Less is More: Label-Guided Efficient Summarization of Procedural Videos

Shreya Rajpal[1*] **Michal Golovanevsky**[2] **Carsten Eickhoff**[3†]
[1]Vellore Institute of Technology [2]Brown University [3]University of Tübingen
shreyarajpal6@gmail.com michal_golovanevsky@brown.edu
carsten.eickhoff@uni-tuebingen.de

Abstract

Video summarization transforms long videos into concise representations that are easier to document and analyze, especially in high-stakes domains such as surgical training. However, long-form videos often require dense frame processing or supervised training pipelines, which can be computationally expensive and may still miss important procedural content. We present PRISM: Procedural Representation via Integrated Semantic and Multimodal Analysis, a zero-shot and training-free framework for frame-efficient procedural video summarization. PRISM selects fewer than 5% of video frames while retaining over 84% semantic content and improving over baselines by up to 7.5%. Rather than exhaustively processing frames, PRISM uses lightweight visual filtering and dynamically generated procedural labels as semantic anchors to select meaningful frame-label groups. This selective inference design preserves key actions, transitions, and contextual details while reducing the number of visual inputs passed to downstream vision-language captioning stages. We evaluate PRISM on YouCook2 and ActivityNet Captions, with additional studies on keyframe selection benchmarks and surgical video datasets. Across procedural and domain-specific video tasks, PRISM achieves strong semantic alignment and accuracy, suggesting that efficient multimodal video understanding can be achieved by grounding generation in dynamically generated semantic anchors.

1 Introduction

Video summarization plays a significant role in making long-form video content easier to analyze, especially in domains such as robotics (Zhang et al., 2021), education (Gonzalez et al., 2023), and surgical training (Yang et al., 2024). However, long-form instructional and surgical videos often exceed 40 minutes (Wang et al., 2022; Lin et al., 2025), making it challenging to process a large number of frames. Therefore, effective summarization must reduce visual processing while preserving important actions, transitions, and procedural context.

A common approach to video summarization is keyframe selection, which extracts informative frames or segments that best represent the content of a video. In high-stakes domains such as surgical video summarization, summaries must do more than shorten the video; they must preserve the order of actions, procedural intent, and contextually important details. This creates a central challenge for long-form procedural summarization, as systems must be computationally efficient while still retaining the semantic and temporal structure needed for accurate summaries. [1]

Existing methods often emphasize either summary quality or efficiency alone. Lightweight approaches based on visual cues, motion, similarity, or clustering reduce redundancy but may miss fine-grained procedural transitions (Cho and Kang, 2019; Mendi et al., 2013; Sandhu and Agarwal, 2015; Gygli et al., 2014; Song et al., 2015). In contrast, deep sequence models, adversarial learning, attention mechanisms, pretrained embeddings, and vision-language models improve semantic relevance and temporal coherence, but often rely on supervised frame-level scorers, video-level captions, predefined queries, rule-based selection, or task-specific training (Rochan et al., 2018; Mahasseni et al., 2017; Argaw et al., 2024; Radford et al., 2021; Li et al., 2022; Narasimhan et al., 2021; Zhao et al., 2024; Tan et al., 2024; Huynh-Lam et al., 2024). This motivates a summarization framework that can reduce the number of visual inputs while preserving procedural meaning without requiring

[*]Work done during an internship at the University of Tübingen.

[†]Corresponding author.

[1]Code is available at https://github.com/Shreyarajpal12/PRISM-code.

supervised annotations or fixed queries.

In this work, we present PRISM: Procedural Representation via Integrated Semantic and Multimodal Analysis, a zero-shot and training-free framework for frame-efficient video summarization. PRISM is designed to maintain both efficiency and semantic grounding. Rather than relying on dense frame processing, PRISM first uses lightweight visual filtering to identify candidate frames with meaningful transitions. It then dynamically generates procedural labels from the video itself, validates them as semantic anchors, and aligns them with relevant frames using vision-language similarity. The selected frame-label groups are temporally aggregated and summarized into a coherent procedural narrative. This allows PRISM to reduce redundant visual inputs while preserving key actions, transitions, and contextual details.

The framework draws inspiration from the behavior of light through prisms. The left prism decomposes raw video into semantically meaningful elements using frame embeddings and adaptive sampling. The center filtering stage validates labels and selects matching frames, acting as a lens that sharpens and aligns relevant content. The right prism recombines the selected frame-label pairs into a focused semantic summary using language models. Like light passing through two prisms and a filter, PRISM decomposes procedural video content, filters it through semantic anchors, and reassembles it into a coherent, context-rich narrative.

We evaluate PRISM on YouCook2 (Zhou et al., 2018b) and ActivityNet Captions (Krishna et al., 2017) for video-level captioning and summarization. PRISM selects fewer than 5% of video frames while preserving over 80% of semantic content, measured using BERTScore. It also improves over existing baselines on video-level captioning tasks, measured using METEOR (Lavie and Agarwal, 2007). We further study PRISM in medical video summarization, where reducing visual processing while preserving procedural structure is especially important. We also do an additional keyframe selection study on TVSum (Song et al., 2015) and SumMe (Gygli et al., 2014) and report results in Appendix F.

Our contributions are:

- We introduce PRISM, a zero-shot and training-free framework for frame-efficient procedural video summarization that does not rely on supervised frame-level labels, fixed user queries, or dataset-provided annotations during inference.
- We propose a label-guided selective inference pipeline that combines lightweight visual filtering with dynamically generated procedural labels. These labels serve as semantic anchors for selecting meaningful frame-label groups, reducing dense visual processing while preserving procedural structure.
- We show that dynamically generated labels improve the grounding of vision-language summarization by guiding frame selection through vision encoder similarity, enabling summaries to be constructed from semantically selected evidence rather than densely processed frames.
- We evaluate PRISM across instructional, activity, keyframe selection, and surgical video settings. Results show that PRISM selects fewer than 5% of frames while maintaining strong semantic alignment and precision, demonstrating efficient multimodal video understanding without supervised training or task-specific fine-tuning.

2 Related Works

Video summarization transforms long-form videos into compact representations, including static keyframes (Junyent et al., 2015), dynamic skim videos (Alam et al., 2020), and textual summaries. Early methods estimated keyframe importance using visual or statistical cues such as color-histogram clustering (de Avila et al., 2008), motion and optical-flow activity (Mendi et al., 2013; Wolf, 1996), and entropy-based selection (Li et al., 2020). Later approaches formulated summarization as visual-importance prediction through LSTM-based frame scorers (Zhang et al., 2016), adversarial and reinforcement-learning frameworks (Mahasseni et al., 2017; Zhou et al., 2018a), and attention-based models such as VASNet and CSTA for temporal or spatio-temporal modeling (Fajtl et al., 2019; Son et al., 2024). These methods improve frame selection, but many treat summarization primarily as a visual importance scoring problem and depend on human annotations.

Recent methods incorporate language, audio, captions, and vision-language models to improve semantic relevance and narrative flow. Query-focused and personalized methods align summaries

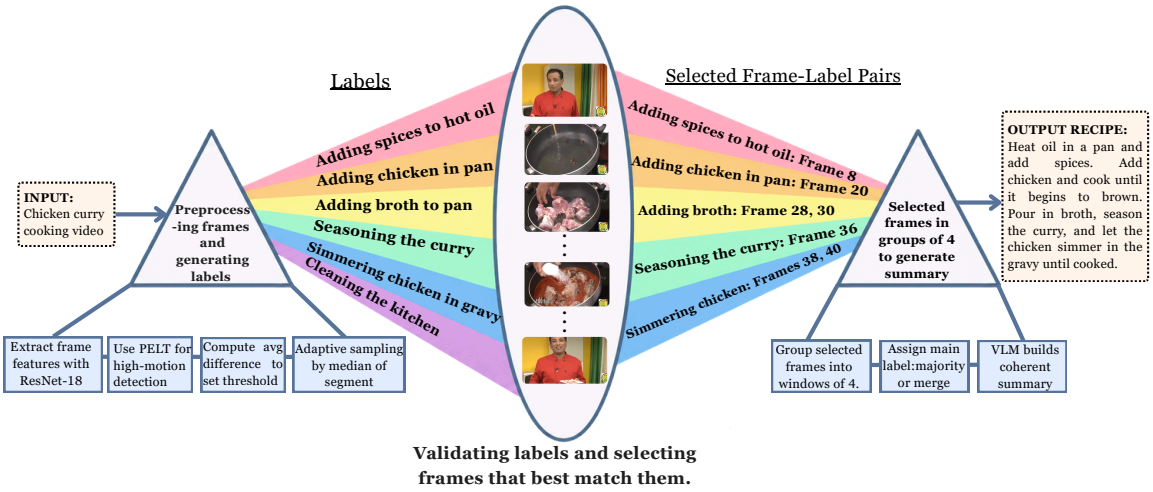

Figure 1: PRISM pipeline: the input video is semantically decomposed, filtered, and recombined into a coherent summary.

with user queries or preferences (Sharghi et al., 2017; Zhang et al., 2018; Alaa et al., 2024; Gunawardena et al., 2019). Caption-based methods use language as a semantic bridge, FrameRank ranks captioned frames (Lei et al., 2019), CLIP-It aligns frames to video-level dense captions using CLIP (Radford et al., 2021; Narasimhan et al., 2021), Cap2Sum uses dense captions as supervision (Zhao et al., 2024), and LMSKE applies shot segmentation and CLIP-feature clustering for sequential keyframe extraction (Tan et al., 2024). LLMVS uniformly samples frames, captions them with a multimodal LLM, and trains an LLM-based visual importance scorer using human annotations (Lee et al., 2025), while ZeroTA performs dense video captioning in a zero-shot setting by optimizing text and moment representations at inference time (Jo et al., 2023). Although these methods improve semantic grounding, many rely on external queries, global or dense captions, user preferences, human annotations, or supervised training. In contrast, PRISM is training-free, reduces the candidate frame set before VLM captioning, and uses dynamically generated procedural labels as semantic anchors rather than ranking uniformly sampled frames.

Beyond summarization accuracy, efficiency and computational sustainability remain central challenges in long-form video understanding. Recent efficient video-language methods reduce visual processing through query-conditioned frame selection, token compression, or saliency-aware retrieval. Multimodal large language model-based frame selection and Frame-Voyager select frames conditioned on a question or task query (Yu et al., 2025); PruneVid and DyCoke reduce redundant visual tokens or KV-cache computation inside video-language models (Huang et al., 2024; Tao et al., 2025); and Sali4Vid improves dense video captioning through saliency-aware frame reweighting derived from timestamp annotations and semantic-segment-based caption retrieval(Jeon et al., 2025). These approaches demonstrate the value of efficient visual processing, but they either require an external task signal, reduce computation after visual inputs have entered the model, or rely on annotation-derived supervision.

PRISM combines frame-level efficiency with bottom-up semantic grounding. It requires no external query, user profile, global caption, dense caption supervision, human annotations, event timestamp, or dataset-provided annotation. Instead, PRISM generates procedural labels from sampled frames, validates them against the video's own context, and uses them as semantic anchors for selecting frame-label pairs through vision-language similarity. Because this selection occurs before downstream summarization, PRISM reduces the number of visual inputs passed to downstream VLM stages to fewer than 5% of the original frames.

3 Method

Our proposed framework, PRISM, is a multi-stage pipeline that enables efficient and context-aware keyframe selection, followed by semantically grounded video summarization. This pipeline is designed to ensure that the summary covers key visuals, preserves contextual meaning and procedu-

ral order, and remains efficient across domains.

3.1 Dataset and Experimentation

Our pipeline targets video summarization, with a focus on video-level dense captioning. We use YouCook2 (Zhou et al., 2018b), a dataset of over 2,000 untrimmed instructional cooking videos spanning 89 recipes, each annotated with temporal segments and stepwise imperative captions. We also use the ActivityNet Captions (Krishna et al., 2017) dataset, which contains temporally localized, natural language descriptions for untrimmed videos, making it a benchmark for dense video captioning and summarization tasks. We evaluate summarization performance using BLEU, ROUGE-L, METEOR, and BERTScore (Papineni et al., 2002; Lin, 2004; Zhang et al., 2020; Lavie and Agarwal, 2007) with a focus on capturing semantic alignment, procedural coherence, and linguistic quality. Lastly, we briefly discuss the system's application in the medical domain.

3.2 Stage 1: Frame Embedding & Adaptive Sampling

We first convert the raw video into a manageable set of candidate frames. Frames are sampled at a fixed rate (e.g., 1 fps) and passed through ResNet-18 to extract 512-dimensional embeddings $f_i \in R^{512}$ that capture high-level spatial features (He et al., 2015). Next, we detect statistically significant visual transitions via the Pruned Exact Linear Time (PELT) algorithm (Killick et al., 2012). PELT identifies a set of change indices $\{t_k\}$ such that embedding differences before and after each t_k exceed a model-cost penalty, yielding an initial subset $F_{PELT} \subset \{f_i\}$.

To further adapt sampling density to visual variability, we partition the video into segments between change points and compute, for each segment s (here, we set s=10), the median Euclidean distance med_s among its frame embeddings, and δ is the median feature-difference threshold. Segments with $med_s < \delta$ retain one representative frame; those with $med_s \geq \delta$ retain two. In our video experiments, we set $\delta = 0.30$. The resulting visually filtered and adaptively sampled frames form Selected Frames Set 1, which is passed to the semantic anchoring stage as shown in Figure 2. The nuances of different hyperparameter settings are discussed in the Appendix.

3.3 Stage 2: Label Generation and Semantic Anchoring

Following adaptive frame sampling, the induced frame set $F = \{f_i\}_{i=1}^{n}$ is processed to generate semantic anchors that represent high-level procedural steps. This stage involves three subprocesses: caption generation, semantic label validation, and label to frame association.

We begin by feeding each sampled frame f_i into a vision-language model. Depending on the domain and setup, models such as Gemma3 (Team et al., 2025), LLaMA 3.2 Vision (Grattafiori et al., 2024), MiniCPM (Yao et al., 2024), or GPT4 (OpenAI et al., 2024) are employed to generate per-frame captions that describe the localized content. These captions are generalized into anchor labels. For example, the caption *"The person in a white apron and clear gloves is carefully sprinkling a large amount of shredded cabbage into a large stainless steel bowl..."* is mapped to the label *"Sprinkling shredded cabbage for kimchi"*, which may span multiple frames.

To ensure label quality, we introduce a label validation step using an LLM (in this case GPT-4). The model filters out labels that are overly vague (e.g., "Cooking instructions being given") or contextually insignificant (e.g., "cleaning utensils" is insignificant to the recipe) and retains only those labels that represent discrete procedural events. Stage 2 and Stage 3 can reduce noisy or hallucinated labels by checking whether they are coherent with the broader procedural context and visually aligned with selected frames. For example, in a recipe about ground beef, a hallucinated label referring to chicken may be removed if it is inconsistent with the remaining generated labels or weakly aligned with the visual evidence. Such frames are discarded whenever possible, as they do not align with the procedural context. This step also eliminates visually insignificant frames that may be missed by ResNet-based analysis, such as black screens, frames dominated by light flicker or glare, and transition frames without identifiable objects, which do not produce meaningful captions. As illustrated in Figure 2, this process filters the initial selected frames and labels into high-confidence frame-label pairs, which form Selected Frames Set 2 for downstream summarization.

The Venn diagram in Figure 3 shows frame-label pairs at a similarity threshold of 0.9, highlighting only high-confidence matches (e.g., "Deep frying

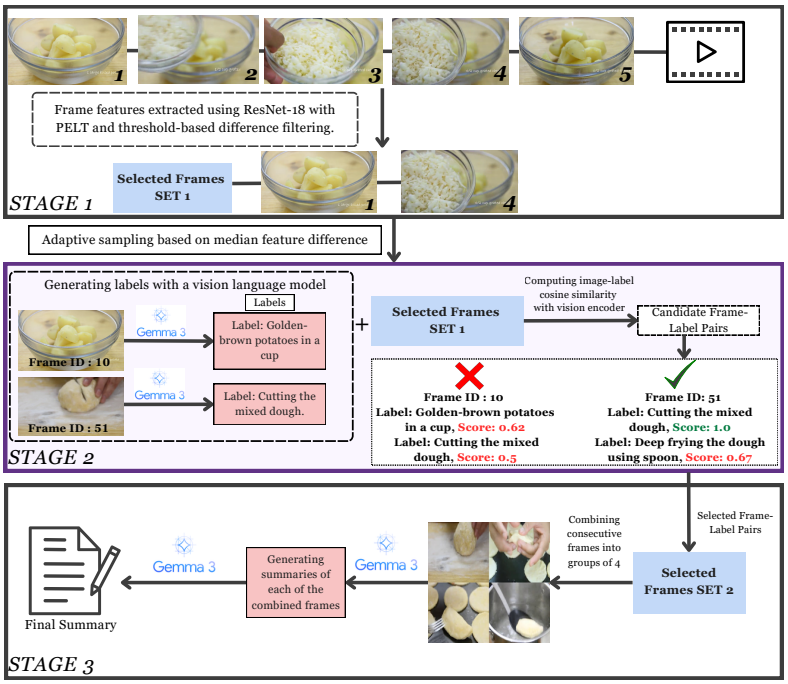

Figure 2: A complete example of the workflow.

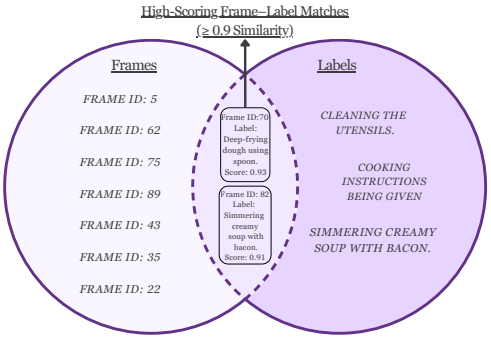

Figure 3: A Venn diagram illustrating overlap between frame embeddings and semantic labels where cosine similarity exceeds 0.9. Highlighted labels reflect procedural relevance across frames.

dough" at 0.93) for summarization. Frames and labels outside this intersection are excluded from the summary due to weak or irrelevant alignment. Next, we perform label to frame association using vision encoders such as CLIP by (Radford et al., 2021), BLIP by (Li et al., 2022), or BioMedCLIP by (Zhang et al., 2025). Each label ℓ_j and frame f_i is embedded into a shared space, and cosine similarity is computed:

$$S(f_j, \ell_i) = \frac{\langle E(f_j), E(\ell_i) \rangle}{\|E(f_j)\| \cdot \|E(\ell_i)\|}$$

- $\mathcal{F} = \{f_1, f_2, \ldots, f_N\}$: sampled frames
- $\mathcal{L} = \{\ell_1, \ell_2, \ldots, \ell_M\}$: validated semantic labels
- $sim(f_i, \ell_j)$: cosine similarity between f_i and ℓ_j
- τ: similarity threshold (set to 0.9)

We assign labels to frames using:

$$A(f_i) = \{\, \ell_i^* \,, if \max_{\ell \in \mathcal{L}} sim(f_i, \ell) \geq \tau, \emptyset, otherwise, \quad (1)$$

Here, ℓ_j^* denotes the best-matching semantic label. Frames with no label exceeding the threshold τ are discarded. This ensures semantic precision and relevance.

Notably, a frame may match multiple labels (e.g., 0.98 for ℓ_a and 0.95 for ℓ_b), reflecting overlapping steps in procedures. These frames are retained and prioritized during summarization.

3.4 Stage 3: Temporal Aggregation and Summary Construction

We now organize the selected frames into temporally ordered groups of four, forming windows:

$$W_i = \{f_{i^-1}^*, f_i^*, f_{i+1}^*, f_{i+2}^*\}$$

Here, f_i^* denotes the filtered frame obtained after Stage 2. Each window is passed into a vision-language summarizer (e.g., GPT4V, Gemini Vision, MiniCPM-V) to generate localized summaries.

Each 4-frame group is assigned a main label based on frame-wise captions. If all labels differ, the main label combines all four. If two labels occur twice (2:2), both are used. Otherwise, the majority label is assigned (e.g., 3:1).

These summaries are then stitched together using a large language model (e.g., GPT-4, LLaMA 3.1) following a layered, tree-like approach. Summaries are grouped based on the model's maximum context size and processed at the leaf level. The resulting outputs are recursively merged at higher levels, enabling the generation of a final, cohesive summary that integrates information from all segments in a structured manner. This step isolates frames and reduces hallucinations or inconsistencies by anchoring each part within the broader narrative.

Figure 2 represents the overview of our three-stage video summarization framework. Stage 1 extracts frame embeddings using ResNet-18 by (He et al., 2015; S Harakannanavar et al., 2022), with adaptive sampling based on thresholded feature difference and PELT segmentation. Stage 2 generates semantic labels using a vision-language model (e.g., Gemma3), followed by label-to-frame similarity scoring using a vision encoder. Only frames with cosine similarity exceeding a threshold (e.g., 0.9) are retained.

Stage 3 combines temporally adjacent filtered frames into groups of four to retain local visual context and generate a summary caption for each group. This helps in progressively constructing the final output narrative. Since the frames have already been filtered and matched with labels, each group contains representative moments rather than consecutive frames. We use four frames as a practical window size, since smaller groups may split a single procedural action, while larger groups may mix different steps and increase inference cost. This multi-stage process yields a coherent, semantically grounded video summary suitable for high-stakes domains like surgery, technical instruction, or training.

4 Results

We evaluate our method, PRISM, across dense video captioning tasks, benchmarking it against prior video summarization models using standard datasets and evaluation metrics. The results are reported using the MiniCPM-V model. Scores are scaled to 0–100 for all metrics.

4.1 Video Level Summarization

We first evaluate summarization quality on the YouCook2 dataset and the ActivityNet Captions dataset. Table 1 compares PRISM against multimodal and vision-only baselines using ROUGE-L, METEOR, and BERTScore. While existing works like UniVL and COOT (Ko et al., 2023) (Ging et al., 2020) focus on leveraging paired video-text pretraining, PRISM achieves strong semantic alignment, particularly outperforming all models on METEOR Score. These results highlight the strength of our summarization approach even with significantly fewer processed frames.

The evaluation uses a mix of traditional and semantic-based metrics. While BLEU and ROUGE are commonly used in video summarization, they fall short for our approach, which prioritizes generating long, detailed summaries over short, high-level captions as given in the dataset. For instance, the sentence "A young woman is seen standing in a room and leads into her dancing" is the ground-truth caption for a video segment in ActivityNet Captions. In contrast, our generated summary captures more detailed dance movements: "The video begins with a ballet dancer's movement in the en croisé position, emphasizing the precision, balance, and control required to execute the transition smoothly." BLEU and ROUGE mainly reward surface overlap through n-grams and longest common subsequences, so they can underrepresent semantically accurate but more detailed summaries. We therefore emphasize METEOR as the primary metric for comparison. V and V+T in Table 1 refer to PRISM's vision-only and vision+text modalities, respectively, where text denotes the audio transcript. PRISM improves over the YouCook2 baseline by 7.5% in METEOR, as shown in Table 1. Table 2 compares state-of-the-art models on the ActivityNet Captions validation set. PRISM achieves the highest METEOR score of 20.04, surpassing the strongest baseline by 3.68% relatively.

4.2 Key frame selection

To capture deeper semantic alignment, we also adopt LLM-based evaluation in ablation studies and case studies, where a language model serves as a judge to assess summary quality based on factual accuracy, detail, specificity, completeness and repetition. Each of these aspects is scored on a scale from 1 to 5. We then compute a weighted average, applying double weight to factual accuracy, and

Model	Modality	ROUGE-L	METEOR	BERT
ZeroTA (Jo et al., 2023)	V	–	2.10	–
Streaming GIT (Zhou et al., 2024)	V	–	3.60	–
PDVC (CLIP) (Wang et al., 2021; Yang et al., 2023)	V	–	5.70	–
CM^2 (Kim et al., 2024a)	V	–	6.08	–
E^2DVC (Wu et al., 2025)	V	–	6.11	–
CACMI (Jia et al., 2025)	V	–	6.21	–
PR-DETR (Li et al., 2025)	V+T	–	6.48	–
Streaming Vid2Seq (Zhou et al., 2024)	V	–	7.10	–
Vid2Seq (Yang et al., 2023)	V	–	9.30	–
VideoBERT (Sun et al., 2019b)	V	27.14	10.81	–
EMT (Zhou et al., 2018c)	V	27.44	11.55	–
$HiCM^2$ (Kim et al., 2024b)	V	–	12.80	–
CBT (Sun et al., 2019a)	V	30.44	12.97	–
ActBERT (Zhu and Yang, 2020)	V	30.56	13.30	–
Sali4Vid (Jeon et al., 2025)	V	–	13.54	–
MCCL (Xie et al., 2024)	V	–	14.69	–
UniVL (Luo et al., 2020)	V	40.09	17.57	–
AT+Video (Hessel et al., 2019)	V+T	36.65	17.77	–
DPC (Shi et al., 2019)	V+T	–	18.08	–
UniVL + MELTR (Ko et al., 2023)	V	41.28	18.19	–
COOT (Ging et al., 2020)	V	37.94	19.85	–
UniVL (Luo et al., 2020)	V+T	46.52	22.35	–
UniVL + MELTR (Ko et al., 2023)	V+T	47.04	22.56	–
PRISM (Ours)	V	16.49	**23.66**	**82.78**
PRISM (Ours)	V+T	23.58	**30.08**	**84.34**

Table 1: Performance comparison on YouCook2 using ROUGE-L, METEOR, and BERTScore.

Model	BLEU-4	METEOR
ZeroTA (Jo et al., 2023)	–	2.7
Vid2Seq (Yang et al., 2023)	–	8.50
CM^2 (Kim et al., 2024a)	2.38	8.55
E^2DVC (Wu et al., 2025)	2.43	8.57
PR-DETR (Li et al., 2025)	2.58	8.72
TSP (Alwassel et al., 2021)	2.02	8.75
Streaming GIT (Zhou et al., 2024)	–	9.00
Streaming Vid2Seq (Zhou et al., 2024)	–	10.00
GVL (Wang et al., 2023)	1.11	10.03
BMT (Iashin and Rahtu, 2020)	1.99	10.90
PDVC (TSN) (Wang et al., 2021)	3.07	11.27
iPerceive (Chadha et al., 2020)	2.98	12.27
MCCL (Xie et al., 2024)	3.89	12.52
ADV-INF + Global (Kanani et al., 2021)	9.45	16.36
PRISM (Ours)	1.02	**20.04**

Table 2: Performance comparison on ActivityNet Captions using BLEU-4 and METEOR.

normalize the result to a score between 0 and 1. This forms our normalized LLM Judge Score.

4.3 Case study: Medical Datasets Video Summarization

We explore the applicability of our summarization pipeline in medical settings through a case study on two surgical datasets: Cholec80 and PIT-VIS. Our pipeline uses MiniCPM to generate summaries, which are then evaluated using BERTScore, SemnCG, and LLM as a judge for phase level coverage (Akter and Karmaker, 2024).

To assess procedural completeness, we extracted 7 standard surgical phases in Cholec80 and 12 annotated phases in PIT-VIS (excluding pre/post phases with insufficient volume) and checked their presence within the generated summaries using a large language model. On average, MiniCPM captured 5.6/7 phases in Cholec80 and 9.24/12 in PIT-VIS, indicating strong phase-level alignment without direct supervision.

Dataset	Model	BERT	Sem-nCG	Phase Cov.
Cholec80	LLaMA 3.2-Vision	87.04	76.67	79.01
	Gemma 3	**87.33**	77.08	77.91
	MiniCPM	86.93	**81.11**	**80.83**
PIT-VIS	LLaMA 3.2-Vision	**85.05**	**100.00**	74.36
	Gemma 3	84.68	**100.00**	79.13
	MiniCPM	84.75	97.22	**81.95**

Table 3: Medical case study results on Cholec80 and PIT-VIS. MiniCPM achieves strong semantic and procedural summarization scores and is used as part of the pipeline. Phase coverage is computed using LLM-based matching.

Data	Setting	BLEU	R-L	BERT	MET	Judge	Frm	Time	Tok
YC	Video-Only	1.82	18.40	82.41	19.83	38.64	22.25	71.78	3,560
	No Stage 2	3.00	28.11	84.12	31.83	79.26	60.07	73.09	9,613
	No Stage 1	2.80	29.87	84.59	32.66	78.49	22.23	236.46	3,557
	No Proc.	2.70	28.03	84.13	31.83	80.82	322.91	1625.40	206,662
AN	Video-Only	0.97	9.04	80.98	17.29	37.54	18.39	40.37	2,942
	No Stage 2	0.99	8.77	81.19	17.59	43.33	20.04	28.52	3,206
	No Stage 1	0.97	8.68	81.29	17.47	43.57	17.39	88.69	2,782
	No Proc.	0.91	7.87	80.56	16.09	43.75	111.88	130.06	71,603

Table 4: Ablation results on YouCook2 (YC) and ActivityNet Captions (AN) across pipeline settings. **BLEU**, **R-L** (ROUGE-L), **BERT** (BERTScore), **MET** (METEOR), and **Judge** (LLM-Judge) are quality metrics scaled to 0–100. **Frm**: selected frames per video after Stage 2 filtering. **Time**: average end-to-end inference time per video in seconds. **Tok**: estimated visual tokens reaching the final summarization VLM, computed as (Frm / 4 grouping) using MiniCPM-V's average token cost across videos. Stage 2 incurs additional VLM captioning cost on the candidate frame set, not reported here.

This experiment was conducted to investigate the feasibility of training-free textual summarization pipelines that could combine multiple surgical subtasks e.g., frame-level labeling, phase annotation, and summary generation, into a unified algorithm. While we do not position these results as clinically validated, they point to a promising direction for automated surgical video understanding. We invite future research and clinical collaborations to rigorously evaluate and refine such pipelines for real-world deployment.

5 Ablation Studies

We ablate PRISM on 25% of YouCook2 and ActivityNet Captions, focusing on summary generation. To clarify the role of each component, we explicitly define the ablation settings. *Video-Only* uses PRISM without transcript fusion and relies only on selected visual frames. *No Stage 2* removes label generation and label-to-frame semantic anchoring, so summaries are generated from visually filtered frames selected by Stage 1. *No Stage 1* removes ResNet- and PELT-based visual filtering and instead relies on semantic label-guided frame selection from all sampled frames (at 1 fps). *No Processing* skips the frame-reduction pipeline and passes all frames directly to the summarization stage. Together, these settings isolate the effects of visual filtering, semantic anchoring, transcript fusion, and dense-frame processing on both summary quality and computational cost. All ablations use a Stage-2 label-confidence threshold of 0.9, a Stage-1 ResNet frame-difference threshold of 30, and an adaptive sampling batch size of 10. Results are reported in Table 4, with all quality metrics scaled to 0–100. The time column reports the average end-to-end inference time per video, while the Tok column estimates the number of visual tokens passed to the final-stage summarization VLM. Full settings appear in Appendix A, and hyperparameter analyses appear in Appendix F.

Accuracy. The full PRISM pipeline (No Stage 2 setting, with final-stage grouping) matches or exceeds No Processing across all quality metrics on both datasets. On YouCook2, No Stage 2 achieves METEOR 31.83 and BERTScore 84.12, comparable to No Processing's 31.83 and 84.13 despite using a fraction of the frames. On ActivityNet, No Stage 2 achieves METEOR 17.59 and BERTScore 81.19, exceeding No Processing's 16.09 and 80.56. We further find that Stage 1 alone is insufficient: since it relies only on motion cues, No Stage 1 outperforms No Stage 2 on ROUGE-L, BERTScore, and METEOR for YouCook2, indicating that semantic label-guided selection contributes more to summary quality than motion-based filtering alone. The Video-Only modality outperforms many supervised baselines in Table 1 while using the fewest visual tokens, showing that label-guided selection retains semantic quality even under aggressive frame reduction. Its LLM-Judge score is lower on YouCook2 because the evaluator favors factual accuracy and complete step reconstruction, and many cooking actions and ingredient details are conveyed primarily in speech rather than visuals; adding transcripts (V+T) recovers this gap.

Efficiency. On YouCook2, the full pipeline reduces end-to-end inference time per video by 22.2× compared to No Processing (1,625.40s to 73.09s) and selected frames by 81% (322.91 to 60.07). Because the final stage groups frames into windows of four before summarization, visual tokens reaching the final-stage VLM drop from approximately 207,000 to 9,600 per video, which is a 95.4% reduction. Removing Stage 1 retains quality but inflates inference time substantially (236.46s vs. 73.09s), confirming that adaptive sampling drives the efficiency while semantic label validation in Stage 2 provides the summary quality grounding.

On ActivityNet Captions, the full pipeline reduces inference time by 4.6× (130.06s to 28.52s) and final-stage visual tokens from ∼72,000 to ∼3,200 per video, a 95.5% reduction. ActivityNet contains short, motion-dense clips where mo-

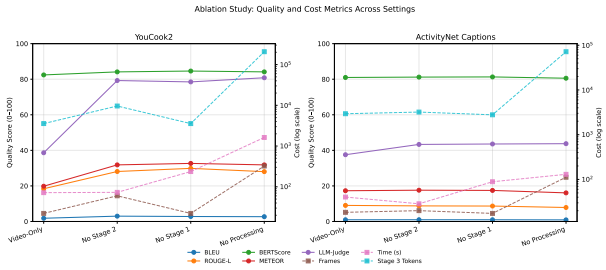

Figure 4: Ablation results on YouCook2 (left) and ActivityNet (right). Solid lines: quality metrics on the left y-axis (0–100). Dashed lines: cost metrics (frames, time, final-stage visual tokens) on the right y-axis (log scale). Quality stays stable across settings while cost rises by orders of magnitude when frame selection is removed.

tion patterns allow motion-based filtering alone to capture most meaningful events; the Video-Only modality also performs strongly here, using 95.9% fewer frames and 69% less inference time, since visual information is informative when transcripts are sparse. This demonstrates that PRISM adapts to different video characteristics: on motion-driven datasets, motion-based filtering captures most events, while on procedurally complex datasets like YouCook2, the label-guided pathway and transcript fusion become essential.

Figure 4 visualizes this quality–cost trade-off across both datasets. Quality remains stable across the three label-guided settings; on YouCook2, METEOR varies by less than 1% (31.83 to 32.66) and BERTScore by 0.5% across No Stage 2, No Stage 1, and No Processing, despite the two-order-of-magnitude difference in cost. Cost rises sharply when frame selection is removed, with final-stage visual tokens climbing from $\sim 10^3$ to over $\sim 10^5$ on YouCook2 under No Processing. The contrast between summary quality and exponentially rising cost illustrates PRISM's central trade-off: good frame reduction preserves summary quality, while the absence of frame reduction inflates compute cost. Final-stage grouping into windows of four is applied to all settings except No Processing.

6 Conclusion

We introduced PRISM, a training-free framework for frame-efficient procedural video summarization. PRISM combines lightweight visual filtering with dynamically generated procedural labels that act as semantic anchors for selecting informative frame-label pairs. Across our experiments, PRISM reduces visual inputs to fewer than 5% of the original frames and lowers inference time by over 95%, while preserving strong semantic content with BERTScore above 80% and improving METEOR by up to 7.5% over YouCook2 baselines. By moving frame reduction upstream of expensive VLM and language-model summarization stages, PRISM shows that efficient procedural video summarization can be achieved without supervised training, dataset annotations, or internal token compression. Future work will explore human evaluation of generated summaries and extensions to surgical report generation.

7 Limitations

PRISM uses vision-language captioning and vision-encoder filtering, but consistency errors can still occur. In visually ambiguous settings, models may generate plausible but incorrect labels, such as mistaking ground beef for chicken, which can lead to a coherent but factually wrong summary. Dual-stage filtering reduces vague, redundant, and hallucinated outputs, but failures may remain in fine-grained or high-stakes domains. Domain-specific or fine-tuned VLMs could help mitigate these errors.

References

Mousumi Akter and Santu Karmaker. 2024. Redundancy aware multi-reference based gainwise evaluation of extractive summarization. *Preprint*, arXiv:2308.02270.

Toqa Alaa, Ahmad Mongy, Assem Bakr, Mariam Diab, and Walid Gomaa. 2024. Video summarization techniques: A comprehensive review. *Preprint*, arXiv:2410.04449.

Imran Alam, Devesh Jalan, Priti Shaw, and Partha Pratim Mohanta. 2020. Motion based video skimming. In *2020 IEEE Calcutta Conference (CALCON)*, pages 407–411.

Humam Alwassel, Silvio Giancola, and Bernard Ghanem. 2021. Tsp: Temporally-sensitive pretraining of video encoders for localization tasks. *Preprint*, arXiv:2011.11479.

Dawit Mureja Argaw, Seunghyun Yoon, Fabian Caba Heilbron, Hanieh Deilamsalehy, Trung Bui, Zhaowen Wang, Franck Dernoncourt, and Joon Son Chung. 2024. Scaling up video summarization pretraining with large language models. *Preprint*, arXiv:2404.03398.

Aman Chadha, Gurneet Arora, and Navpreet Kaloty. 2020. iperceive: Applying common-sense reasoning to multi-modal dense video captioning and video question answering. *Preprint*, arXiv:2011.07735.

Sung Cho and Suk-Ju Kang. 2019. Histogram shape-based scene-change detection algorithm. *IEEE Access*, PP:1–1.

Sandra E. F. de Avila, Antonio Jr., Arnaldo de A. Araújo, and Matthieu Cord. 2008. Vsumm an approach for automatic video summarization and quantitative evaluation. In *2008 XXI Brazilian Symposium on Computer Graphics and Image Processing*, pages 103–110.

Jiri Fajtl, Hajar Sadeghi Sokeh, Vasileios Argyriou, Dorothy Monekosso, and Paolo Remagnino. 2019. Summarizing videos with attention. *Preprint*, arXiv:1812.01969.

Junaid Ahmed Ghauri, Sherzod Hakimov, and Ralph Ewerth. 2021. Supervised video summarization via multiple feature sets with parallel attention. *Preprint*, arXiv:2104.11530.

Simon Ging, Mohammadreza Zolfaghari, Hamed Pirsiavash, and Thomas Brox. 2020. Coot: Cooperative hierarchical transformer for video-text representation learning. *Preprint*, arXiv:2011.00597.

Hannah Gonzalez, Jiening Li, Helen Jin, Jiaxuan Ren, Hongyu Zhang, Ayotomiwa Akinyele, Adrian Wang, Eleni Miltsakaki, Ryan Baker, and Chris Callison-Burch. 2023. Automatically generated summaries of video lectures may enhance students' learning experience. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 382–393, Toronto, Canada. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, and Alex Vaughan et al. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Pawara Gunawardena, Heshan Sudarshana, Oshada Amila, Rashmika Nawaratne, Damminda Alahakoon, Amal S. Perera, and Charith Chitraranjan. 2019. Interest-oriented video summarization with keyframe extraction. In *2019 19th International Conference on Advances in ICT for Emerging Regions (ICTer)*, volume 250, pages 1–8.

Michael Gygli, Helmut Grabner, Hayko Riemenschneider, and Luc Van Gool. 2014. Creating summaries from user videos. In *European conference on computer vision*, pages 505–520. Springer.

Bo He, Jun Wang, Jielin Qiu, Trung Bui, Abhinav Shrivastava, and Zhaowen Wang. 2023. Align and attend multimodal summarization with dual contrastive losses. *Preprint*, arXiv:2303.07284.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep residual learning for image recognition. *Preprint*, arXiv:1512.03385.

Jack Hessel, Bo Pang, Zhenhai Zhu, and Radu Soricut. 2019. A case study on combining asr and visual features for generating instructional video captions. *Preprint*, arXiv:1910.02930.

Xiaohu Huang, Hao Zhou, and Kai Han. 2024. Prunevid: Visual token pruning for efficient video large language models. *Preprint*, arXiv:2412.16117.

Hai-Dang Huynh-Lam, Ngoc-Phuong Ho-Thi, Minh-Triet Tran, and Trung-Nghia Le. 2024. *Cluster-Based Video Summarization with Temporal Context Awareness*, page 15–28. Springer Nature Singapore.

Vladimir Iashin and Esa Rahtu. 2020. A better use of audio-visual cues: Dense video captioning with bi-modal transformer. *Preprint*, arXiv:2005.08271.

MinJu Jeon, Si-Woo Kim, Ye-Chan Kim, HyunGee Kim, and Dong-Jin Kim. 2025. Sali4vid: Saliency-aware video reweighting and adaptive caption retrieval for dense video captioning. *Preprint*, arXiv:2509.04602.

Mingda Jia, Weiliang Meng, Zenghuang Fu, Yiheng Li, Qi Zeng, Yifan Zhang, Ju Xin, Rongtao Xu, Jiguang Zhang, and Xiaopeng Zhang. 2025. Explicit temporal-semantic modeling for dense video captioning via context-aware cross-modal interaction. *Preprint*, arXiv:2511.10134.

Hao Jiang and Yadong Mu. 2022. Joint video summarization and moment localization by cross-task sample transfer. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16367–16377.

Yongrae Jo, Seongyun Lee, Aiden SJ Lee, Hyunji Lee, Hanseok Oh, and Minjoon Seo. 2023. Zero-shot dense video captioning by jointly optimizing text and moment. *Preprint*, arXiv:2307.02682.

Marc Junyent, Pablo Beltran, Miquel Farré, Jordi Pont-Tuset, Alexandre Chapiro, and Aljoscha Smolic. 2015. Video content and structure description based on keyframes, clusters and storyboards. In *2015 IEEE 17th International Workshop on Multimedia Signal Processing (MMSP)*, pages 1–6.

Chandresh S. Kanani, Sriparna Saha, and Pushpak Bhattacharyya. 2021. Global object proposals for improving multi-sentence video descriptions. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7.

R. Killick, P. Fearnhead, and I. A. Eckley. 2012. Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association*, 107(500):1590–1598.

Minkuk Kim, Hyeon Bae Kim, Jinyoung Moon, Jinwoo Choi, and Seong Tae Kim. 2024a. Do you remember? dense video captioning with cross-modal memory retrieval. *Preprint*, arXiv:2404.07610.

Minkuk Kim, Hyeon Bae Kim, Jinyoung Moon, Jinwoo Choi, and Seong Tae Kim. 2024b. Hicm^2: Hierarchical compact memory modeling for dense video captioning. *Preprint*, arXiv:2412.14585.

Dohwan Ko, Joonmyung Choi, Hyeong Kyu Choi, Kyoung-Woon On, Byungseok Roh, and Hyunwoo J. Kim. 2023. Meltr: Meta loss transformer for learning to fine-tune video foundation models. *Preprint*, arXiv:2303.13009.

Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. 2017. Dense-captioning events in videos. *Preprint*, arXiv:1705.00754.

Alon Lavie and Abhaya Agarwal. 2007. Meteor: an automatic metric for mt evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, StatMT '07, page 228–231, USA. Association for Computational Linguistics.

Min Jung Lee, Dayoung Gong, and Minsu Cho. 2025. Video summarization with large language models. *Preprint*, arXiv:2504.11199.

Zhuo Lei, Chao Zhang, Qian Zhang, and Guoping Qiu. 2019. Framerank: A text processing approach to video summarization. *Preprint*, arXiv:1904.05544.

Haopeng Li, Qiuhong Ke, Mingming Gong, and Tom Drummond. 2023. Progressive video summarization via multimodal self-supervised learning. In *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 5573–5582.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. *Preprint*, arXiv:2201.12086.

WenLin Li, DeYu Qi, ChangJian Zhang, Jing Guo, and JiaJun Yao. 2020. Video summarization based on mutual information and entropy sliding window method. *Entropy*, 22(11).

Yizhe Li, Sanping Zhou, Zheng Qin, and Le Wang. 2025. Pr-detr: Injecting position and relation prior for dense video captioning. *Preprint*, arXiv:2506.16082.

Guoqiang Liang, Yanbing Lv, Shucheng Li, Xiahong Wang, and Yanning Zhang. 2022. Video summarization with a dual-path attentive network. *Neurocomputing*, 467(C):1–9.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Jingyang Lin, Jialian Wu, Ximeng Sun, Ze Wang, Jiang Liu, Yusheng Su, Xiaodong Yu, Hao Chen, Jiebo Luo, Zicheng Liu, and Emad Barsoum. 2025. Unleashing hour-scale video training for long video-language understanding. *Preprint*, arXiv:2506.05332.

Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, Jason Li, Taroon Bharti, and Ming Zhou. 2020. Univl: A unified video and language pre-training model for multimodal understanding and generation. *Preprint*, arXiv:2002.06353.

Behrooz Mahasseni, Michael Lam, and Sinisa Todorovic. 2017. Unsupervised video summarization with adversarial lstm networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2982–2991.

Engin Mendi, Hélio B. Clemente, and Coskun Bayrak. 2013. Sports video summarization based on motion analysis. *Computers and Electrical Engineering*, 39(3):790–796. Special issue on Image and Video Processing Special issue on Recent Trends in Communications and Signal Processing.

Medhini Narasimhan, Anna Rohrbach, and Trevor Darrell. 2021. Clip-it! language-guided video summarization. *Preprint*, arXiv:2107.00650.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, and Shyamal Anadkat et al. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA. Association for Computational Linguistics.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. *Preprint*, arXiv:2103.00020.

Mrigank Rochan, Linwei Ye, and Yang Wang. 2018. Video summarization using fully convolutional sequence networks. *Preprint*, arXiv:1805.10538.

Sunil S Harakannanavar, Shaik Sameer, Vikash Kumar, Sunil B, Adithya A, and Veena Puranikmath. 2022. Robust video summarization algorithm using supervised machine learning. *Global Transitions Proceedings*, 3.

Sharanjeet Kaur Sandhu and Anupam Agarwal. 2015. Summarizing videos by key frame extraction using ssim and other visual features. In *Proceedings of the Sixth International Conference on Computer and Communication Technology 2015*, ICCCT '15, page 209–213, New York, NY, USA. Association for Computing Machinery.

Aidean Sharghi, Jacob S. Laurel, and Boqing Gong. 2017. Query-focused video summarization: Dataset, evaluation, and a memory network based approach. *Preprint*, arXiv:1707.04960.

Botian Shi, Lei Ji, Yaobo Liang, Nan Duan, Peng Chen, Zhendong Niu, and Ming Zhou. 2019. Dense procedure captioning in narrated instructional videos. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6382–6391, Florence, Italy. Association for Computational Linguistics.

Jaewon Son, Jaehun Park, and Kwangsu Kim. 2024. Csta cnn-based spatiotemporal attention for video summarization. *Preprint*, arXiv:2405.11905.

Yale Song, Jordi Vallmitjana, Amanda Stent, and Alejandro Jaimes. 2015. Tvsum: Summarizing web videos using titles. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5179–5187.

Chen Sun, Fabien Baradel, Kevin Murphy, and Cordelia Schmid. 2019a. Learning video representations using contrastive bidirectional transformer. *Preprint*, arXiv:1906.05743.

Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. 2019b. Videobert: A joint model for video and language representation learning. *Preprint*, arXiv:1904.01766.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2014. Going deeper with convolutions. *Preprint*, arXiv:1409.4842.

Kailong Tan, Yuxiang Zhou, Qianchen Xia, Rui Liu, and Yong Chen. 2024. Large model based sequential keyframe extraction for video summarization. *Preprint*, arXiv:2401.04962.

Keda Tao, Can Qin, Haoxuan You, Yang Sui, and Huan Wang. 2025. Dycoke: Dynamic compression of tokens for fast video large language models. *Preprint*, arXiv:2411.15024.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, and Morgane Rivière et al. 2025. Gemma 3 technical report. *Preprint*, arXiv:2503.19786.

Hacene Terbouche, Maryan Morel, Mariano Rodriguez, and Alice Othmani. 2023. Multi-annotation attention model for video summarization. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 3143–3152.

Junyan Wang, Yang Bai, Yang Long, Bingzhang Hu, Zhenhua Chai, Yu Guan, and Xiaolin Wei. 2020. Query twice dual mixture attention meta learning for video summarization. In *Proceedings of the 28th ACM International Conference on Multimedia*, MM 20, page 4023–4031. ACM.

Teng Wang, Jinrui Zhang, Feng Zheng, Wenhao Jiang, Ran Cheng, and Ping Luo. 2023. Learning grounded vision-language representation for versatile understanding in untrimmed videos. *Preprint*, arXiv:2303.06378.

Teng Wang, Ruimao Zhang, Zhichao Lu, Feng Zheng, Ran Cheng, and Ping Luo. 2021. End-to-end dense video captioning with parallel decoding. *Preprint*, arXiv:2108.07781.

Ziyi Wang, Bo Lu, Yonghao Long, Fangxun Zhong, Tak-Hong Cheung, Qi Dou, and Yunhui Liu. 2022. Autolaparo: A new dataset of integrated multi-tasks for image-guided surgical automation in laparoscopic hysterectomy. *Preprint*, arXiv:2208.02049.

W. Wolf. 1996. Key frame selection by motion analysis. In *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, volume 2, pages 1228–1231 vol. 2.

Kangyi Wu, Pengna Li, Jingwen Fu, Yizhe Li, Yang Wu, Yuhan Liu, Jinjun Wang, and Sanping Zhou. 2025. Event-equalized dense video captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8417–8427.

Zhuyang Xie, Yan Yang, Yankai Yu, Jie Wang, Yongquan Jiang, and Xiao Wu. 2024. Exploring temporal event cues for dense video captioning in cyclic co-learning. *Preprint*, arXiv:2412.11467.

Antoine Yang, Arsha Nagrani, Paul Hongsuck Seo, Antoine Miech, Jordi Pont-Tuset, Ivan Laptev, Josef Sivic, and Cordelia Schmid. 2023. Vid2seq: Large-scale pretraining of a visual language model for dense video captioning. *Preprint*, arXiv:2302.14115.

Shu Yang, Luyang Luo, Qiong Wang, and Hao Chen. 2024. Surgformer: Surgical transformer with hierarchical temporal attention for surgical phase recognition. *Preprint*, arXiv:2408.03867.

Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, Qianyu Chen, Huarong Zhou, Zhensheng Zou, Haoye Zhang, Shengding Hu, Zhi Zheng, Jie Zhou, Jie Cai, Xu Han, and 4 others. 2024. Minicpm-v: A gpt-4v level mllm on your phone. *Preprint*, arXiv:2408.01800.

Sicheng Yu, Chengkai Jin, Huanyu Wang, Zhenghao Chen, Sheng Jin, Zhongrong Zuo, Xiaolei Xu, Zhenbang Sun, Bingni Zhang, Jiawei Wu, Hao Zhang, and Qianru Sun. 2025. Frame-voyager: Learning to query frames for video large language models. *Preprint*, arXiv:2410.03226.

Ke Zhang, Wei-Lun Chao, Fei Sha, and Kristen Grauman. 2016. Video summarization with long short-term memory. *Preprint*, arXiv:1605.08110.

Sheng Zhang, Yanbo Xu, Naoto Usuyama, Hanwen Xu, Jaspreet Bagga, Robert Tinn, Sam Preston, Rajesh Rao, Mu Wei, Naveen Valluri, Cliff Wong, Andrea Tupini, Yu Wang, Matt Mazzola, Swadheen Shukla,

Lars Liden, Jianfeng Gao, Angela Crabtree, Brian Piening, and 5 others. 2025. Biomedclip: a multimodal biomedical foundation model pretrained from fifteen million scientific image-text pairs. *Preprint*, arXiv:2303.00915.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. *Preprint*, arXiv:1904.09675.

Yujia Zhang, Michael Kampffmeyer, Xiaodan Liang, Min Tan, and Eric P. Xing. 2018. Query-conditioned three-player adversarial network for video summarization. *Preprint*, arXiv:1807.06677.

Yujia Zhang, Qianzhong Li, Xiaoguang Zhao, and Min Tan. 2021. Robot learning through observation via coarse-to-fine grained video summarization. *Applied Soft Computing*, 99:106913.

Yunzuo Zhang, Yameng Liu, Weili Kang, and Ran Tao. 2023. Vss-net visual semantic self-mining network for video summarization. *IEEE Transactions on Circuits and Systems for Video Technology*, PP:1–1.

Bin Zhao, Maoguo Gong, and Xuelong Li. 2022. Hierarchical multimodal transformer to summarize videos. *Neurocomputing*, 468:360–369.

Bin Zhao, Xuelong Li, and Xiaoqiang Lu. 2018. Hsa-rnn hierarchical structure-adaptive rnn for video summarization. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7405–7414.

Cairong Zhao, Chutian Wang, Zifan Song, Guosheng Hu, Haonan Chen, and Xiaofan Zhai. 2024. Cap2sum: Learning to summarize videos by generating captions. *Preprint*, arXiv:2408.12800.

Kaiyang Zhou, Yu Qiao, and Tao Xiang. 2018a. Deep reinforcement learning for unsupervised video summarization with diversity-representativeness reward. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Luowei Zhou, Nathan Louis, and Jason J. Corso. 2018b. Weakly-supervised video object grounding from text by loss weighting and object interaction. *Preprint*, arXiv:1805.02834.

Luowei Zhou, Yingbo Zhou, Jason J. Corso, Richard Socher, and Caiming Xiong. 2018c. End-to-end dense video captioning with masked transformer. *Preprint*, arXiv:1804.00819.

Xingyi Zhou, Anurag Arnab, Shyamal Buch, Shen Yan, Austin Myers, Xuehan Xiong, Arsha Nagrani, and Cordelia Schmid. 2024. Streaming dense video captioning. *Preprint*, arXiv:2404.01297.

Linchao Zhu and Yi Yang. 2020. Actbert: Learning global-local video-text representations. *Preprint*, arXiv:2011.07231.

Wencheng Zhu, Yucheng Han, Jiwen Lu, and Jie Zhou. 2022. Relational reasoning over spatial-temporal graphs for video summarization. *IEEE Transactions on Image Processing*, 31:3017–3031.

Wencheng Zhu, Jiwen Lu, Jiahao Li, and Jie Zhou. 2021. Dsnet: A flexible detect-to-summarize network for video summarization. *IEEE Transactions on Image Processing*, 30:948–962.

Appendix

A. Experimental Settings

All experiments were conducted on NVIDIA A100 GPUs (40GB HBM2 memory per GPU). The software stack consisted of Python 3.10, PyTorch 2.1, and CUDA 11.8. Inference of vision and language models was performed using the Ollama library and OpenAI GPT-4 API, with a temperature set to 0.0 for deterministic outputs where applicable. Data preprocessing and frame extraction used OpenCV, with all videos sampled at 1 frame per second (fps) for efficiency. ResNet18 was used with torchvision's pre-trained models.

B. Dataset Statistics

Table 5 presents key statistics for YouCook2 and ActivityNet Captions datasets, including the number of videos and average video duration. These statistics provide essential context for the scale and diversity of each dataset, helping to interpret experimental results and facilitate reproducibility. All ablation studies and sensitivity analyses in this work were performed exclusively on the validation sets, as test labels are not publicly available.

Dataset	#Videos	Avg Length
YouCook2	414	5.26
ActivityNet Captions	4917	2

Table 5: Summary statistics for the official dataset. Avg Length denotes average time duration of the videos in minutes. Videos denotes number of videos.

For all experiments, videos were sampled at 1 frame per second (fps) to ensure a balance between comprehensive temporal coverage and manageable computational resource usage. The combination of YouCook2 and ActivityNet Captions enables us to comprehensively evaluate our pipeline's effectiveness in diverse real-world scenarios. YouCook2 consists of long-form cooking videos, where each video follows a detailed procedural sequence to fulfill a recipe. ActivityNet Captions encompasses

Predicted and ground truth video captions	Scores
Segment 1 **Predicted:** In this instructional tutorial, Amelia Smith shares her professional insights on achieving voluminous curls using hot roller pins. **Ground Truth:** A lady is combing another lady's hair as the title is shown.	BLEU: 1.3 ROUGE-L: 0.0 METEOR: 3.4 BERTScore F1:84.4 Sem-nCG: 100 LLM Judge: 20
Segment 2 **Predicted:** The video provides a step-by-step approach, beginning with a demonstration of necessary tools—styling comb, rollers, and pins—emphasizing the superior functionality of alligator-type clips. **Ground Truth:** We see the supplies the lady will be using as she talks to the camera.	BLEU: 1.2 ROUGE-L: 9.3 METEOR: 11.7 BERTScore F1:86.1 Sem-nCG 100 LLM Judge: 63.4
Segment 3 **Predicted:** Amelia guides viewers through the process of preparing the hair: combing, sectioning, and properly aligning rollers for optimal curl definition. **Ground Truth:** The stylist combs the customer's hair, and adds a roller.	BLEU: 3.1 ROUGE-L: 32.2 METEOR: 37.7 BERTScore F1:88.3 Sem-nCG: 100 LLM Judge: 50
Segment 4 **Predicted:** She offers practical tips, such as adjusting the pin's angle to ensure a secure hold, adapting the pin placement depending on the roller orientation, and modifying pins over time to maintain their snugness. **Ground Truth**: The stylist removes the roller and adds another.	BLEU: 1.9 ROUGE-L: 19 METEOR: 31.2 BERTScore F1:86.2 Sem-nCG: 100 LLM Judge: 23.4
Segment 5 **Predicted:** The tutorial concludes with essential final adjustments, showcasing Amelia's dedication to detail in setting the perfect hairstyle. **Ground Truth:** The stylist removes that one and adds a vertical one.	BLEU: 1.5 ROUGE-L: 7.1 METEOR: 8.4 BERTScore F1:86.4 Sem-nCG: 100 LLM Judge: 30
Segment 6 **Predicted:** Her approachable explanations and expert advice empower viewers to confidently craft stunning curls, fostering both beginner and seasoned stylists to enhance their hairstyling skills from the comfort of home. **Ground Truth**: The stylist speaks and the video end	BLEU: 1.2 ROUGE-L: 11.1 METEOR: 19.4 BERTScore F1:85.3 Sem-nCG: 100 LLM Judge: 33.4

Figure 5: Predicted and ground-truth video captions with scores for a sample video from the ActivityNet Captions dataset.

a wide variety of activities, ranging from sports and hobbies to daily routines represented by much shorter video segments. This diversity allows us to demonstrate our pipeline's ability to deliver robust content coverage and summarization quality across both long, structured instructional content and shorter, more varied event-driven videos, while operating under practical computational constraints.

C. Evaluating Video Captioning Performance

All scores here are scaled to 0-100. Figure 5 represents the PRISM-generated caption vs. ground truth on a sample video from ActivityNet Captions dataset; the predicted captions capture the instructional flow of the video, aligning well with the ground truth across segments. Despite minor lexical variations, high BERTScore and consistent LLM Judge scores indicate strong semantic overlap and overall quality. The model demonstrates a robust ability to generate coherent, contextually appropriate summaries.

Figure 6 shows the PRISM-generated caption vs. ground truth on a sample video from YouCook2 dataset. The predicted captions closely align with the ground truth, especially in capturing sequential steps, as reflected in the high BERTScore and LLM Judge values. This highlights the system's effectiveness in both procedural understanding and

Predicted and ground truth video captions	Scores
Segment 1 **Predicted**: Begin by boiling water in a large pot. Once boiling, add miso paste, soy sauce, frozen veggies, and mushrooms, then let the mixture sit for a few minutes. **Ground Truth**: add miso paste soy sauce frozen veggies and the mushrooms to the pot of water. Mix and boil the ingredients.	BLEU: 6.3 ROUGE-L: 41.7 METEOR: 51.7 BERTScore: 91.2 Sem-nCG: 100 LLM Judge: 80
Segment 2 **Predicted**: Once the broth starts boiling again, add the pre-cooked udon noodles and let them cook for a couple of minutes. **Ground Truth**: add some udon noodles to the broth.	BLEU: 4.2 ROUGE-L: 21.4 METEOR: 45.3 BERTScore F1: 90.7 Sem-nCG: 100 LLM Judge: 66.6
Segment 3 **Predicted**: Finally, incorporate shredded green onions and cubed tofu into the pot. Throughout the process, monitor the cooking to ensure everything is perfectly done. Serve the soup hot, enjoying the blend of flavors. **Ground Truth**: add some leaves of chard and tofu to the broth.	BLEU: 1.2 ROUGE-L: 19 METEOR: 21.9 BERTScore F1: 88.4 Sem-nCG: 100 LLM Judge: 50

Figure 6: Predicted and ground-truth video captions with scores for a sample video from the YouCook2 dataset.

summarization of activities or tasks with shorter durations.

D. Implementation Details

For each dataset, videos were downloaded and processed according to the original dataset splits. Frames were extracted at 1 fps, then the frame feature difference threshold was set to 30 unless otherwise noted. Label assignment was conducted using the specified VLM (Gemma3, MiniCPM, GPT-4V), followed by GPT-4-based semantic filtering with temperature 0 for deterministic output. Label–frame pairs required cosine similarity ≥ 0.9 with a vision encoder (e.g., CLIP) to be accepted. Sliding window-based summaries were generated and aggregated using LLMs as described in the main paper. No manual annotation or fine-tuning was performed.

E. Keyframe Selection

Our pipeline targets video summarization with a focus on keyframe selection guided by semantic and temporal alignment. We evaluate this using the TVSum and SumMe datasets. TVSum consists of 50 user annotated videos across various categories with frame-level importance scores (Song et al., 2015), while SumMe contains 25 user-generated summaries for videos from various scenes of daily life (Gygli et al., 2014). Following the evaluation in the CSTA framework(Son et al., 2024), we align with their observation that traditional metrics like F1 score may not adequately capture the alignment between predicted and human-generated summaries, especially when ground truth annotations vary across users. Instead, we also adopt

Kendall's τ and Spearman's ρ to assess ranking correlation between predicted and ground truth importance scores. Table 6 presents a comprehensive comparison of the TVSum and SumMe datasets. These datasets contain short videos with diverse content and frame level human annotations. Our method, PRISM, ranks competitively among recent state-of-the-art models, particularly achieving $\tau = 0.1406$ and $\rho = 0.18$ on TVSum, while reaching $\tau = 0.1617$ and $\rho = 0.2212$ on SumMe. We attribute this to our algorithm's ability to identify semantically rich anchors and perform label-aware clustering. For the TVSum and SumMe experiments, we compute frame importance scores using a layered approach. First, we normalize ResNet feature differences. Then, during adaptive sampling, we re-normalize within frame groups. We also add CLIP-based vision encoder scores. At the grouping stage, we assign weights based on label distribution across four frames. If all labels are the same (1:1:1:1), each frame gets 0.25. For 3:1, the majority gets 0.75, the minority 0.25. In a 2:2 split, all receive 0.5. In 2:1:1, the majority gets 0.5, and others get 0.25. All scores are averaged per frame, and if any frame gets dropped at a stage, it gets +0 for every stage after that. The abbreviations in brackets denote the input modalities used by each method: (T) indicates temporal features, (S) spatial features, and (M) multimodal inputs (e.g., vision-language or audio-visual). (ST) refers to models that jointly leverage both spatial and temporal features.

F. Hyperparameter Sensitivity

We conduct hyperparameter sensitivity experiments using a stable configuration of Stage 2 threshold = 0.9, Stage 1 threshold = 30, and adaptive sampling batch size = 10, unless stated otherwise. When a specific setting is mentioned (e.g., Stage 2 = 0.5), only that parameter is varied, while the others remain fixed. These experiments are performed on both YouCook2 and ActivityNet Captions. All evaluation metrics and relevant frame-label statistics are reported to ensure transparency and reproducibility. All scores are scaled to 0-100.

We chose these defaults as a representative baseline, as a lower Stage 2 threshold like 0.5 might intuitively select more frames; in our setup the number of high-scoring frames remains stable due to the nature of the datasets and label distributions. In longer videos (e.g., 40 minutes or more) or high-stakes domains (e.g., surgical videos), strict thresholds (e.g., 0.9 in stage 2) would likely filter out noise or ambiguity in frames, resulting in more significant differences.

For Stage 1, in the ResNet frame feature difference threshold, we compare thresholds of 10 vs. 30 vs. 50, where a lower value retains more candidate frames, and a higher value aims to significantly reduce redundancy. While experimenting with ResNet variants (ResNet-18, ResNet-50, etc.), we observed that deeper networks like ResNet-50 result in fewer frames passing Stage 1, reducing diversity in Stage 1 itself. Hence, we prioritize maintaining a diverse and sufficient sample for Stage 2 and downstream processing, settling on ResNet-18 as a practical trade-off. We also explore adaptive sampling batch sizes (10 vs. 15), where smaller sizes tend to generate more label diversity, and larger sizes produce more conservative label sets.

Note: *Stage 2* refers to the vision encoder similarity threshold , *Stage 1* is the ResNet frame feature difference threshold, and *adaptive* indicates the batch size for adaptive sampling.

These tables provide a comprehensive view of how key hyperparameters affect both summary quality metrics and frame/label statistics on the two primary video summarization datasets in our study. The hyperparameter sensitivity experiment shows that summary quality on YouCook2 and ActivityNet Captions remains consistent to changes in hyperparameters, with little effect on output quality. It should also be noted that in long-form videos (e.g., 40 minutes or more) with a higher visual similarity (e.g., colour consistency, less movement, etc.) amongst frames, small hyperparameter adjustments can result in significantly fewer or more selected frames. This highlights that while activities or instructional domains allow flexible tuning, careful hyperparameter selection is crucial for datasets with visually similar frames, especially in high-stakes domains.

G. Prompts used in the pipeline

Throughout this code pipeline, we use prompts at multiple stages: beginning with frame-level visual descriptions via vision-language models, followed by label generation using GPT or LLaMA from those outputs, validation of those labels, detailed multi-frame image descriptions, and finally recursive and integrated summarization of chunked out-

Method	SumMe			TVSum		
	Rank	τ	ρ	Rank	τ	ρ
Random	–	0.000	0.000	–	0.000	0.000
Human	–	0.205	0.213	–	0.177	0.204
dppLSTM (Zhang et al., 2016)	15	0.040	0.049	21	0.042	0.055
HSA-RNN (Zhao et al., 2018)	12.5	0.064	0.066	19.5	0.082	0.088
DAN (Liang et al., 2022)	–	–	–	19.5	0.071	0.099
DSNet-AB (Zhu et al., 2021)	14.5	0.051	0.059	15	0.108	0.129
HMT (Zhao et al., 2022)	11.5	0.079	0.080	17.5	0.096	0.107
CLIP-It (Narasimhan et al., 2021)	–	–	–	13.5	0.108	0.147
iPTNet (Jiang and Mu, 2022)	9.5	0.101	0.119	11	0.134	0.163
A2Summ (He et al., 2023)	8	0.108	0.129	10.5	0.137	0.165
VASNet (Fajtl et al., 2019)	7	0.160	0.170	9	0.160	0.170
AAAM (Terbouche et al., 2023)	–	–	–	6.5	0.169	0.223
MAAM (Terbouche et al., 2023)	–	–	–	5.5	0.179	0.236
VSS-Net (Zhang et al., 2023)	–	–	–	3	0.190	0.249
DMASum (Wang et al., 2020)	11	0.063	0.089	**1**	**0.203**	**0.267**
RR-STG (Zhu et al., 2022)	2.5	0.211	0.234	7.5	0.162	0.212
MSVA (Ghauri et al., 2021)	3.5	0.200	0.230	5.5	0.190	0.210
SSPVS (Li et al., 2023)	3*	0.192	0.257	4.5	0.181	0.238
GoogleNet (Szegedy et al., 2014)	5	0.176	0.197	11.5	0.129	0.163
CSTA (Son et al., 2024)	**1**	**0.246**	**0.274**	2	0.194	0.255
PRISM	6	0.162	0.221	9.5	0.141	0.180

Table 6: Comparison with prior video summarization methods on SumMe and TVSum using rank, Kendall's τ, and Spearman's ρ. T, M, and ST denote text-based, multimodal, and spatio-temporal methods, respectively.

puts to generate a coherent activity summary. The dataset description refers to a one-line description about the videos in the dataset (e.g., YouCook2: instructional cooking video).

- **Frame Description Prompt**

  Explain what is happening in the image. This is a frame from a video of an activity <dataset description>

- **Label Generation Prompt**

  Extract info from: {VLM_output}
  Tell me in few words (5–6) by giving a label for the most important details that you find from the text description of the activity happening in the image.
  Don't make the label vague like only the heading of the main activity; tell me exactly what is going on in the image and only give the label in simple words.

- **Label Validation Prompt**

  Imagine you are an expert on validating labels. Given this label: {label}, do you think it is valid to check an image in an important video?
  For example, is it not useful (like a black screen) or too general (like "this is a surgery video" or "this is a girl standing")?
  I don't want labels that are not useful, say nothing about the image, or are too general.
  Based on your judgement, return 0 if you think this is unimportant or too general; return 1 if you think it is an important label. Return only a number, nothing else.

- **Combined Frame Description Prompt**

  Context: This is a combination of 4 images from a video of an activity spanning <dataset description>
  Each small picture represents a step in the sequence:
  The current major label is '{majority_label}'.
  - Top-right: Step 2
  - Top-left: Step 1

Setting	BERT	BLEU	METEOR	ROUGE	Sel. Frames	Ext. Frames	Filt. Frames	Labels /Video
Stage 2: 0.5	84.87	2.31	31.98	24.53	58.67	325.13	60.51	9.54
Stage 2: 0.7	84.70	2.29	32.61	24.29	58.87	324.87	60.46	9.49
Stage 2: 0.9	85.00	2.42	32.44	24.78	58.90	324.73	60.48	9.53
Stage 1: 10	**85.06**	2.24	**35.33**	**25.32**	59.40	324.73	60.98	9.53
Stage 1: 30	84.91	2.35	33.58	25.04	58.60	324.73	60.48	9.53
Stage 1: 50	84.91	2.27	33.65	25.15	59.22	324.73	60.47	9.52
Adaptive: 10	84.92	2.22	34.55	25.26	59.29	324.73	60.98	9.53
Adaptive: 15	84.77	2.17	33.03	24.97	58.36	324.73	60.98	9.53

Table 7: Hyperparameter sensitivity analysis on YouCook2. We report semantic and lexical metrics, selected/extracted/filtered frames, and labels per video.

Setting	BERT	BLEU	METEOR	ROUGE	Sel. Frames	Ext. Frames	Filt. Frames	Labels /Video
Stage 2: 0.5	82.88	0.83	19.03	12.38	17.90	116.28	20.83	3.55
Stage 2: 0.7	83.03	0.86	20.05	12.43	19.16	120.53	21.61	3.68
Stage 2: 0.9	**83.23**	**1.00**	19.41	13.00	17.90	116.28	20.83	3.55
Stage 1: 10	83.20	0.89	19.47	12.96	18.36	118.00	21.21	3.59
Stage 1: 30	82.70	0.83	18.65	12.26	19.16	119.95	21.47	3.68
Stage 1: 50	83.22	0.99	19.77	12.71	18.11	119.89	21.05	3.53
Adaptive: 10	83.16	0.86	19.83	12.85	18.53	120.11	21.50	3.68
Adaptive: 15	83.06	0.91	**20.38**	**13.08**	18.67	118.00	21.13	3.62

Table 8: Hyperparameter sensitivity analysis on ActivityNet Captions. We report semantic and lexical metrics, selected/extracted/filtered frames, and labels per video.

- Bottom-right: Step 4
- Bottom-left: Step 3

Provide a detailed description for each of the 4 images.

- **Recursive Summarization Prompt**

  You are summarizing partial text from a video of an activity spanning <dataset description>.
  Try to give a name to this activity (e.g., a lady doing movements could be dancing). Try to correlate the different activities to speak for one main act or theme.
  Rewrite the text in a frame-wise, narrative format with transitions between steps:

  {chunk}

- **Final Integration Prompt**

  We have a final summary of video frames:

  {final_summary_text}

  We also have the raw transcript from the entire audio:

  {transcript_text}

  Please produce a unified, cohesive summary of all the steps in the activity happening in the video, which could be related to one of these: <dataset description>
  Incorporate relevant information from the audio transcript. If the transcript provides additional details or clarifications, weave them into the final summary.
  If the transcript includes extraneous content, omit it.
  Focus on a coherent storyline of the entire action or activity of the video.

H. Limitations and Error Analysis

Despite the use of dual check mechanisms such as vision-language model-based captioning + vision encoder filtering, limitations still exist. One major challenge is model consistency, particularly with respect to hallucinations and mislabeling. Both vision-language and language models can generate plausible but incorrect outputs when operating without sufficient context. For instance, in a cooking video, the model may misinterpret ground beef as chicken based on visual similarity alone. Without audio cues or additional metadata, the generated summary may inaccurately describe the recipe as involving chicken, leading to a consistent but factually incorrect narrative. These issues tend to surface in visually homogeneous or ambiguous settings, where subtle distinctions are difficult to detect. While our multimodal encoder-based filtering strategies help mitigate such errors, isolated failures still occur. In high-stakes domains, this can be mitigated to an extent using fine-tuned vision encoders and vision-language models. Additionally, the dual-stage validation process introduces increased computational overhead compared to simpler pipelines. However, this trade-off is necessary for improved reliability in the generated summaries.