DO NOT STOP AT FRAME SCORING: LEARNING TO COMPOSE VIDEO SUMMARIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Video summarization is a task to extract a sequence of segments to preserve the key content of a video. Recent work on this task tends to focus only on predicting frame importance scores, leaving the segmentation and budget-constrained selection as a fixed pipeline. However, our empirical analysis reveals that a more precise importance score does not reliably translate into a better summary, suggesting that summary composition itself should be trained. Directly learning to compose a summary, however, requires structured summary-level supervision, which is prohibitive to obtain at scale. To address this bottleneck, we introduce offline pseudo-summary construction, transforming existing frame-importance annotations and video content into a structured, budget-aware pseudo-summary. We then introduce SumComposer, a model to autoregressively add a segment to the summary, conditioned on preceding selections. Once trained, it requires neither importance annotations nor language model calls to summarize a video. Our experiments verify that our pseudo-summaries are better aligned with human annotations than fixed construction, and SumComposer trained on these pseudo-labels substantially improves matching and boundary alignment over fixed composition.

1 INTRODUCTION

Video summarization condenses a long video into a short sequence of segments that preserves its key content within a limited time budget, helping viewers browse and understand it without watching in full. However, manually creating summary labels from a raw video is prohibitively costly, since summarizing a single video requires annotators to skim the entire video, divide it into coherent segments with semantic boundaries, compare their relative importance, and select the most relevant segments under a summary duration budget. As a result, datasets providing manually annotated summaries (Gygli et al., 2014; Song et al., 2015; Patel et al., 2026) remain extremely small in scale.

Against this scarcity of summary-level supervision, much of prior work (Son et al., 2024; Lee et al., 2025; Guo et al., 2026) has formulated video summarization as frame-level importance prediction, leaving the construction of the final summary to a fixed composition pipeline. As illustrated in Fig. 1(a), they first predict frame-level saliency scores, and then convert them into a final summary through fixed segmentation (*e.g.*, KTS (Potapov et al., 2014)) and budget-constrained selection (*e.g.*, knapsack). This formulation reduces the summarization problem to estimating a relevance score at each temporal position. This frame-level importance supervision scales naturally to large datasets, since such signals can be collected from viewer behaviors on video-sharing platforms, as demonstrated by Mr. HiSum (Sul et al., 2023) and MoSu (Kim et al., 2026).

The decomposition relies on an expectation that a more precise importance estimation will translate into a better video summary, but this assumption does not necessarily hold in reality. This is because the frame-level importance alone does not specify the composition decisions that determine the final summary. Otani et al. (2019) showed that fixed segmentation and selection can substantially shape the resulting summary, and our motivational diagnosis in Sec. 2 reconfirms that improvements in frame-importance estimation do not reliably translate into better final summaries under fixed composition (Fig. 2). These observations motivate us to directly train composing a summary.

Some prior works (Zhao et al., 2018; Gong et al., 2014; Zhu et al., 2021) have explored aspects of summary composition beyond frame-level importance prediction, including temporal structure and sequential selection. Recent approaches (Hua et al., 2025; Argaw et al., 2024) have also explored

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

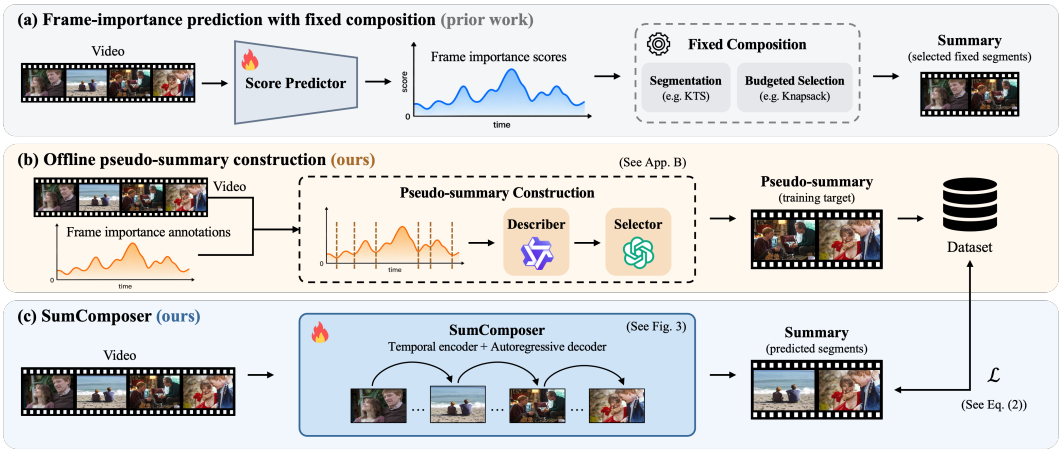


Figure 1: **From importance to composition.** (a) Frame-importance prediction with fixed KTS+knapsack composition. (b) Pseudo-summary construction from video content and importance annotations. (c) SumComposer predicts budgeted summary segments from video features.

LLM-based summary selection or structured supervision. However, these methods either address the composition process only partially, rely on fixed construction rules, or require additional modalities. For this reason, it is still an open question how to obtain scalable summary-level supervision for learning budget-aware summary composition.

To address this problem, we propose to **construct a pseudo-summary** that transforms large-scale frame-importance signals and video content into structured summary-level supervision, as outlined in Fig. 1(b). We partition the video into candidate segments, describe their visual content using a vision-language model (VLM), and select a diverse and relevant subset under the summary budget employing a large language model (LLM). The selected segments constructs a pseudo-summary to supervise segment selection and temporal boundaries. This offline construction enables summary-level supervision at scale without manual efforts. Further details are presented in Sec. 3.

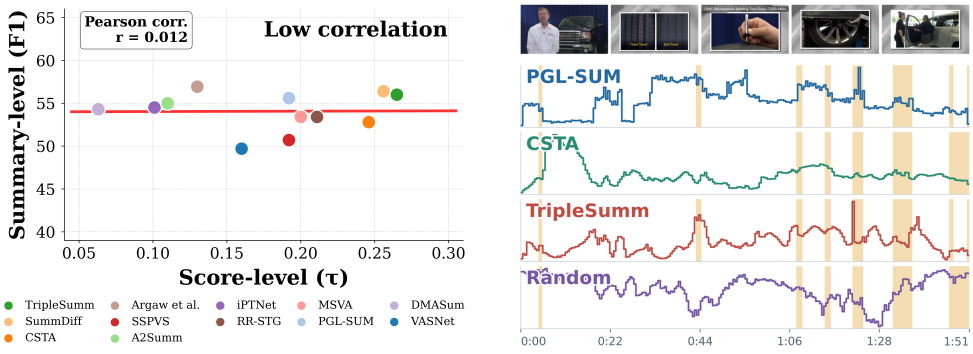
On these pseudo-summary targets, we propose **SumComposer** to directly learn the summary composition process (Fig. 1(c)). From the frame features of a video, it iteratively selects a segment to be included to the summary, conditioned on the preceding selections. This iterative approach is essential to consider the dynamic value of each segment, substantially depending on what other content has been included in the summary. Also, our SumComposer allows any existing video summarization model as its backbone temporal encoder, potentially leveraging temporal dynamics learned to estimate the frame importance. At inference, our model produces a summary from the input video only, without importance annotations, external VLM or LLM calls, or a fixed composition pipeline. The aforementioned pseudo-summary construction is used only to prepare training data.

Through extensive experiments, we verify both the quality of our pseudo-summary construction and the effectiveness of SumComposer for summary composition. Our contributions are as follows:

- We conduct an **empirical analysis** revealing that improvements in frame-importance prediction do not reliably translate into better final summaries under a fixed composition pipeline, motivating the need to learn summary composition itself.
- We propose a **scalable pseudo-summary construction** pipeline that transforms existing importance annotations and video content into structured summary-level supervision.
- We introduce **SumComposer**, which learns to compose summary segments directly from video features by conditioning each decision on preceding selections.

2 Background and Motivation

Frame-level importance signals can be readily obtained at scale from viewer replay statistics (Sul et al., 2023), but they provide per-frame relevance rather than supervision for how frames or segments should be *jointly* selected, accounting for redundancy and a duration budget. In contrast, manually composing complete summaries requires substantial effort. As a result, human-annotated



(a) Cross-metric correlation

(b) Score–summary mismatch

Figure 2: **Diagnostics of fixed composition.** (a) Near-zero correlation between frame-importance ranking and summary F1 on SumMe ($r = 0.012$). (b) Fixed composition collapses different frame-importance scores into nearly identical summary segments (TVSum video 5).

video-summarization benchmarks remain modest in scale, with 25, 50, and 72 videos in SumMe (Gygli et al., 2014), TVSum (Song et al., 2015), and LVSum (Patel et al., 2026), respectively. Given the limited summary-level supervision, the community has trained only a frame scorer and delegate summary composition to a fixed procedure like KTS segmentation (Potapov et al., 2014) and budgeted knapsack selection. Thus, segmentation and selection remain outside the training objective.

However, accurate frame-level importance does not automatically translate into a high-quality summary, because frame-level supervision does not directly supervise the structured decisions required for composition, such as selection dependencies, redundancy, and temporal boundaries. Thus, optimizing only for frame-importance prediction while relying on a fixed composition pipeline could be suboptimal. For example, KTS fixes candidate boundaries based on visual changes, preventing the selector from adjusting them. Moreover, knapsack-based selection typically assigns each candidate segment a utility based on its average frame score, which remains fixed throughout the selection process regardless of redundancy with already selected content. These limitations motivate bringing temporal boundary placement and selection decisions into the learned summarization model.

Early video summarization work commonly evaluated the overlap between predicted and reference summaries using F1. Otani et al. (2019) showed that F1 can be strongly affected by the fixed segmentation and selection steps, and therefore advocated ranking-based metrics such as Kendall's τ (Kendall, 1945) and Spearman's ρ (Zwillinger & Kokoska, 1999) to better isolate the quality of frame-importance prediction. While this shift was useful for evaluating scoring models, it also moved attention away from the summary composition stage itself.

We therefore examine whether improvements in frame-importance prediction actually translate into better final summaries. To investigate this, we probe the relationship between the former measured in Kendall's τ and the latter in summary-overlap F1. Fig. 2(a) exhibits near-zero correlation across existing metrics, with Pearson correlations of 0.012 on SumMe and 0.020 on TVSum. The corresponding TVSum analysis in App. A.1 also leads to a similar conclusion. Fig. 2(b) further demonstrates that the fixed composition collapses markedly different frame-importance scores from various models into nearly identical summaries. (See App. A.2 for additional examples.) This occurs because the predicted scores are subsequently constrained by the same segmentation and budgeted selection procedure, which can strongly shape the final summary. Together, these diagnostics suggest that a superior frame-importance prediction does not consistently translate into a better final summary when the composition step remains fixed.

These observations naturally motivate to train summary composition beyond frame scoring, which requires suitable structured supervision. In fact, a few prior works have explored towards this direction, but leave some limitations. HSA-RNN (Zhao et al., 2018) models temporal segments hierarchically, but maps segment importance back to frames for training with frame-level supervision. seqDPP (Gong et al., 2014) captures dependencies by sequential selections, but operates on subset selection rather than explicit segment composition. DSNet (Zhu et al., 2021) learns temporal proposals from segment-level targets, but constructs these targets using fixed KTS segmentation and knapsack selection. LfVS (Argaw et al., 2024) constructs sequential summary targets from LLM-

selected timestamped transcripts, but requires speech-aligned transcripts. V2Xum-LLM (Hua et al., 2025) decodes summaries with an LLM, beyond frame-level importance regression, but represents the output as discrete temporal indices rather than intervals with explicit boundaries. Taken together, these approaches either learn only part of the composition process or obtain structured supervision through fixed construction rules or additional modalities.

In contrast, we ask if existing frame-importance annotations can be repurposed, together with video content, to directly supervise summary composition. For this, we construct budget-aware pseudo-summaries from frame importance and visual content, and use them to train a model to directly compose summary segments conditioned on preceding selections. This requires neither speech-aligned transcripts for supervision construction nor a fixed composition pipeline at inference.

3 PSEUDO-SUMMARY CONSTRUCTION

We first describe how we construct pseudo-summary labels from existing frame-level importance annotations and video content. All steps in this section are performed offline and used only for data construction. The overall pseudo-summary construction pipeline is illustrated in App. B.1.

Notation and Preliminaries. Given a video V with T sampled frames, let $\mathbf{f} = \{f_1, \ldots, f_T\} \in \mathbb{R}^T$ denote the annotated frame importance scores. We represent a summary as a chronologically ordered sequence of K non-overlapping segments, $S = \langle s_1, \ldots, s_K \rangle$, where K denotes the number of segments. Each segment is defined as $s_k = [a_k, b_k]$, where a_k and b_k denote the start and end positions of the segment s_k, satisfying $1 \leq a_k \leq b_k \leq T$. We denote the length of a segment by $\ell(s_k) = b_k - a_k + 1$. For a budget ratio $\beta \in (0, 1)$, the selected segments should satisfy $\sum_{k=1}^{K} \ell(s_k) \leq \beta T$. Our pseudo-summary construction maps $(V, \mathbf{f})$ to a target segment sequence $S^\star$, providing summary-level supervision for segment selection and temporal boundaries.

Importance-based Video Segmentation. In order to avoid excessively short segments beyond human perception, we first partition the video into disjoint candidate segments using the frame-importance signal $\mathbf{f}$. Specifically, we smooth the frame-importance signal and compute its absolute temporal gradient. We use local gradient peaks exceeding 10% of the maximum absolute gradient within each video as segment boundaries. Together with the video endpoints, these boundaries define a set of disjoint candidate segments, $\widetilde{\mathcal{S}} = \{\tilde{s}_1, \ldots, \tilde{s}_N\}$, where N is the number of candidate segments. Each candidate is denoted by $\tilde{s}_n$, with $n \in \{1, \ldots, N\}$.

Mean Importance and Description. For each candidate $\tilde{s}_n$, we compute its mean importance $I_n = \left(\sum_{t \in \tilde{s}_n} f_t\right) / \ell(\tilde{s}_n)$, where $\ell(\tilde{s}_n)$ is the length of the candidate segment. Also, we describe its visual content by prompting a vision-language model (VLM) with "Describe this video.", producing a textual description c_n. Putting them together, we complete a candidate record $r_n = (\tilde{s}_n, \ell(\tilde{s}_n), I_n, c_n)$, containing a segment, duration, mean importance, and visual description.

Budgeted Segment Selection. Once we have all candidate records $\{r_1, \ldots, r_N\}$ for a video, we perform budgeted selection with a large language model (LLM) by instructing it to use importance scores as relevance cues, compare visual descriptions to reduce redundancy and improve content coverage, and satisfy a duration budget β. Based on these criteria, the LLM ranks and selects a subset of candidate segments. We then programmatically verify the selected subset, including the validity of the returned candidates and the total duration constraint. Construction details, including the prompt, and examples are provided in App. B and App. F, respectively.

Constructing Pseudo-Summary. We chronologically order the selected candidate segments and merge consecutive ones. This yields a pseudo-summary $S^\star = \langle s_1^\star, \ldots, s_{K^\star}^\star \rangle$, which provides summary-level supervision for training summary composition, where $K^\star$ denotes the number of participating segments.

4 SUMCOMPOSER: LEARNING SUMMARY COMPOSITION

In this section, we introduce SumComposer, which learns from the pseudo-summaries constructed in Sec. 3 to directly compose summary segments from video features, as illustrated in Fig. 3. At inference, it requires neither frame-importance annotations nor external VLMs or LLMs, and performs learned summary composition without relying on the fixed composition pipeline.

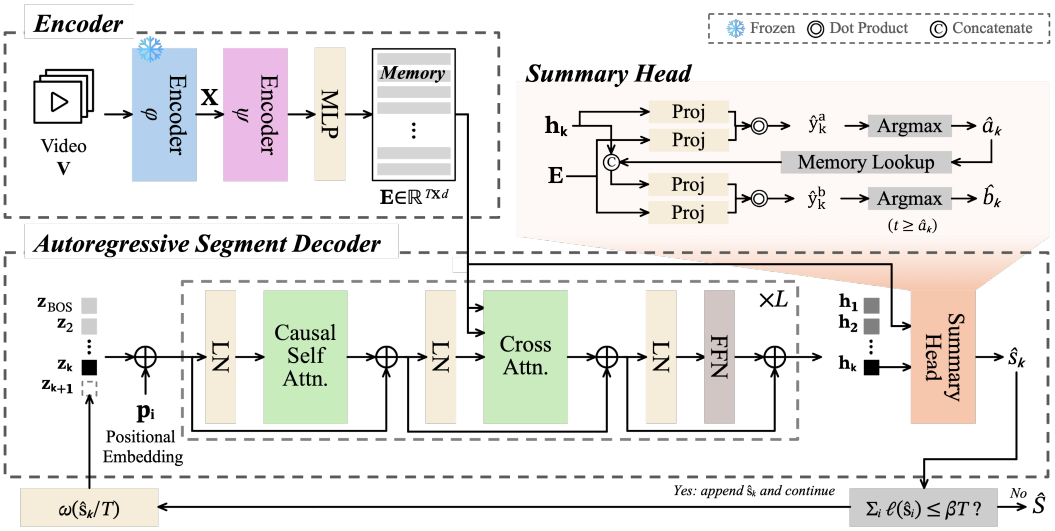


Figure 3: **Overview of SumComposer.** Given a video, the encoder constructs video memory, and the decoder autoregressively composes a summary by learning segment selection and temporal boundaries conditioned on preceding selections.

4.1 Temporal Encoder and Memory Construction

Given a video V, a frozen pretrained feature extractor ϕ produces frame-level features $\mathbf{X} = \phi(V) \in \mathbb{R}^{T\times D}$, which are further contextualized by a temporal encoder ψ, denoted by $\mathbf{E} = \mathrm{MLP}(\psi(\mathbf{X})) \in \mathbb{R}^{T\times d}$. The summary decoder refers to $\mathbf{E}$, which contains temporally contextualized frame representations, as video memory through cross-attention. Importantly, ψ can be instantiated with a range of existing temporal video encoders, including those developed for frame-importance prediction, allowing SumComposer to leverage their temporal modeling capabilities for summary composition.

4.2 Autoregressive Summary Composition

Our SumComposer builds on a standard Transformer decoder (Vaswani et al., 2017), autoregressively predicting one segment at a time so that each decision can depend on the content already included in the summary. To apply it to summary composition, we adapt its input and output spaces as follows: 1) instead of discrete tokens, it takes a sequence of embeddings of previously selected temporal segments as input, 2) cross-attends to the video memory $\mathbf{E}$, and 3) predicts a new segment $s_k = [a_k, b_k]$ to be added to the summary as output.

Segment Embedding Token. We represent each segment included in the summary as a token embedding. At decoding step k, $\mathbf{z}_k$ encodes the segment selected at the previous step through its start and end coordinates, normalized by the video length T:

$$\mathbf{z}_k = \begin{cases} \mathbf{z}_{\text{BOS}}, & \text{if } k = 1, \\ \omega([a_{k-1}/T,\ b_{k-1}/T]), & \text{if } k > 1, \end{cases} \tag{1}$$

where $\mathbf{z}_{\text{BOS}}$ is a learned beginning-of-sequence token and ω is a learnable projector that maps the normalized coordinates to a d-dimensional embedding. Thus, the decoder input at step k is denoted by $[\mathbf{z}_1, ..., \mathbf{z}_k] \equiv [\mathbf{z}_{\text{BOS}}, \omega(s_1), \ldots, \omega(s_{k-1})]$, representing the summary composed so far.

Autoregressive Segment Decoder. Taking the input sequence, the decoder adds a learnable positional embedding $\mathbf{p}_i \in \mathbb{R}^d$ to each segment token $\mathbf{z}_i$ for $i = 1, ..., k$. Then, it applies causal self-attention and cross-attention to the video memory $\mathbf{E}$ for L times, following the standard Transformer decoder. We denote the resulting representations by $\mathbf{h}_i$ for $i = 1, ..., k$, which are expected to contain essential content of each segment, contextualized over the entire video.

Summary Head. Among the k output tokens from the decoder, we take $\mathbf{h}_k$ to predict the next segment. Unlike a vocabulary prediction head in the original Transformer, our summary head maps $\mathbf{h}_k$ to a temporal segment by predicting its start and end boundaries sequentially. For the start position, we linearly project $\mathbf{h}_k$ and each memory vector $\mathbf{E}_t$ and compute their dot-product similarities over

time. For the end position, we additionally condition the query on the memory representation $\mathbf{E}_{\bar{a}_k}$ at the start position and consider only positions $t \geq \bar{a}_k$. We use $\bar{a}_k = a_k^\star$ during training and $\bar{a}_k = \hat{a}_k$ at inference. Let $\hat{\mathbf{y}}_k^a, \hat{\mathbf{y}}_k^b \in \mathbb{R}^T$ denote the corresponding softmax-normalized distributions, with positions $t < \bar{a}_k$ masked out before normalization for the end prediction. Formally,

$$\hat{a}_k = \underset{t=1,\ldots,T}{\arg\max} \underbrace{(\mathbf{W}_q^a \mathbf{h}_k)^\top (\mathbf{W}_m^a \mathbf{E}_t)}_{\hat{y}_{k,t}^a}, \quad \hat{b}_k = \underset{t=\bar{a}_k,\ldots,T}{\arg\max} \underbrace{\left(\mathbf{W}_q^b [\mathbf{h}_k; \mathbf{E}_{\bar{a}_k}]\right)^\top (\mathbf{W}_m^b \mathbf{E}_t)}_{\hat{y}_{k,t}^b}, \tag{2}$$

where $[\cdot;\cdot]$ denotes concatenation and $\mathbf{W}_q^a$, $\mathbf{W}_m^a$, $\mathbf{W}_q^b$, and $\mathbf{W}_m^b$ are learnable linear projections. At inference, the predicted boundaries form the segment $\hat{s}_k = [\hat{a}_k, \hat{b}_k]$, whose normalized coordinates are embedded as the next segment token according to Eq. (1).

4.3 Boundary-Supervised Training and Budgeted Inference

Training. Let the pseudo-summary target be $S^\star = \langle s_1^\star, \ldots, s_K^\star \rangle$, where $s_k^\star = [a_k^\star, b_k^\star]$. We apply teacher forcing by constructing the decoder input at step k from the target preceding segments $s_1^\star, \ldots, s_{k-1}^\star$ according to Eq. (1), while conditioning the end prediction on the target start position $a_k^\star$. Since the pseudo-summary boundaries can be noisy, we smooth the boundary targets using Gaussian kernels rather than exact one-hot labels. For the start boundary, we define $q_{k,t}^a \propto \exp[-(t - a_k^\star)^2/(2\sigma^2)]$ over $t = 1, \ldots, T$. For the end boundary, we apply the same Gaussian smoothing around $b_k^\star$ only over the feasible positions $t = a_k^\star, \ldots, T$, set $q_{k,t}^b = 0$ for $t < a_k^\star$, and renormalize the distribution. Both $\mathbf{q}_k^a$ and $\mathbf{q}_k^b$ are normalized to sum to one. We minimize

$$\mathcal{L} = \sum_{k=1}^{K_{\max}} m_k \left[D_{\mathrm{KL}}(\mathbf{q}_k^a \| \hat{\mathbf{y}}_k^a) + D_{\mathrm{KL}}(\mathbf{q}_k^b \| \hat{\mathbf{y}}_k^b) \right], \tag{3}$$

where $m_k \in \{0, 1\}$ indicates whether step k corresponds to a valid target segment, and $K_{\max}$ is the maximum number of decoding steps.

Budgeted Inference. Starting from $\mathbf{z}_{\mathrm{BOS}}$, the decoder autoregressively grows the summary with predicted segments. At step k, it first predicts the start and subsequently the end position through Eq. (2). The segment $\hat{s}_k = [\hat{a}_k, \hat{b}_k]$ is accepted and decoding continues if $\sum_{i=1}^k \ell(\hat{s}_i) \leq \beta T$; otherwise, decoding terminates. Decoding continues for at most $K_{\max}$ steps, and the accepted segments are concatenated in decoding order to form the final summary.

5 Experiments

We first assess how well our pseudo-summaries align with human annotations, to validate the effectiveness of our pseudo-summaries to train summary composition (Sec. 5.2). We then evaluate SumComposer against fixed composition and a summary-composition baseline (Sec. 5.3). Comparisons across temporal encoders and training scales examine when learned summary composition is effective, and qualitative results illustrate its effects on segment selection and boundary placement.

5.1 Experimental Setup

Dataset. We evaluate on three representative video summarization benchmarks: SumMe (Gygli et al., 2014), TVSum (Song et al., 2015), and Mr. HiSum (Sul et al., 2023). SumMe and TVSum contain 25 and 50 videos with human annotations, respectively, and are used to evaluate how well our constructed pseudo-summaries align with their human annotations. We then apply our proposed pseudo-summary construction to the large-scale Mr. HiSum dataset, which contains 31,892 videos. The resulting pseudo-summaries provide summary-level supervision for training and evaluating SumComposer, allowing us to study learned summary composition at scale.

Evaluation Metrics. While conventional evaluation often relies on frame-level correlation metrics such as Kendall's τ and Spearman's ρ, we introduce two complementary summary-level metrics. *Segment matching* score (S_{match}) measures segment-level correspondence between the predicted segments $\hat{\mathcal{S}}$ and true segments $\mathcal{S}$ by

$$S_{\mathrm{match}}(\theta) = \frac{2|M_\theta|}{|\hat{\mathcal{S}}| + |\mathcal{S}|}, \tag{4}$$

Table 1: **Evaluation of pseudo-summaries.** Our pseudo-summary better aligns with human annotations in both segment selection and boundary placement. (The gain is compared to Fixed.)

Level	Method	SumMe			TVSum		
		$S_{\text{match}} \uparrow$	$B^{\text{start}}_{\text{align}} \uparrow$	$B^{\text{end}}_{\text{align}} \uparrow$	$S_{\text{match}} \uparrow$	$B^{\text{start}}_{\text{align}} \uparrow$	$B^{\text{end}}_{\text{align}} \uparrow$
Video Level	Fixed	44.6	41.0	38.7	12.3	26.3	27.3
	SeqDPP	51.6	36.9	35.2	16.5	30.1	30.3
	Ours	**61.9** (+17.3)	**54.7** (+13.7)	**49.9** (+11.2)	**21.6** (+9.3)	**32.0** (+5.7)	**32.8** (+5.5)
User Level	Fixed	13.9	22.6	21.0	12.3	26.3	27.3
	SeqDPP	14.4	20.4	20.7	16.5	30.1	30.3
	Ours	**27.0** (+13.1)	**31.8** (+9.2)	**31.8** (+10.8)	**21.6** (+9.3)	**32.0** (+5.7)	**32.8** (+5.5)

where $M_\theta \subseteq \hat{\mathcal{S}} \times \mathcal{S}$ is the set of one-to-one matched pairs satisfying $\text{tIoU}(\hat{s}, s) > \theta$, where tIoU is temporal Intersection-over-Union (Rezatofighi et al., 2019). We use $\theta = 0.5$ unless noted otherwise. A higher score indicates stronger performance, $0 \leq S_{\text{match}} \leq 1$.

Boundary alignment score (B_{align}) is designed to measure the accuracy of boundary localization. For predicted and reference boundary sets $\hat{\mathcal{P}}$ and $\mathcal{P}$, separately for starts and ends, we greedily perform nearest-neighbor one-to-one matching within a tolerance σ:

$$B_{\text{align}}(\sigma) = \frac{2|M_\sigma|}{|\hat{\mathcal{P}}| + |\mathcal{P}|}, \tag{5}$$

where M_σ is the resulting pairs satisfying $|\hat{p} - p| \leq \sigma$ with $\sigma = 0.01T$, where T is the video length. Again, higher values indicate better performance, with $0 \leq B_{\text{align}} \leq 1$.

Implementation Details. We instantiate the temporal encoder ψ with the 7 frame-importance models in Tab. 2. The video features are sampled at 2 fps for SumMe and TVSum, while 1 fps for Mr. HiSum. For Mr. HiSum, we use 1,024-D PCA-reduced features extracted by an ImageNet-pretrained Inception-v3 (Szegedy et al., 2016). We set the summary budget ratio to $\beta = 0.15$. Pseudo-summary construction employs Qwen3-VL-8B-Instruct (Bai et al., 2025) as the VLM and gpt-oss-120b (Agarwal et al., 2025) as the LLM. For each encoder, we follow its original architecture and optimization settings, with further implementation details provided in App. C.

5.2 Evaluation of Pseudo-Summary

To evaluate the quality of our pseudo-summary construction, we compare it with the conventional KTS+Knapsack pipeline and SeqDPP (Gong et al., 2014), a learned sequential selection baseline. Reflecting the inherent subjectivity of video summarization, we aggregate scores from multiple annotators in two ways. For video-level evaluation, we first aggregate the annotations for each video into a single ground truth. Following the standard, we take *maximum* over user annotations for SumMe (Gygli et al., 2014) and the *mean* for TVSum (Song et al., 2015). Then, we compute the evaluation metrics by comparing this true and predicted scores. For user-level evaluation, each predicted summary is evaluated against each individual annotation and averaged across users.

As shown in Tab. 1, our pseudo-summary construction consistently outperforms the prevalent KTS+Knapsack (denoted by 'Fixed') and SeqDPP across all metrics at both video and user levels. The gains in S_{match} indicate that jointly considering importance, visual content, redundancy, and the summary budget produces segment selections that better align with human summaries than fixed score-based composition. The consistent improvements in B_{align} further verify that importance-driven candidate boundaries better capture the temporal structure of human summaries than KTS-based segmentation. App. D.1 provides another complementary per-instance win-rate analysis, further confirming that these improvements are consistent across videos and individual annotations. Qualitative comparisons on SumMe and TVSum further show that our pseudo-summaries better cover human-selected regions while producing more coherent temporal summaries than Fixed (see App. D.2 for SumMe and App. D.3 for TVSum).

5.3 Evaluation of SumComposer

Comparison with Fixed Composition. Tab. 2 compares SumComposer with the prevalent fixed composition on top of seven frame importance scoring models, on Mr. HiSum. SumComposer tends

Table 2: **Comparison with fixed composition (KTS + knapsack) on Mr. HiSum.**

Method	S_{match} ↑			$B^{\text{start}}_{\text{align}}$ ↑			$B^{\text{end}}_{\text{align}}$ ↑		
	Fixed	Ours	Δ	Fixed	Ours	Δ	Fixed	Ours	Δ
Ground Truth (Sul et al., 2023)	19.07			24.24			21.58		
VASNet (Fajtl et al., 2018) [ACCV'18]	4.56	1.97	-2.59	13.30	15.31	+2.01	13.45	10.67	-2.78
Joint-VA (Badamdorj et al., 2021) [ICCV'21]	4.42	1.52	-2.90	13.38	14.30	+0.92	13.49	9.48	-4.01
PGL-SUM (Apostolidis et al., 2021b) [ISM'21]	4.85	3.16	-1.69	13.94	17.40	+3.46	13.63	11.11	-2.52
UMT (Liu et al., 2022) [CVPR'22]	5.94	17.63	+11.69	15.63	26.25	+10.62	13.45	18.79	+5.34
CSTA (Son et al., 2024) [CVPR'24]	7.63	17.70	+10.07	19.24	26.90	+7.66	12.40	17.83	+5.43
SummDiff (Kim et al., 2025) [ICCV'25]	5.25	15.87	+10.62	14.47	**30.16**	**+15.69**	13.37	**22.70**	**+9.33**
TripleSumm (Kim et al., 2026) [ICLR'26]	7.66	**21.94**	**+14.28**	16.81	28.55	+11.74	13.33	20.49	+7.16

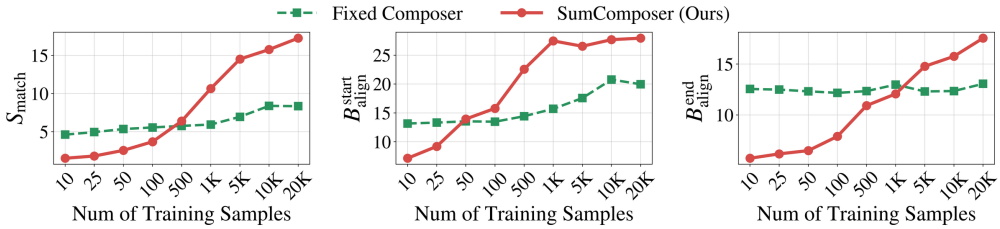


Figure 4: **Scalability analysis.** SumComposer benefits increasingly from larger training sets, while fixed composition largely plateaus with CSTA on Mr. HiSum.

to yield prominent and consistent improvements with recent, stronger temporal encoders (UMT, CSTA, SummDiff, and TripleSumm), while the gains are limited or mixed with earlier encoders (VASNet, Joint-VA, and PGL-SUM). This pattern suggests that our proposed method effectively carries the temporal dynamics to segment selection and boundary prediction, eventually producing superior video summaries. However, its effect can be limited if the visual encoder does not posses sufficient understanding on temporal dynamics. Surprisingly, our highest performance even exceeds that of fixed composition on top of the ground truth (top-most line in Tab. 2), provided by the dataset.

Comparison on Summary Composition. We further compare SumComposer with baselines that directly produce the final summary. To the best of our knowledge, HSA-RNN (Zhao et al., 2018) is still the strongest baseline directly composing the summary, in spite of its age. This is probably because the research community has focused on scoring frame importance during the last few years (see Sec. 2). LfVS (Argaw et al., 2024) is another recent work that directly composes a summary for speech-heavy videos, but this model is not applicable to general videos like Mr. HiSum.

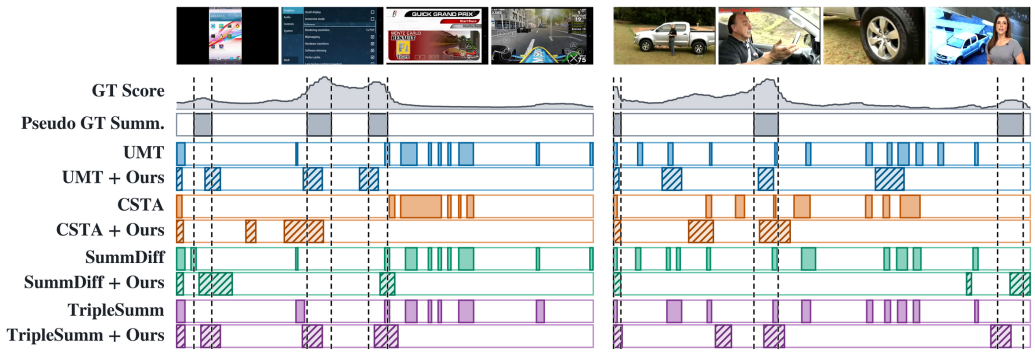
Table 3: **Evaluation on summary composition.**

Method	S_{match} ↑	$B^{\text{start}}_{\text{align}}$ ↑	$B^{\text{end}}_{\text{align}}$ ↑
HSA-RNN	12.03	23.45	7.99
Ours	**21.94**	**28.55**	**20.49**

As shown in Tab. 3, SumComposer (with TripleSumm (Kim et al., 2026)) outperforms HSA-RNN across all three metrics. These results show that accurate importance scores alone do not fully determine summary quality under fixed composition, and that learned summary composition can further improve the resulting summary structure.

Scalability. We evaluate SumComposer on Mr. HiSum with training-set sizes ranging from 10 to 20k videos. As shown in Fig. 4, fixed composition improves only modestly as more training samples are added, whereas SumComposer continues to benefit from increased data scale. Although fixed composition performs slightly better in the smallest-data regimes, SumComposer surpasses it as the training set grows with increasingly larger gains across all three metrics beyond 1K training videos. This scaling behavior highlights the advantage of learned summary composition, which can directly benefit from additional summary-level supervision, unlike a fixed composition pipeline whose segmentation and selection rules remain unchanged.

Qualitative Results. Fig. 5 demonstrates two examples from Mr. HiSum. Under fixed composition, several temporal encoders produce fragmented summaries or select segments poorly aligned with the pseudo-summary, even around highly important regions. In contrast, SumComposer composes more coherent segments that better follow prominent importance peaks and boundaries, while avoiding unnecessary cuts and redundant selections. These examples visually reinforce our central finding that identifying important moments alone is insufficient for producing a well-structured summary. By learning the composition process itself, SumComposer better translates temporal representations into a coherent summary, rather than relying on fixed segmentation and selection.



(a) Video ID: bzbc (b) Video ID: MsSO

Figure 5: **Qualitative comparison on Mr. HiSum.** Hatched bars denote SumComposer, and dashed lines mark pseudo-summary boundaries. Compared with fixed composition, SumComposer produces more coherent intervals that better align with the pseudo-summary reference.

6 Related Work

Video Summarization. Prior work largely learns frame importance using recurrent (Zhang et al., 2016; Zhao et al., 2017; 2018; 2021b; Jung et al., 2019), attention-based (Fajtl et al., 2018; Wang et al., 2020; Apostolidis et al., 2021b; Son et al., 2024), graph-based (Park et al., 2020; Zhao et al., 2021a), generative (Mahasseni et al., 2017; Yuan et al., 2019; Apostolidis et al., 2019; 2021a; Shang et al., 2025; Kim et al., 2025), and multimodal architectures (Badamdorj et al., 2021; Narasimhan et al., 2021; Huang et al., 2021; Liu et al., 2022; Li et al., 2023; He et al., 2023; Guo et al., 2025; Kim et al., 2026). LLM-based methods (Lee et al., 2025; Guo et al., 2026; Hua et al., 2025) leverage multimodal reasoning and instruction tuning. Importance-based ones commonly convert predicted scores into summaries through KTS (Potapov et al., 2014) and knapsack (Otani et al., 2019). Some works explore aspects of composition: hierarchical structure (Zhao et al., 2018), sequential subset selection (Gong et al., 2014), temporal proposals (Zhu et al., 2021), and autoregressive moment retrieval (Argaw et al., 2024). We discuss remaining limitations of these composition methods in Sec. 2. Our work instead aims summary composition learned from importance-derived summary-level supervision without a fixed composition pipeline or language-model calls at inference.

Datasets for Video Summarization. Existing supervision includes manually composed summaries in SumMe (Gygli et al., 2014) and LVSum (Patel et al., 2026), paired textual and keyframe or video summaries in BLiSS (He et al., 2023), VideoXum (Lin et al., 2024), and MMSum (Qiu et al., 2024), and importance signals from ratings or viewer replay statistics in TVSum (Song et al., 2015), Mr. HiSum (Sul et al., 2023), and MoSu (Kim et al., 2026). These importance signals do not directly specify complete budgeted summaries. LfVS (Argaw et al., 2024) obtains structured supervision from ASR transcripts, and thus is applicable only to the script-heavy videos. Our method, on the other hand, constructs pseudo-summary without such a restriction, applicable to any general videos.

7 Summary

We revisit video summarization from the perspective of summary composition. Our diagnostics reveal that improvements in frame-importance prediction do not reliably translate into better final summaries when segmentation and selection remain fixed. To automate the costly summary label collection, we introduce pseudo-summary construction, which transforms existing importance annotations and video content into budget-aware summary-level supervision. Our SumComposer model learns to compose a summary directly from video features by conditioning each decision on preceding selections, without requiring importance annotations, LLM calls, or a fixed composition pipeline at inference. Our constructed pseudo-summaries better align with human summaries than fixed composition, and SumComposer improves segment and boundary alignment.

AI USE STATEMENT

Qwen3-VL-8B-Instruct and gpt-oss-120b were used as part of the offline pseudo-summary construction pipeline, as described in Sec. 3. Generative AI tools were also used to assist with manuscript language editing, including grammar, wording, and typographical corrections. All AI-generated or AI-assisted outputs were reviewed and verified by the authors, who take full responsibility for the final content of this work.

REFERENCES

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, et al. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025.

Evlampios Apostolidis, Eleni Adamantidou, Alexandros I Metsai, Vasileios Mezaris, and Ioannis Patras. Unsupervised video summarization via attention-driven adversarial learning. In *Proceedings of the International Conference on Multimedia Modeling (MMM)*, 2019.

Evlampios Apostolidis, Eleni Adamantidou, Alexandros I. Metsai, Vasileios Mezaris, and Ioannis Patras. Ac-sum-gan: Connecting actor-critic and generative adversarial networks for unsupervised video summarization. *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, 31(8):3278–3292, 2021a.

Evlampios Apostolidis, Georgios Balaouras, Vasileios Mezaris, and Ioannis Patras. Combining global and local attention with positional encoding for video summarization. In *Proceedings of the IEEE International Symposium on Multimedia (ISM)*, 2021b.

Dawit Mureja Argaw, Seunghyun Yoon, Fabian Caba Heilbron, Hanieh Deilamsalehy, Trung Bui, Zhaowen Wang, Franck Dernoncourt, and Joon Son Chung. Scaling up video summarization pretraining with large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Taivanbat Badamdorj, Mrigank Rochan, Yang Wang, and Li Cheng. Joint visual and audio learning for video highlight detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Jiri Fajtl, Hajar Sadeghi Sokeh, Vasileios Argyriou, Dorothy Monekosso, and Paolo Remagnino. Summarizing videos with attention. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, 2018.

Boqing Gong, Wei-Lun Chao, Kristen Grauman, and Fei Sha. Diverse sequential subset selection for supervised video summarization. In *Advances in Neural Information Processing Systems (NIPS)*, volume 27, 2014.

Miaotian Guo, Shuguang Dou, Yin Li, Aidong Men, and Dongsheng Jiang. Agentic video summarization via self-reflecting multimodal understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Yaowei Guo, Jiazheng Xing, Xiaojun Hou, Shuo Xin, Juntao Jiang, Demetri Terzopoulos, Chenfanfu Jiang, and Yong Liu. Cfsum: A transformer-based multi-modal video summarization framework with coarse-fine fusion. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2025.

Michael Gygli, Helmut Grabner, Hayko Riemenschneider, and Luc Van Gool. Creating summaries from user videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014.

Bo He, Jun Wang, Jielin Qiu, Trung Bui, Abhinav Shrivastava, and Zhaowen Wang. Align and attend: Multimodal summarization with dual contrastive losses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Hang Hua, Yunlong Tang, Chenliang Xu, and Jiebo Luo. V2xum-llm: Cross-modal video summarization with temporal prompt instruction tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2025.

Jia-Hong Huang, Luka Murn, Marta Mrak, and Marcel Worring. Gpt2mvs: Generative pre-trained transformer-2 for multi-modal video summarization. In *Proceedings of the ACM International Conference on Multimedia Retrieval (ICMR)*, 2021.

Yunjae Jung, Donghyeon Cho, Dahun Kim, Sanghyun Woo, and In So Kweon. Discriminative feature learning for unsupervised video summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2019.

Maurice G Kendall. The treatment of ties in ranking problems. *Biometrika*, 33(3):239–251, 1945.

Kwanseok Kim, Jaehoon Hahm, Sumin Kim, Jinhwan Sul, Byunghak Kim, and Joonseok Lee. Summdiff: Generative modeling of video summarization with diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

Sumin Kim, Hyemin Jeong, Mingu Kang, Yejin Kim, Yoori Oh, and Joonseok Lee. Triplesumm: Adaptive triple-modality fusion for video summarization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2026.

Min Jung Lee, Dayoung Gong, and Minsu Cho. Video summarization with large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

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

Jingyang Lin, Hang Hua, Ming Chen, Yikang Li, Jenhao Hsiao, Chiuman Ho, and Jiebo Luo. Videoxum: Cross-modal visual and textural summarization of videos. *IEEE Transactions on Multimedia (TMM)*, 26:5548–5560, 2024.

Ye Liu, Siyuan Li, Yang Wu, Chang-Wen Chen, Ying Shan, and Xiaohu Qie. Umt: Unified multi-modal transformers for joint video moment retrieval and highlight detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Behrooz Mahasseni, Michael Lam, and Sinisa Todorovic. Unsupervised video summarization with adversarial lstm networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Medhini Narasimhan, Anna Rohrbach, and Trevor Darrell. Clip-it! language-guided video summarization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Mayu Otani, Yuta Nakashima, Esa Rahtu, and Janne Heikkila. Rethinking the evaluation of video summaries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Jungin Park, Jiyoung Lee, Ig-Jae Kim, and Kwanghoon Sohn. Sumgraph: Video summarization via recursive graph modeling. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

Alkesh Patel, Melis Ozyildirim, Ying-Chang Cheng, and Ganesh Nagarajan. Lvsum: A benchmark for timestamp-aware long video summarization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2026.

Danila Potapov, Matthijs Douze, Zaid Harchaoui, and Cordelia Schmid. Category-specific video summarization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014.

Jielin Qiu, Jiacheng Zhu, William Han, Aditesh Kumar, Karthik Mittal, Claire Jin, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Ding Zhao, et al. Mmsum: A dataset for multimodal summarization and thumbnail generation of videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Zirui Shang, Yubo Zhu, Hongxi Li, Shuo Yang, and Xinxiao Wu. Video summarization using denoising diffusion probabilistic model. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2025.

Jaewon Son, Jaehun Park, and Kwangsu Kim. Csta: Cnn-based spatiotemporal attention for video summarization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Yale Song, Jordi Vallmitjana, Amanda Stent, and Alejandro Jaimes. Tvsum: Summarizing web videos using titles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

Jinhwan Sul, Jihoon Han, and Joonseok Lee. Mr. hisum: A large-scale dataset for video highlight detection and summarization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

Junyan Wang, Yang Bai, Yang Long, Bingzhang Hu, Zhenhua Chai, Yu Guan, and Xiaolin Wei. Query twice: Dual mixture attention meta learning for video summarization. In *Proceedings of the ACM International Conference on Multimedia (MM)*, 2020.

Li Yuan, Francis EH Tay, Ping Li, Li Zhou, and Jiashi Feng. Cycle-sum: Cycle-consistent adversarial lstm networks for unsupervised video summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2019.

Ke Zhang, Wei-Lun Chao, Fei Sha, and Kristen Grauman. Video summarization with long short-term memory. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

Bin Zhao, Xuelong Li, and Xiaoqiang Lu. Hierarchical recurrent neural network for video summarization. In *Proceedings of the ACM International Conference on Multimedia (MM)*, 2017.

Bin Zhao, Xuelong Li, and Xiaoqiang Lu. Hsa-rnn: Hierarchical structure-adaptive rnn for video summarization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Bin Zhao, Haopeng Li, Xiaoqiang Lu, and Xuelong Li. Reconstructive sequence-graph network for video summarization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 44(5):2793–2801, 2021a.

Bin Zhao, Xuelong Li, and Xiaoqiang Lu. Tth-rnn: Tensor-train hierarchical recurrent neural network for video summarization. *IEEE Transactions on Industrial Electronics (TIE)*, 68(4):3629–3637, 2021b.

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

Daniel Zwillinger and Stephen Kokoska. *CRC standard probability and statistics tables and formulae*. Crc Press, 1999.

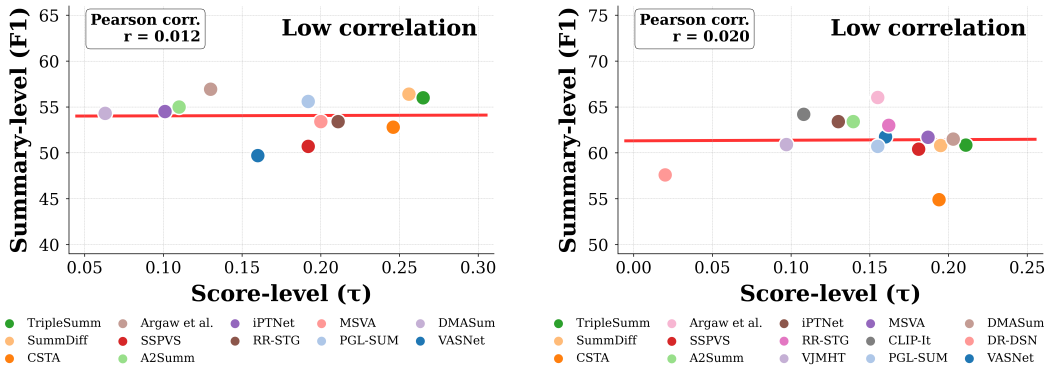
APPENDIX

The appendix is organized as follows. App. A provides additional diagnostics of fixed composition, App. B details pseudo-summary construction, and App. C provides implementation details. App. D presents additional evaluation of pseudo-summary construction, App. E provides additional evaluation of SumComposer, and App. F provides concrete construction examples.

A ADDITIONAL DIAGNOSTICS OF FIXED COMPOSITION

The main manuscript motivates learned summary composition through two complementary diagnostics of the conventional fixed composition pipeline. First, we examine whether improvements in frame-importance prediction are associated with improvements in the resulting summaries across methods. Second, we examine how different frame-importance score sequences are transformed after passing through the same fixed segmentation and budgeted selection procedure. Here, we provide additional results for both analyses.

A.1 CROSS-METHOD CORRELATION ON SUMME AND TVSUM



(a) SumMe ($r = 0.012$)

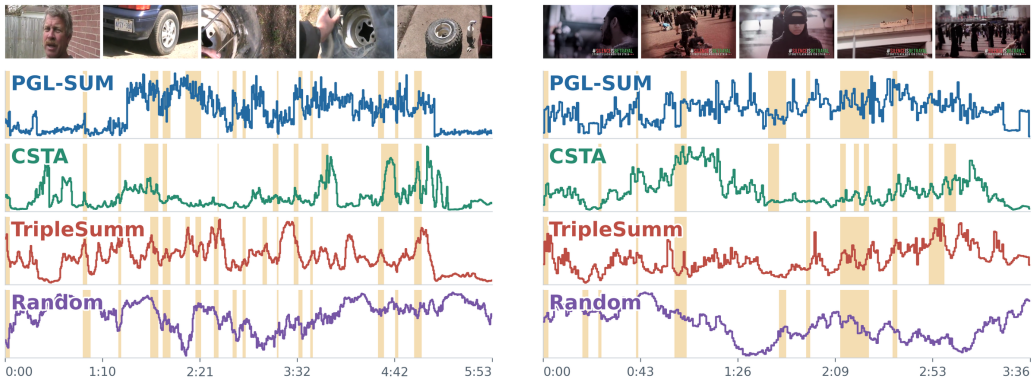
(b) TVSum ($r = 0.020$)

Figure I: **Cross-method correlation between frame-importance prediction and final-summary quality.** We compare frame-importance ranking quality, measured by Kendall's τ, with final-summary quality, measured by summary-overlap F1, across existing methods. The two quantities exhibit near-zero Pearson correlations on both SumMe ($r = 0.012$) and TVSum ($r = 0.020$).

Fig. I presents the cross-method analysis side by side on SumMe and TVSum. For each method, the horizontal axis measures the quality of frame-importance prediction using Kendall's τ, while the vertical axis measures the quality of the final summary using summary-overlap F1. Despite noticeable differences in frame-importance ranking performance across methods, neither dataset shows a meaningful linear association with final-summary quality, with Pearson correlations of 0.012 on SumMe and 0.020 on TVSum.

The consistent behavior across the two benchmarks supports the observation that improving frame-importance prediction alone does not reliably improve the final summary when segmentation and budgeted selection remain fixed. Frame scoring and summary composition therefore capture distinct aspects of the summarization problem, motivating the latter to be modeled explicitly rather than treated solely as fixed post-processing.

A.2 Score–Summary Mismatch Examples



(a) TVSum: Video 1 (b) TVSum: Video 31

Figure II: **Additional examples of score–summary mismatch under fixed composition.** Frame-importance scores from PGL-SUM, CSTA, TripleSumm, and a random score sequence differ substantially, whereas applying the same fixed composition pipeline produces highly similar summary segments.

Fig. II provides two additional examples illustrating how markedly different frame-importance predictions can result in similar final summaries under fixed composition. For each video, the frame-importance score sequences produced by PGL-SUM, CSTA, and TripleSumm exhibit substantially different temporal patterns. We additionally include a random score sequence to provide an extreme comparison.

Despite these differences, the resulting summary intervals are highly similar across score sequences. This behavior arises because every score sequence is passed through the same KTS segmentation and budgeted knapsack selection procedure. KTS restricts all methods to a common set of candidate segment boundaries, while knapsack selects among these candidates according to utilities aggregated from the frame-level scores under the same summary budget. Consequently, substantial differences in frame-level predictions can be attenuated during composition and lead to similar selected intervals.

Together, the cross-method correlation analysis and these instance-level examples highlight two complementary effects of fixed composition. Across methods, better frame-importance ranking does not reliably correspond to better final summaries; within individual videos, substantially different score sequences can be mapped to similar summary structures. These observations motivate learning the composition decisions themselves, including segment selection and temporal boundary placement.

B Details of Pseudo-Summary Construction

This section provides additional details of the pseudo-summary construction pipeline introduced in Sec. 3. The entire pipeline is performed offline and converts existing frame-importance annotations and video content into structured, budget-aware summary targets. We first describe the complete construction procedure, and then provide details of the vision-language model (VLM) used for segment description and the large language model (LLM) used for budgeted segment selection.

B.1 Pipeline Overview

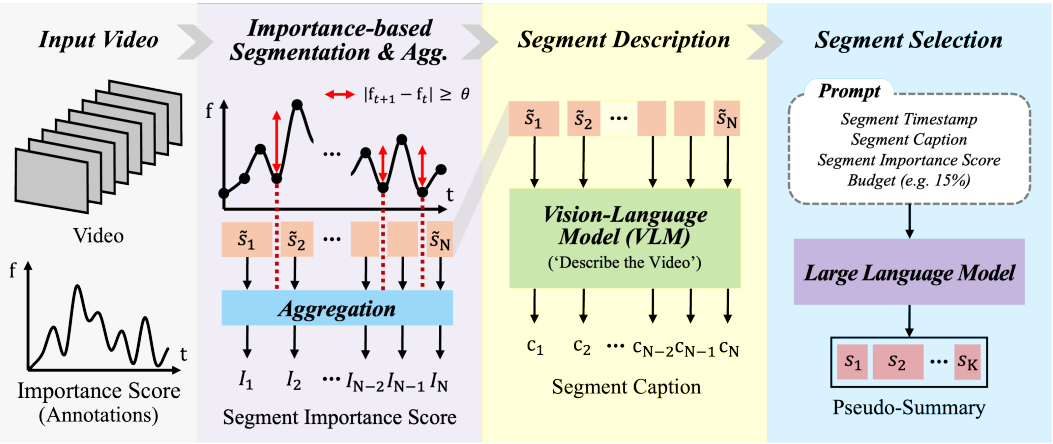


Figure III: **Pseudo-summary construction pipeline.**

Fig. III illustrates the complete offline construction pipeline. Given a video V represented by T sampled frames and its frame-importance annotations $\mathbf{f} = \{f_1, \ldots, f_T\}$, we first partition the video into a set of disjoint candidate segments $\tilde{\mathcal{S}} = \{\tilde{s}_1, \ldots, \tilde{s}_N\}$. As described in the main text, we smooth the frame-importance signal and compute its absolute temporal gradient. Local gradient peaks exceeding 10% of the maximum absolute gradient within the video are used as segment boundaries, together with the two video endpoints. This importance-based segmentation allows the candidate boundaries to reflect temporal changes in the annotated importance signal rather than relying on a separate fixed visual segmentation procedure.

For each candidate segment $\tilde{s}_n$, we compute its mean frame importance as

$$I_n = \frac{1}{\ell(\tilde{s}_n)} \sum_{t \in \tilde{s}_n} f_t,$$

where $\ell(\tilde{s}_n)$ denotes its duration in sampled frames. We additionally obtain a visual description c_n of the segment using a VLM, as detailed in App. B.2. Each candidate is then represented by

$$r_n = \big(\tilde{s}_n,\ \ell(\tilde{s}_n),\ I_n,\ c_n\big),$$

which contains its temporal interval, duration, mean importance, and visual description.

The resulting candidate records $\{r_1, \ldots, r_N\}$ are provided to an LLM together with the summary budget. The LLM uses the importance scores as relevance cues and the visual descriptions to compare candidate content, ranking the candidates and retaining a subset that balances relevance, coverage, and redundancy under the duration constraint. We describe this stage in detail in App. B.3.

Finally, the selected candidates are reordered according to their original temporal positions. When multiple selected candidates are consecutive in the original video, they are merged into a single segment. The resulting chronologically ordered sequence forms the pseudo-summary target $\mathcal{S}^\star$ used as summary-level supervision for SumComposer.

B.2 SEGMENT DESCRIPTION

Vision-language model. We employ `Qwen3-VL-8B-Instruct` (Bai et al., 2025) to describe the visual content of each candidate segment. Given sampled frames $\mathcal{F}(\tilde{s}_n)$ from candidate $\tilde{s}_n$, the VLM produces a textual description c_n of the segment. These descriptions provide semantic information that is complementary to the scalar importance score I_n, allowing the subsequent selection stage to compare the content of different candidates when considering coverage and redundancy.

Prompt. We use the same generic instruction for every candidate segment:

VLM Prompt

```
Describe this video.
```

We intentionally use no additional summarization-specific instruction at this stage. The VLM is used only to describe the visual content of each candidate. The decision of whether the candidate should be included in the pseudo-summary is deferred to the subsequent budgeted selection stage. The generated description c_n is combined with the candidate interval, duration, and mean importance to form the candidate record r_n.

B.3 BUDGETED SEGMENT SELECTION

Large language model. We employ `gpt-oss-120b` (Agarwal et al., 2025) to rank and select summary segments from the candidate records. For each video, the LLM receives the temporal interval, duration, mean importance score, and visual description of every candidate segment, together with the summary budget ratio β. The LLM is instructed to use the importance scores as relevance cues while comparing the visual descriptions to improve content coverage and reduce redundancy among the selected candidates.

Importantly, the LLM does not generate new temporal segments. It selects only from the candidate segments produced by the preceding importance-based segmentation stage. We further require the returned segments to preserve their original timestamps and captions so that the output can be mapped directly back to the video.

Prompt. The complete instruction used for budgeted segment selection is shown below.

LLM Prompt

```
I am providing you with a list of video segment captions.
Each segment includes a timestamp, its duration, captions and an "Importance Score".

Please generate an extractive summary that is both comprehensive and diverse.

Instructions:

1. COVERAGE & DIVERSITY: Represent the entire narrative arc.
2. IMPORTANCE: Use the provided "Importance Score" as a key reference to identify
critical moments.
3. FAITHFULNESS: Do not paraphrase. Use the exact wording and original timestamps.
4. RANKING: Rank the segments by their contribution to the overall story (Rank 1 is
the most essential).
5. BUDGET: Total duration must be <= [0.15 x Video Duration] seconds. Stop before
exceeding this limit.

Output Format (Strictly follow this):
Rank 1: [Start_Time - End_Time] Original Caption
Rank 2: [Start_Time - End_Time] Original Caption
...
```

The LLM returns candidate segments in ranked order, where the rank reflects each candidate's contribution to the overall video. From this ranked output, we retain the highest-ranked candidates while ensuring that their cumulative duration does not exceed the summary budget βT. We then programmatically verify that every selected segment corresponds to a valid candidate and that the duration constraint is satisfied. Finally, the selected candidates are restored to chronological order, and consecutive candidates are merged to form the final pseudo-summary target $\mathcal{S}^{\star}$.

This construction uses the VLM and LLM only for offline supervision generation. Neither model is required when SumComposer is trained to compose summary segments from video features or when it is used at inference.

C Implementation Details

We provide additional implementation details for the experiments in the main manuscript. Across all experiments, SumComposer is paired with the temporal encoder ψ of the corresponding video summarization method. We preserve the original architecture and training configuration of each temporal encoder, while using the same SumComposer decoder architecture across all backbones.

Video features and temporal sampling. Video features are sampled at 2 fps for SumMe and TVSum and at 1 fps for Mr. HiSum. For Mr. HiSum, we use 1,024-dimensional PCA-reduced features extracted by an ImageNet-pretrained Inception-v3 (Szegedy et al., 2016). We use a summary budget ratio of $\beta = 0.15$ throughout all experiments.

The pseudo-summary targets used for training are constructed offline prior to SumComposer training. Accordingly, the VLM and LLM described in App. B are used only for supervision construction and are not invoked during SumComposer optimization or inference.

SumComposer decoder architecture. The architectural hyperparameters of the SumComposer decoder are summarized in Tab. I. These hyperparameters are shared across all temporal encoders.

Table I: **SumComposer decoder architecture.** Decoder hyperparameters shared across all experimental settings.

Hyperparameter	Value
Decoder dimension	256
Decoder layers	4
Attention heads	8
Dropout ratio	0.1

The SumComposer decoder consists of four Transformer decoder layers with eight attention heads per layer. The decoder dimension is set to 256, with a dropout ratio of 0.1. We keep this decoder configuration fixed across all backbone choices so that the effect of the temporal encoder can be studied under a common composition architecture.

Training configuration. For each temporal encoder ψ, we follow the training configuration of its original implementation. This includes the encoder-specific optimization settings and other training hyperparameters used by the corresponding prior method. SumComposer is trained together with the temporal encoder under this backbone-specific training protocol, while the decoder architecture in Tab. I remains unchanged.

This setup preserves the established training recipe of each temporal encoder and allows the comparison between fixed composition and SumComposer to focus on the difference in how the final summary is composed, rather than on changes to the underlying backbone configuration.

D Additional Evaluation of Pseudo-Summary Construction

The main manuscript evaluates the quality of our pseudo-summary construction by measuring its alignment with human annotations on SumMe and TVSum. Here, we complement the aggregate results with two additional analyses. First, we examine per-instance win rates to determine how broadly the improvements over fixed composition are distributed across videos and users. We then provide qualitative comparisons on SumMe and TVSum to visualize the differences in segment selection and temporal boundary placement.

All results in this section evaluate the constructed pseudo-summaries themselves, rather than the predictions of SumComposer. We use the conventional KTS segmentation followed by budgeted knapsack selection as the fixed composition baseline, denoted by *Fixed*.

D.1 Win-Rate Analysis

The aggregate results in Tab. 1 show that our pseudo-summary construction achieves higher segment and boundary alignment than Fixed on both SumMe and TVSum. To examine whether these gains are broadly observed across individual instances, we additionally compare the two methods on a per-instance basis. For each metric, we count the percentage of instances for which either method obtains a higher score, while recording equal scores as ties.

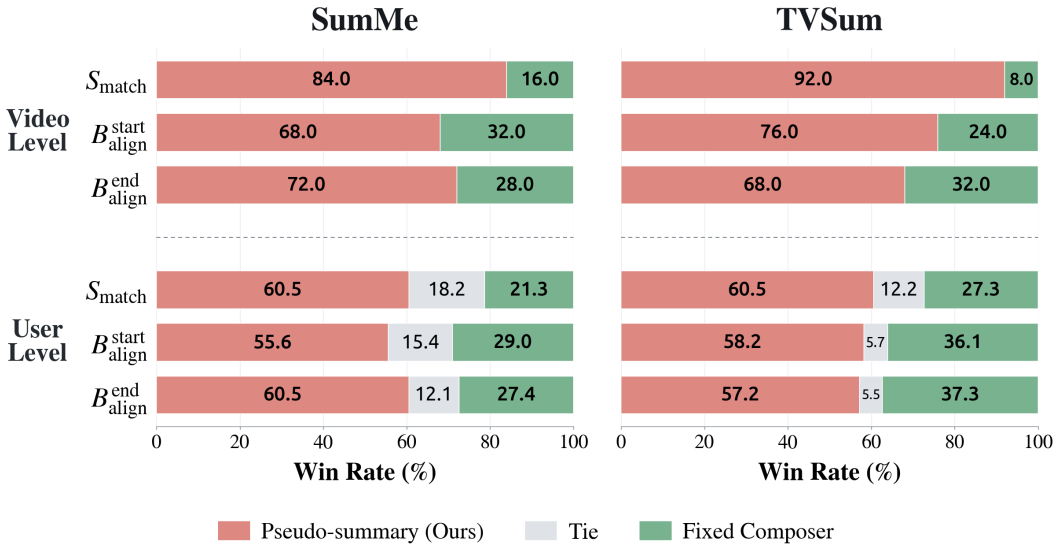
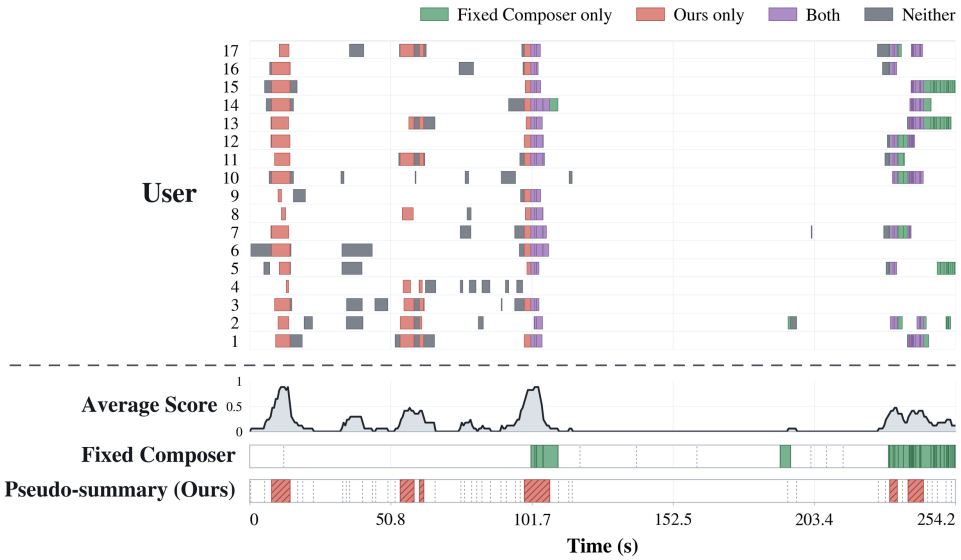


Figure IV: **Win-rate analysis of pseudo-summary construction.** Per-instance comparison between our pseudo-summary construction and fixed composition on SumMe and TVSum. Results are reported for S_{match} and start/end boundary alignment at both video and user levels. Ties indicate instances for which the two methods obtain the same score.

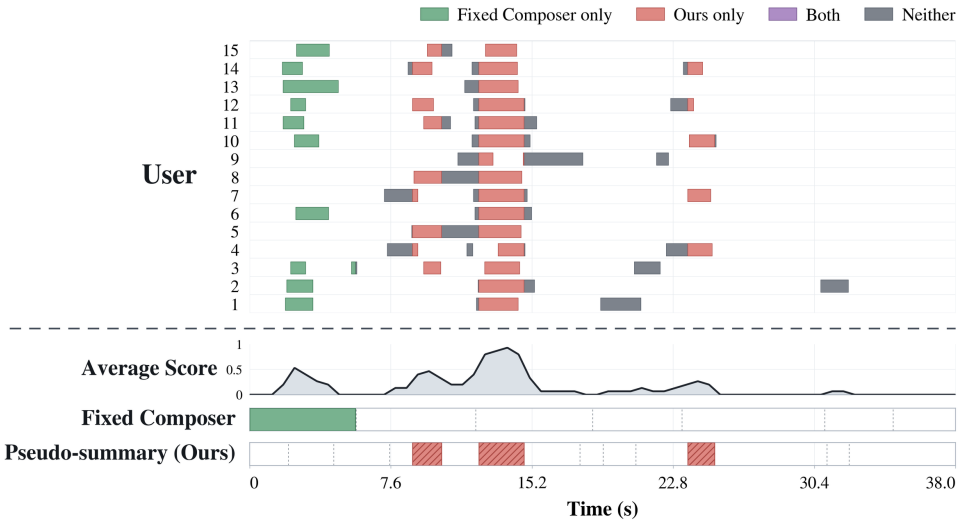
As shown in Fig. IV, our construction achieves a higher S_{match} on 84.0% of SumMe videos and 92.0% of TVSum videos. The same pattern holds for boundary alignment. At the video level, our method outperforms Fixed on 68.0% and 72.0% of SumMe videos for start and end boundaries, respectively, and on 76.0% and 68.0% of TVSum videos.

The improvements remain broadly distributed under user-level evaluation. Across the three metrics, our construction achieves higher scores on 55.6–60.5% of individual SumMe comparisons and 57.2–60.5% of individual TVSum comparisons, whereas Fixed achieves higher scores on 21.3–29.0% and 27.3–37.3%, respectively. Thus, the aggregate improvements reported in the main manuscript are not driven by only a small subset of videos or annotations, but are observed across a broad range of instances.

D.2 Qualitative Results on SumMe



(a) Video 15: "Paintball"

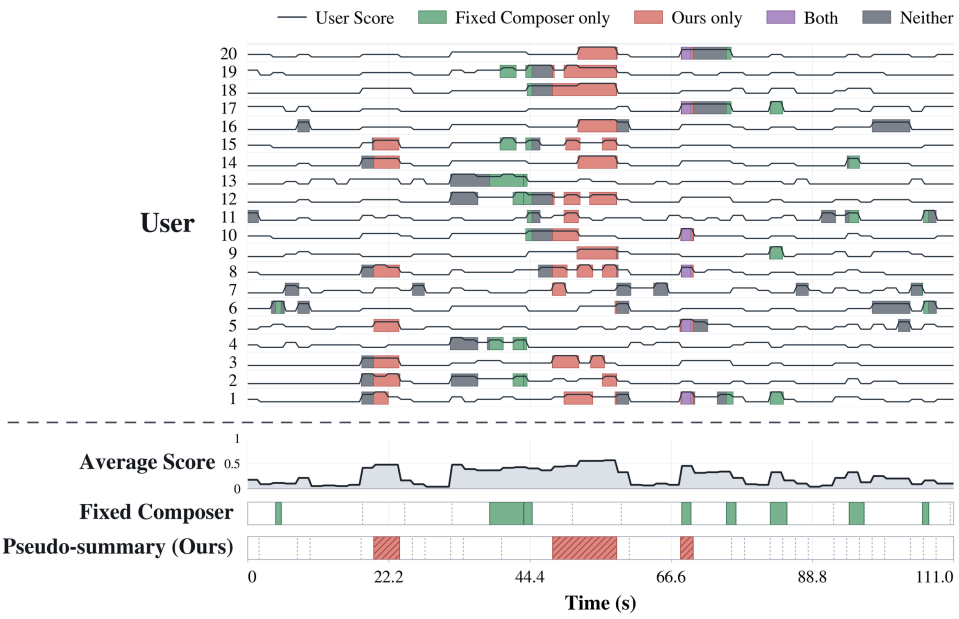


(b) Video 12: "Jumps"

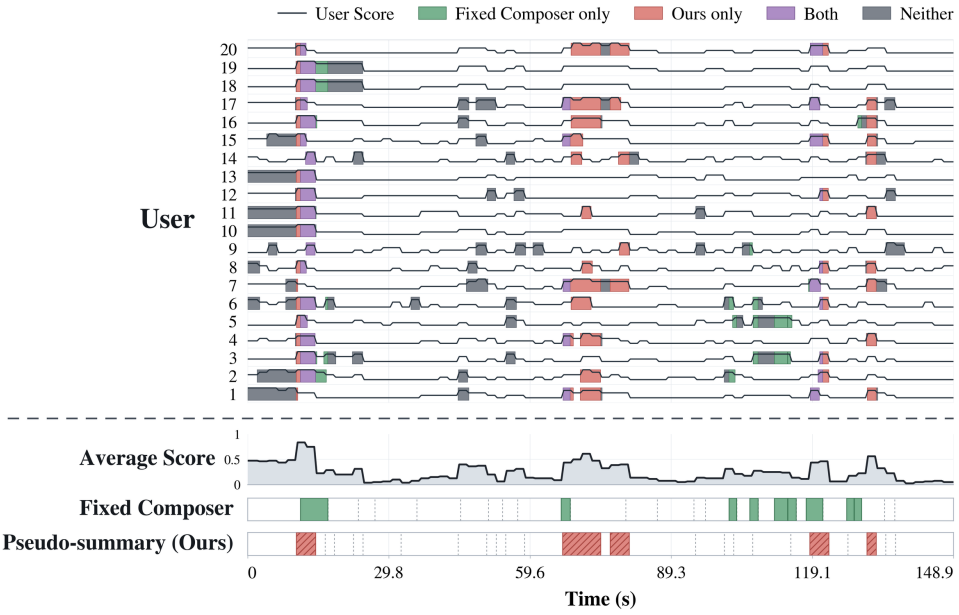
Figure V: **Qualitative comparison on SumMe.** Human-selected regions are categorized by their coverage under Fixed and our pseudo-summary construction.

Fig. V compares Fixed and our pseudo-summary against individual human summaries, with each human-selected region categorized as covered by Fixed only, ours only, both, or neither. In Video 15, our pseudo-summary covers several human-selected regions around the early and central events that are missed by Fixed, whereas Fixed allocates more of its budget to later intervals with relatively limited user agreement. In Video 12, many human selections concentrate around the central events, where our pseudo-summary covers substantially more of the selected regions, while Fixed mainly allocates its budget to an earlier interval. These examples qualitatively illustrate the differences in segment selection underlying the higher human alignment reported in the main manuscript.

D.3 QUALITATIVE RESULTS ON TVSUM



(a) Video 5: "When to Replace Your Tires GMC"



(b) Video 7: "BBC – Train crash 2013"

Figure VI: **Qualitative comparison on TVSum.** Per-user reference regions are categorized by their coverage under Fixed and our pseudo-summary.

Fig. VI compares Fixed and our pseudo-summary against per-user reference summaries. In Video 5, our pseudo-summary concentrates its budget on a coherent central interval covered by many users, whereas Fixed distributes its selections across several temporally separated regions. In Video 7, our pseudo-summary covers several user-selected regions missed by Fixed while avoiding some fragmented selections. Overall, these examples illustrate how incorporating visual content together with importance can improve coverage of human-selected regions and produce a more coherent temporal summary structure.

E CONTROLLED TARGET–COMPOSER ANALYSIS

We examine the interaction between summary targets and composition strategies through a 2×2 comparison.

Experimental setup. All settings use the same TripleSumm temporal encoder architecture, data split, and hyperparameter search budget. (A) and (B) use targets constructed by KTS+Knapsack, while (C) and (D) use our proposed pseudo-summary targets. For (A) and (C), each target is converted into a binary frame-selection mask to supervise a frame-selection model, whose predictions are converted into final summaries using KTS+Knapsack. For (B) and (D), the corresponding segment sequences directly supervise SumComposer to learn summary composition.

All settings are evaluated on the same held-out Mr. HiSum videos using our proposed pseudo-summaries as references. This measures agreement with the constructed reference structure, rather than independently evaluating summary quality. The quality of the pseudo-summary targets themselves is evaluated separately against human annotations in Sec. 5.2.

Table II: **Controlled analysis of summary targets and composition strategies.**

Setting	$S_{\text{match}} \uparrow$	$B^{\text{start}}_{\text{align}} \uparrow$	$B^{\text{end}}_{\text{align}} \uparrow$
(A) KTS target + KTS/Knapsack	7.22	15.77	10.00
(B) KTS target + SumComposer	6.28	15.76	10.99
(C) Proposed target + KTS/Knapsack	11.93	20.62	11.58
(D) Proposed target + SumComposer	**21.94**	**28.55**	**20.49**

Results and interpretation. Replacing KTS-derived targets with our proposed targets improves all three metrics under both composition strategies. The gain in S_{match} is 4.71 points from (A) to (C), compared with 15.66 points from (B) to (D). Thus, changing the targets improves reference alignment even with fixed composition, but yields a larger gain when the segment sequences directly supervise learned composition.

Under the proposed targets, (D) outperforms (C) by 10.01 points in S_{match} and improves both boundary metrics. This comparison changes both the supervision representation and composition procedure, and does not isolate their individual contributions. In contrast, replacing fixed composition with SumComposer under KTS-derived targets does not yield consistent improvements. From (A) to (B), segment matching decreases and start boundary alignment remains nearly unchanged, while end boundary alignment improves. Taken together, these results suggest that the benefit of learned composition depends on the summary targets used for supervision, supporting the complementary design of our pseudo-summary construction and SumComposer.

F PSEUDO-SUMMARY CONSTRUCTION EXAMPLES

We provide three examples illustrating how the candidate segments described in App. B are converted into final pseudo-summary targets. All examples use the budgeted segment selection procedure described in App. B.3. For readability, we show only a subset of the candidate records in each example.

For each example, we show the top-ranked candidates retained under the summary budget and the resulting pseudo-summary target $S^\star$ after chronological reordering and merging consecutive candidates.

F.1 EXAMPLE 1: TEPPANYAKI COOKING

The first example is an 85.7-second video with a maximum summary duration of 12.9 seconds (15% of the video). Several candidate segments depict similar stages of cooking, while later segments contain more distinctive events.

Candidate Records (Excerpt)

```
Total Video Duration: 85.7 seconds
Summary Time Budget (Max 15%): 12.9s

[0.00-6.53] (6.5s, Score: 0.000)
A chef skillfully uses two metal spatulas to toss and flip mushrooms and other
ingredients on a large, hot teppanyaki grill, expertly managing the cooking process in
front of an audience.

[6.53-10.07] (3.5s, Score: 0.059)
A chef skillfully uses two metal spatulas to stir-fry a mixture of ingredients on a
large, hot teppanyaki grill, moving them around the cooking surface with practiced
motions.

[10.07-18.67] (8.6s, Score: 0.007)
A chef in a white uniform expertly cooks a variety of vegetables, including mushrooms,
onions, and asparagus, on a large, hot teppanyaki grill while skillfully using two
spatulas to stir and flip the ingredients.

[18.67-21.67] (3.0s, Score: 0.039)
A chef in a white uniform expertly cooks a variety of ingredients, including sliced
onions, asparagus, and what appears to be minced meat or seafood, on a large, hot
teppanyaki grill while a glass of wine sits nearby.

[21.67-24.67] (3.0s, Score: 0.020)
A chef skillfully cooks a teppanyaki meal on a large, hot griddle, expertly flipping
and arranging ingredients like sliced onions, asparagus, and shrimp while entertaining
guests.

[24.67-28.73] (4.1s, Score: 0.044)
A chef in a white uniform and red hat skillfully cooks a variety of ingredients,
including asparagus and sliced onions, on a large, hot teppanyaki grill while
entertaining an audience.

[28.73-32.80] (4.1s, Score: 0.250)
A chef in a white uniform skillfully uses a spatula to flip and cook sliced onions on a
large, hot teppanyaki grill while asparagus and other ingredients are arranged nearby.

...
```

The following top-ranked candidates are retained under the duration budget.

Selected Candidates

```
Rank 1: [70.00-73.07]
A chef performs a dramatic teppanyaki cooking show, skillfully manipulating a large,
roaring flame on the griddle while preparing food, captivating an audience seated at
tables in a restaurant.

Rank 2: [51.93-54.00]
A chef in a white uniform with red trim is skillfully cooking on a large, hot
teppanyaki grill, pouring sauce over ingredients including asparagus and onions while
steam rises from the sizzling surface.

Rank 3: [32.80-35.73]
A chef skillfully uses a spatula to flip and cook rings of onion on a hot teppanyaki
grill alongside a pile of asparagus spears.

Rank 4: [35.73-39.80]
A chef skillfully uses a spatula to flip and cook a variety of ingredients, including
sliced onions, asparagus, and what appears to be mushrooms or scallops, on a large,
hot teppanyaki grill in front of diners.
```

The selected candidates $[32.80, 35.73]$ and $[35.73, 39.80]$ are temporally consecutive. After chronological reordering and merging these candidates, the final pseudo-summary target is $S^\star = \langle [32.80, 39.80], [51.93, 54.00], [70.00, 73.07] \rangle$.

F.2 Example 2: Jumping into a Natural Pool

The second example is an 85.8-second video with a 12.9-second summary budget. Many candidate segments depict visually similar views of people gathered around a natural pool, while several candidates capture the jumping events.

Candidate Records (Excerpt)

```
Total Video Duration: 85.8 seconds
Summary Time Budget (Max 15%): 12.9s

[0.00-1.00] (1.0s, Score: 0.000)
A serene, lush jungle scene features a stone arch bridge spanning a rocky river with a
small waterfall, where several people are gathered on the rocks and bridge, some
standing and watching while others are in the water below.

[1.00-2.77] (1.8s, Score: 0.111)
A group of people enjoy a serene day at a natural swimming hole, with two men wading
in the water below a small waterfall and three others standing on a large rock beneath
a stone arch bridge, all surrounded by lush green forest.

[2.77-4.77] (2.0s, Score: 0.083)
A group of people enjoy a sunny day at a natural swimming hole beneath a stone arch
bridge, with two men in the water in the foreground and three others standing on a
large rock ledge, one of whom is on the bridge itself.

[4.77-7.27] (2.5s, Score: 0.080)
A group of people enjoy a sunny day in a lush, natural setting, with some standing on a
large rock near a small waterfall and others swimming in the pool below, all under the
arch of an old stone bridge.

[7.27-26.86] (19.6s, Score: 0.002)
A group of people enjoy a sunny day at a natural swimming hole, with some standing on a
large rock beneath an old stone arch bridge while others swim in the water below a
small waterfall.

...
```

The following top-ranked candidates are retained under the duration budget.

Selected Candidates

```
Rank 1: [60.69-64.00]
Under a stone arch bridge, a person leaps from a rocky ledge into a pool below,
creating a splash, while others watch from the water and the rocks.

Rank 2: [41.91-42.94]
Under the arch of an old stone bridge, a group of people enjoy a sunny day by a river,
with two men watching from the water as a woman prepares to jump off a large rock into
the pool below.

Rank 3: [42.94-45.41]
A person leaps from a large rock into a pool below, creating a splash, while others
watch from the water and a stone bridge arches overhead.

Rank 4: [64.00-65.50]
Under the arch of an old stone bridge, a group of people enjoy a natural pool fed by a
small waterfall, with two individuals standing on the rocks above and others swimming
and relaxing in the water below.

Rank 5: [39.94-41.91]
A group of people enjoy a sunny day at a scenic natural pool, with two men standing in
the water in the foreground and others on a large rock beneath a stone arch bridge,
while a small waterfall cascades nearby.

Rank 6: [65.50-66.53]
Under the arch of an old stone bridge, a group of people enjoy a natural pool fed by a
small waterfall, with two individuals standing on a rock ledge above and others
swimming and relaxing in the water below.
```

After chronological reordering, the intervals $[39.94, 41.91]$, $[41.91, 42.94]$, and $[42.94, 45.41]$ form one consecutive sequence. Likewise, $[60.69, 64.00]$, $[64.00, 65.50]$, and $[65.50, 66.53]$ are consecutive. Merging each sequence produces $S^\star = \langle [39.94, 45.41], [60.69, 66.53] \rangle$. This example illustrates how selected candidates can become temporally adjacent after chronological reordering and form a smaller number of segments in the final pseudo-summary target.

F.3 Example 3: Burning a Matchstick Structure

The third example is a 53.7-second video with an 8.1-second summary budget. The candidate importance scores increase substantially around the ignition of the matchstick structure, while later candidates capture another burning event.

Candidate Records (Excerpt)

```
Total Video Duration: 53.7 seconds
Summary Time Budget (Max 15%): 8.1s

[0.00-2.27] (2.3s, Score: 0.000)
A meticulously constructed, tall tower of matchsticks stands upright, its structure
featuring a complex, spiraling pattern where the red-tipped ends of the matches form
diagonal lines against the light-colored wooden bodies, creating a visually intricate
and balanced sculpture.

[2.27-8.27] (6.0s, Score: 0.006)
A delicate, spiraling tower of matchsticks stands precariously upright, its intricate
structure illuminated by a warm, shifting light that casts a soft glow on the white
surface below.

[8.27-9.60] (1.3s, Score: 0.178)
A tall, intricate tower made of numerous matchsticks is carefully lit from the top,
causing a cascading chain reaction as the flames spread downward, igniting the entire
structure in a controlled, mesmerizing display.

[9.60-10.80] (1.2s, Score: 0.567)
A hand holds a lit matchstick, steadily applying its flame to the top of a tall,
spiraling tower constructed from numerous matchsticks, causing the topmost match to
ignite.

[10.80-12.30] (1.5s, Score: 0.800)
```

```
A single lit match is used to ignite a tall, intricate spiral structure made of
numerous unlit matches, causing the entire structure to catch fire and burn brightly
from top to bottom.

[12.30-13.80] (1.5s, Score: 0.422)
A tall, spiral structure made of numerous lit matchsticks burns brightly, casting a
warm, flickering glow as the flames consume the wooden sticks from the top down.

...
```

The following top-ranked candidates are retained under the duration budget.

Selected Candidates

```
Rank 1: [44.37-45.43]
A small pile of matches ignites into a vigorous, dancing flame that grows taller and
brighter, casting a warm glow on the surrounding surface as it consumes the fuel.

Rank 2: [10.80-12.30]
A single lit match is used to ignite a tall, intricate spiral structure made of
numerous unlit matches, causing the entire structure to catch fire and burn brightly
from top to bottom.

Rank 3: [9.60-10.80]
A hand holds a lit matchstick, steadily applying its flame to the top of a tall,
spiraling tower constructed from numerous matchsticks, causing the topmost match to
ignite.

Rank 4: [12.30-13.80]
A tall, spiral structure made of numerous lit matchsticks burns brightly, casting a
warm, flickering glow as the flames consume the wooden sticks from the top down.
```

The three selected candidates spanning 9.60–13.80 seconds become consecutive after chronological reordering and are therefore merged. The final target is $S^\star = \langle [9.60, 13.80], [44.37, 45.43] \rangle$.

Across these examples, the selected candidates are reordered chronologically and merged when consecutive to form the supervision used by SumComposer.