address the temporal isolation of independent chunks, we introduce a *read-write memory loop* (Fig. 1). Specifically, the complete input sequence $\mathbf{x}_t$ to $\mathcal{G}_\theta$ for chunk t is formulated as:

$$\mathbf{x}_t = [\,\mathbf{m}_{t-1};\ \mathbf{s}_t;\ \mathbf{d};\ \mathbf{q}\,], \tag{1}$$

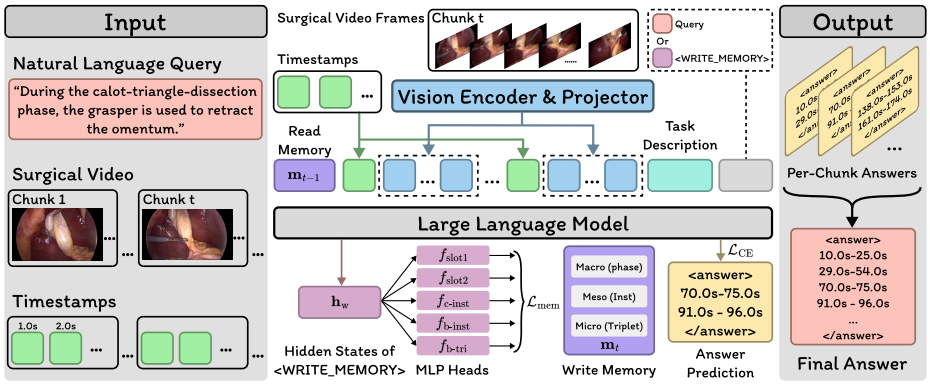

Fig. 1. Overview of our Memory-Augmented VLM Framework. The *read-write memory loop* transfers contextual state across video chunks. **Read:** The inherited memory prompt $\mathbf{m}_{t-1}$ is prepended to the input sequence. **Write:** A `<WRITE_MEMORY>` token aggregates a representation $\mathbf{h}_{\mathrm{w}}$, decoded by auxiliary heads into an updated three-tier memory state $\mathbf{m}_t$ for the next chunk.

where $\mathbf{m}_{t-1}$ is the textual memory prompt summarising the surgical state up to chunk $t-1$ (i.e., **Memory Read (Injection)** in Sec. 2.3), $\mathbf{d}$ is a task description specifying the grounding task and expected output format, and $\mathbf{q}$ is a natural-language query. All textual components ($\mathbf{m}_{t-1}$, $\mathbf{d}$, $\mathbf{q}$) are tokenized into text token sequences by the LLM tokenizer. Given $\mathbf{x}_t$, the model dynamically generates answer tokens $\hat{\mathbf{y}}$ in the format `<answer>`$\hat{s}_1$–$\hat{e}_1$ $\hat{s}_2$–$\hat{e}_2$ $\cdots$`</answer>`, which are decoded into the predicted temporal segments $\hat{\mathcal{Y}} = \{(\hat{s}_k, \hat{e}_k)\}_k$.

2.3 Hierarchical Read-Write Memory Loop

The core of our pipeline is maintaining cross-chunk procedural context. To achieve this, we define a structured memory state and establish mechanisms for reading inherited context and writing updated states for subsequent chunks.
Structured Memory Design. The three grounding levels naturally prescribe what inter-chunk memory should encode: the current phase determines which instruments are expected, and the active instruments constrain which triplets can occur. Following this hierarchy, we define a three-tier memory state $\mathbf{m}_t$ carried from chunk t to chunk $t+1$: **(i)** *Macro* (phase): the cumulative phase transition path and the current phase identity, encoding global procedural progress; **(ii)** *Meso* (instrument): the set of instruments active during and at the boundary of the current chunk, conveying instrumentation continuity; **(iii)** *Micro* (triplet): the action triplets ⟨I, V, T⟩ active at the chunk boundary, capturing fine-grained action state. Each tier provides contextual priors for the corresponding grounding level and for all subsequent finer levels.
Memory Read (Injection). The inherited memory state $\mathbf{m}_{t-1}$ is serialised into natural language, delimited by `<MEMORY_START>` and `<MEMORY_END>` tokens, and prepended to the input sequence as shown in Eq. 1.

Memory Write (Generation). During inference, annotations are unavailable, while the model needs to seamlessly update the memory. To realize this, a learnable token `<WRITE_MEMORY>` is appended after task description $\mathbf{d}$. `<WRITE_MEMORY>` is constrained to attend only to the context preceding the query, generating an effective input:

$$\mathbf{x}_t^{\mathrm{w}} = [\,\mathbf{m}_{t-1};\ \mathbf{s}_t;\ \mathbf{d};\ \texttt{<WRITE\_MEMORY>}\,]. \tag{2}$$

Accordingly, the last-layer hidden state corresponding to the `<WRITE_MEMORY>` token, $\mathbf{h}_{\mathrm{w}} \in \mathbb{R}^d$, aggregates information from the memory, visual content, and task description, while remaining unaffected by the query or answer tokens. We therefore apply $\mathbf{h}_{\mathrm{w}}$ as a query-agnostic summary of the chunk. Five lightweight MLP heads are attached to $\mathbf{h}_{\mathrm{w}}$ to predict the specific components of $\mathbf{m}_t$. To handle phase transitions, we allocate two categorical phase slots to predict phases temporally: **(i)** f_{slot1}: the identity of the first phase; **(ii)** f_{slot2}: the identity of the second phase, or a special `NONE` class if only one phase occurs. These two slots naturally encode phase occurrences within a chunk. We omit additional slots, as chunks containing more than two phases are extremely rare (7 out of 1602 chunks, $\sim$0.44%) and are subsequently treated as minor perturbations. **(iii)** $f_{\text{c-inst}}$: multi-label instrument presence within the chunk; **(iv)** $f_{\text{b-inst}}$: multi-label instrument presence at the chunk boundary; **(v)** $f_{\text{b-tri}}$: multi-label action triplet presence at the chunk boundary. The decoded predictions are assembled into the three-tier format and serialised as $\mathbf{m}_t$ for the next chunk, completing the autoregressive memory rollout without ground-truth labels.

2.4 Training

Training Objectives. The primary *grounding loss* is the standard auto-regressive cross-entropy over the target answer tokens:

$$\mathcal{L}_{\mathrm{CE}} = -\sum_{i=1}^{N_y} \log P\big(y_i \mid \mathbf{x}_t,\, y_{<i};\, \theta\big), \tag{3}$$

where N_y is the number of target tokens in the answer and the probability is conditioned on the input sequence $\mathbf{x}_t$ and the preceding target tokens. The auxiliary *memory loss* supervises the five prediction heads attached to $\mathbf{h}_{\mathrm{w}}$:

$$\mathcal{L}_{\mathrm{mem}} = \sum_{k=1}^{5} \lambda_k\, \mathcal{L}_k, \tag{4}$$

where $\mathcal{L}_k$ consists of the cross-entropy loss for f_{slot1} and f_{slot2}, the binary cross-entropy losses for $f_{\text{c-inst}}$, $f_{\text{b-inst}}$, and $f_{\text{b-tri}}$; λ_k are per-head balancing weights.

Three-stage Strategy. Jointly optimising both objectives may encounter mutual interference: the VLM has not yet learned to ground events, and unstable hidden states make auxiliary head training noisy. We therefore apply a progressive approach: **(i)** *Grounding with oracle memory.* Only $\mathcal{L}_{\mathrm{CE}}$ is active; memory prompts are constructed from ground-truth annotations. $\mathcal{G}_\theta$ is adapted via LoRA

with a frozen $\mathcal{V}$, establishing reliable grounding capability. **(ii)** *Memory head pre-training.* The entire VLM is frozen. Only the five auxiliary heads are trained on $\mathbf{h}_{\mathrm{w}}$ with $\mathcal{L}_{\mathrm{mem}}$, learning to get surgical state from stable representations. **(iii)** *Joint fine-tuning.* Both objectives are combined: $\mathcal{L} = \mathcal{L}_{\mathrm{CE}} + \lambda_{\mathrm{aux}}\mathcal{L}_{\mathrm{mem}}$, where λ_{aux} balances the auxiliary multi-task learning. LoRA modules θ and the memory heads are trained jointly, allowing the hidden representations to co-adapt for both grounding and memory generation.

3 Experiments

3.1 Benchmark Construction

We build our HMS-SurgVTG benchmark upon CholecT50 [15], a publicly available dataset of 50 laparoscopic cholecystectomy videos annotated at 1 fps with surgical phases, instruments, and action triplets. We follow the official split: 35 videos for training, 5 for validation, and 10 for testing. Since the original CholecT50 provides frame-level labels rather than temporal segments, we design a specific pipeline to convert these annotations into hierarchical multi-segment temporal grounding labels of our HMS-SurgVTG benchmark.

Specifically, our pipeline applies three main steps: **(i)** *Frame-to-segment conversion.* For each video, we aggregate frame-level labels into contiguous time segments at each level: phase, instrument (conditioned on phase), and action triplet (conditioned on phase and instrument), producing multi-segment ground truth for every query. **(ii)** *Short segment filtering.* To suppress annotation noise, we discard segments shorter than 2 s (i.e., ≤ 3 frames). While prior work [17] treats actions under 3 s as noise, we adopt this more conservative threshold to account for the lower temporal resolution (1 FPS) and the higher annotation reliability of the original CholecT50. We quantify the impact of this step at each hierarchy level: at the phase level, none of the 100,863 annotated frames are affected; at the instrument level, 1,295 out of 138,993 frames are removed (0.93%); at the triplet level, 3,148 out of 150,532 frames are removed (2.09%). **(iii)** *Hierarchical query generation.* Natural-language queries are generated from the dedicated templates by explicitly incorporating the hierarchical context. A phase-level query only asks for the phase itself; an instrument-level query explicitly references its parent phase to resolve ambiguities; while a triplet-level query conditions on both the parent phase and instrument to preserve coarse-to-fine dependencies. The resulting HMS-SurgVTG benchmark comprises 2,666 entries (339 phase, 750 instrument, 1,577 triplet) with hierarchical parent–child references across videos.

Our Memory-Augmented VLM framework is built upon Qwen3-VL-8B [19]. We freeze the vision encoder and projector, fine-tuning only the LLM backbone via LoRA. Videos are processed in 64-frame chunks (i.e., 64 seconds per chunk at 1 fps), following a common practice in general-domain VTG works [18]. Training proceeds in three stages using the AdamW [12] optimizer, cosine scheduling, and a batch size of 1. In the first stage, we fine-tune the LLM for temporal grounding generation for 2 epochs. In the second stage, we freeze the LLM and train five

lightweight memory heads for 2 epochs to establish memory writing ability. Each memory head is implemented as a two-layer MLP with GELU activation. These heads predict two phase slots via Cross-Entropy (CE) loss, alongside chunk-internal/boundary instruments and boundary triplets via Binary Cross-Entropy (BCE) loss, using empirical loss balancing weights of 2.0, 0.5, and 0.2, respectively. In the third stage, we jointly train the whole model end-to-end for 1 epoch, combining the generation CE loss and auxiliary memory losses with the empirical weight $\lambda_{\text{aux}} = 0.1$. All experiments are conducted on a single NVIDIA H200 GPU with DeepSpeed ZeRO-2 and Flash Attention.

Evaluation Metrics. Since HMS-SurgVTG requires a top-1 multi-segment temporal answer for each query, we evaluate with mean Intersection-over-Union (mIoU) and Recall@1 under IoU thresholds of 0.3 and 0.5, following common VTG practice.

3.2 Experimental Results

Table 1. Main results on the HMS-SurgVTG benchmark. We report mIoU, R1@0.3, and R1@0.5 across the Phase, Instrument, and Triplet levels. *Ours (Final)* denotes our complete framework optimized via the three-stage joint training. *Ours (w/ GT Memory)* indicates the theoretical upper bound (Oracle), evaluated by providing the model with ground-truth memory descriptions during inference to validate the necessity and efficacy of our memory-guided design.

Method	Existing Baselines		Ours		Upper Bound
	Qwen3	UniVTG	*w/o* Memory	**Final**	*w/* GT Memory
Phase					
mIoU	21.33	2.11	16.03	19.23	*67.67*
R1@0.3	32.00	1.00	20.00	25.00	*88.00*
R1@0.5	20.00	0.00	5.00	3.00	*74.00*
Instrument					
mIoU	1.95	0.49	43.14	52.97	*52.65*
R1@0.3	0.91	0.00	62.98	70.31	*70.26*
R1@0.5	0.00	0.00	45.11	59.38	*59.48*
Triplet					
mIoU	0.95	0.24	35.23	37.26	*40.34*
R1@0.3	1.02	0.00	44.65	47.98	*52.24*
R1@0.5	0.00	0.00	29.26	34.10	*35.90*
Overall					
mIoU	3.74	0.54	35.17	38.85	*46.19*
R1@0.3	4.83	0.12	46.49	50.56	*60.36*
R1@0.5	2.48	0.00	30.56	36.43	*45.61*

Baselines. To comprehensively evaluate our Memory-Augmented VLM framework, we benchmark against top-performing generic video foundation models and specialized temporal grounding paradigms. For all evaluated models, training is performed exclusively on the training set, while the reported metrics are computed across the combined validation and test sets. We compare our method against two main classes: (1) *Temporal Grounding Specialist.* We select Uni-VTG [11] evaluated in a zero-shot inference setting, which is a state-of-the-art approach for locating specific actions in untrimmed videos; (2) *Direct Baseline.* We evaluate Qwen3-VL-8B, our underlying visual-language backbone, after supervised fine-tuning on our dataset. This quantifies the backbone model's foundational ability to perform our task without the addition of our proposed hierarchical structural design and long-term memory tuning.

Main Results. Table 1 shows that our method consistently outperforms the remaining baselines on the *fine-grained* levels (Instrument and Triplet) across all metrics, demonstrating the effectiveness of memory-guided hierarchical reasoning for local, interaction-centric evidence. In contrast, Phase-level grounding is less improved and can be lower than Qwen3, suggesting that coarse workflow stages rely more on global temporal priors and long-range context than on local visual-memory cues emphasized by our design.

Impact of Memory-Guided Architecture. Removing the lightweight MLP memory heads ("Ours (*w/o* Mem)") leads to a substantial performance degradation across all semantic levels, particularly in boundary-heavy instrument localization. This confirms that explicitly maintaining and querying an internal temporal memory provides crucial historical context, effectively mitigating temporal forgetting and hallucination over lengthy surgical videos compared to relying solely on standard LLM self-attention.

Analysis of the Theoretical Bottleneck. To isolate our system's learning bottleneck, we evaluate an oracle setting ("Ours (*w/* GT Mem)") injected with actual ground-truth memory observations. With oracle prompts, phase mIoU increases to 67.67%, surpassing all baselines, while fine-grained instrument performance remains largely unchanged. This contrast yields a key insight: fine-grained localization in our framework primarily depends on immediate visual features that our model already reliably captures, whereas accurate macro-level phase grounding is fundamentally bottlenecked by the quality of historical context compression. Bridging this semantic gap between predicted and actual memory presents values for future improvements.

4 Conclusion and Future Work

In this work, we introduce a novel research problem of Surgical Video Temporal Grounding (SurgVTG) and establish HMS-SurgVTG, a new benchmark dedicated to localizing hierarchical procedural events in long-form surgical videos. To address long-range temporal dependencies and cross-chunk continuity issues, we propose a memory-augmented VLM framework explicitly designed to retain global procedural context, enhancing the effectiveness in temporal grounding of

surgical videos. For future work, first, we can develop adaptive reasoning mechanisms to dynamically adjust attention resolution across different semantic hierarchies, accelerating the learning process while preserving fine-grained localization precision. Second, SurgVTG can be systematically expanded by broadening its scope to include diverse surgical modalities, and by advancing its task complexity to support critical intraoperative sub-tasks, such as the temporal grounding of bleeding and adverse events.

Acknowledgments. The work described in this paper was supported by the Research Grants Council of the Hong Kong Special Administrative Region, China, under Project T45-401/22-N; and in part by The Chinese University of Hong Kong, under Projects 4055299 and 6907743.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Birkmeyer, J.D., Finks, J.F., O'reilly, A., Oerline, M., Carlin, A.M., Nunn, A.R., Dimick, J., Banerjee, M., Birkmeyer, N.J.: Surgical skill and complication rates after bariatric surgery. New England Journal of Medicine **369**(15), 1434–1442 (2013)
2. Chen, Y., He, S., Jin, Y., Qin, J.: Surgical activity triplet recognition via triplet disentanglement. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 451–461. Springer (2023)
3. Czempiel, T., Paschali, M., Keicher, M., Simson, W., Feussner, H., Kim, S.T., Navab, N.: Tecno: Surgical phase recognition with multi-stage temporal convolutional networks. In: International conference on medical image computing and computer-assisted intervention. pp. 343–352. Springer (2020)
4. Gao, X., Jin, Y., Long, Y., Dou, Q., Heng, P.A.: Trans-svnet: Accurate phase recognition from surgical videos via hybrid embedding aggregation transformer. In: International conference on medical image computing and computer-assisted intervention. pp. 593–603. Springer (2021)
5. Garrow, C.R., Kowalewski, K.F., Li, L., Wagner, M., Schmidt, M.W., Engelhardt, S., Hashimoto, D.A., Kenngott, H.G., Bodenstedt, S., Speidel, S., et al.: Machine learning for surgical phase recognition: a systematic review. Annals of surgery **273**(4), 684–693 (2021)
6. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al.: Lora: Low-rank adaptation of large language models. Iclr **1**(2), 3 (2022)
7. Jin, Y., Li, H., Dou, Q., Chen, H., Qin, J., Fu, C.W., Heng, P.A.: Multi-task recurrent convolutional network with correlation loss for surgical video analysis. Medical image analysis **59**, 101572 (2020)
8. Kim, J., Choi, D., Lee, N., Beane, M., Kim, J.: Surch: Enabling structural search and comparison for surgical videos. In: Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems. pp. 1–17 (2023)
9. Lei, J., Berg, T.L., Bansal, M.: Detecting moments and highlights in videos via natural language queries. Advances in Neural Information Processing Systems **34**, 11846–11858 (2021)

10. Li, Z., Di, S., Zhai, Z., Huang, W., Wang, Y., Xie, W.: Universal video temporal grounding with generative multi-modal large language models. In: The Thirty-ninth Annual Conference on Neural Information Processing Systems
11. Lin, K.Q., Zhang, P., Chen, J., Pramanick, S., Gao, D., Wang, A.J., Yan, R., Shou, M.Z.: Univtg: Towards unified video-language temporal grounding. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 2794–2804 (2023)
12. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: International Conference on Learning Representations (2019), https://openreview.net/forum?id=Bkg6RiCqY7
13. Loukas, C., Gazis, A., Kanakis, M.A.: Surgical performance analysis and classification based on video annotation of laparoscopic tasks. JSLS: Journal of the Society of Laparoscopic & Robotic Surgeons **24**(4), e2020–00057 (2020)
14. Maier-Hein, L., Vedula, S.S., Speidel, S., Navab, N., Kikinis, R., Park, A., Eisenmann, M., Feussner, H., Forestier, G., Giannarou, S., et al.: Surgical data science for next-generation interventions. Nature Biomedical Engineering **1**(9), 691–696 (2017)
15. Nwoye, C.I., Yu, T., Gonzalez, C., Seeliger, B., Mascagni, P., Mutter, D., Marescaux, J., Padoy, N.: Rendezvous: Attention mechanisms for the recognition of surgical action triplets in endoscopic videos. Medical Image Analysis **78**, 102433 (2022)
16. Ren, S., Yao, L., Li, S., Sun, X., Hou, L.: Timechat: A time-sensitive multimodal large language model for long video understanding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 14313–14323 (June 2024)
17. Ríos, M.S., Molina-Rodriguez, M.A., Londoño, D., Guillén, C.A., Sierra, S., Zapata, F., Giraldo, L.F.: Cholec80-cvs: An open dataset with an evaluation of strasberg's critical view of safety for ai. Scientific Data **10**(1), 194 (2023)
18. Shen, X., Zhang, W., Chen, J., Elhoseiny, M.: Vgent: Graph-based retrieval-reasoning-augmented generation for long video understanding. In: The Thirty-ninth Annual Conference on Neural Information Processing Systems (2025), https://openreview.net/forum?id=5xPvWat3IX
19. Team, Q.: Qwen3 technical report (2025), https://arxiv.org/abs/2505.09388
20. Xu, R., Xiao, G., Chen, Y., He, L., Peng, K., Lu, Y., Han, S.: StreamingVLM: Real-time understanding for infinite video streams. In: The Fourteenth International Conference on Learning Representations (2026), https://openreview.net/forum?id=gVbPWbA97s
21. Zeng, X., Li, K., Wang, C., Li, X., Jiang, T., Yan, Z., Li, S., Shi, Y., Yue, Z., Wang, Y., Wang, Y., Qiao, Y., Wang, L.: Timesuite: Improving MLLMs for long video understanding via grounded tuning. In: The Thirteenth International Conference on Learning Representations (2025), https://openreview.net/forum?id=nAVejJURqZ