KeyScore: Caption-Grounded Frame Scoring with Spatio-Temporal Clustering for Scalable Video–Language Understanding

Anonymous CVPR submission

Paper ID 22338

Abstract

Selecting a compact yet informative subset of frames is crucial for efficient video understanding, but existing heuristics often overlook semantic grounding and fail to generalize across tasks. We introduce ***KeyScore****, a caption-grounded frame scoring framework that integrates three cues: semantic relevance to captions, temporal distinctiveness, and contextual drop impact. KeyScore assigns importance scores to frames that guide keyframe extractors or multimodal transformers—without any task-specific retraining. We further propose* ***STACFP*** *(Spatio-Temporal Adaptive Clustering for Frame Proposals), which adaptively partitions videos into diverse, non-redundant segments for compact and representative coverage. Together, KeyScore and STACFP achieve up to* ***99% frame reduction*** *over full-frame processing and over* ***70% reduction*** *relative to 8-frame encoders, consistently outperforming them in* ***zero-shot*** *settings across benchmarks for video–language retrieval, keyframe extraction, and action classification. Our approach enables efficient and transferable* ***zero-shot video understanding*** *across diverse domains. This is the first unified caption-grounded and spatio-temporal adaptive framework for zero-shot video understanding.*

1. Introduction

With the exponential growth of video content, video understanding has become a central challenge in multimedia research, powering tasks such as video captioning [1], video-text retrieval [52], and action recognition [54]. A persistent bottleneck across these domains is the need to process long, redundant, and often noisy frame sequences. Such inefficiency not only strains computation but also dilutes semantic signals. Selecting a compact yet informative set of keyframes—those that best capture the core content of a video—offers a promising path toward both efficiency and accuracy. Figure 1 illustrates the goal of our caption-aware frame scoring approach: to highlight semantically relevant and diverse frames while suppressing those that are visually redundant or off-topic with respect to the caption.

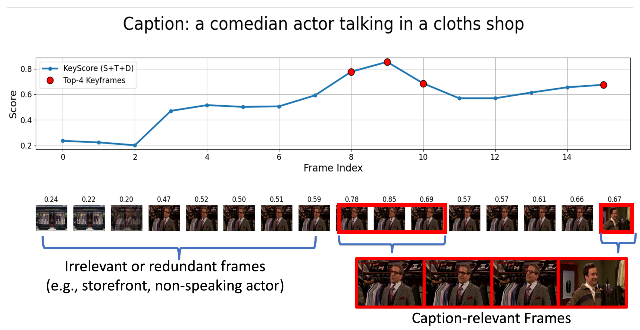

Figure 1. **Motivating example of our frame scoring.** Given the caption *"a comedian actor talking in a cloths shop"*, our method selects keyframes that are semantically aligned with the caption (e.g., actor speaking), while avoiding irrelevant or repetitive frames (e.g., storefront, similar poses).

Despite its importance, **keyframe scoring remains underexplored from a semantic perspective**. Prior methods [12, 27, 41, 42] rely on low-level features, heuristics, or unsupervised clustering, overlooking caption semantics. Uniform sampling, common in video encoders and Video-LLMs, misses key events and repeats redundant frames. Clustering-based approaches such as **SCFP** [23] improve diversity but ignore temporal dynamics, semantic grounding, and require dataset-specific tuning of k. **KeyScore** addresses these gaps by combining caption-grounded scoring with adaptive spatio-temporal clustering, bridging semantics and temporal structure.

To address these limitations at the proposal stage, we introduce **Spatio-Temporal Adaptive Clustering for Frame Proposals (STACFP)**, which augments clustering with temporal encoding and automatically selects the optimal number of clusters via silhouette analysis. Unlike SCFP, STACFP adaptively allocates more proposals to dynamic regions while avoiding redundancy in static segments, producing a compact yet diverse set of candidate frames that better reflect the temporal structure of the video.

CVPR #22338

On top of these proposals, we introduce **KeyScore**, a caption-aware frame scoring method designed to identify the most informative frames in video–language tasks. KeyScore integrates three complementary signals: (1) *semantic similarity* between frames and captions, (2) *temporal representativeness* to ensure coverage of the video timeline, and (3) *contextual drop impact* to account for redundancy and diversity. Together, these signals provide frame-level importance scores that can guide keyframe extraction, improve the efficiency of video encoders, and accelerate inference in Video-LLMs. Unlike prior work that treats semantics and temporal coverage independently, we propose a unified scoring function that harmonizes both axes while being encoder-agnostic and plug-and-play for any Video-LLM.

KeyScore offers two key advantages. First, it provides a flexible framework that can be applied directly to large-scale video–caption datasets, generating frame-level importance scores without requiring manual annotations. Second, it enables new evaluation paradigms where frame quality is judged by **semantic alignment and downstream task performance** rather than heuristics alone.

We extensively validate KeyScore across retrieval (MSR-VTT, MSVD, DiDeMo), keyframe extraction (TVSum20, SumMe), and zero-shot action classification (HMDB-51). Results show that KeyScore consistently outperforms uniform sampling and clustering-based baselines, improving accuracy while reducing frame usage by up to 97–99% compared to raw videos and 63–75% compared to standard 8-frame encoders. These findings demonstrate that caption-aware frame scoring is a powerful tool for content-efficient video understanding. To our knowledge, KeyScore is the first framework to unify caption-grounded semantics, temporal structure, and contextual dependency into a single, training-free frame scoring pipeline.

Our contributions are three-fold:

- We propose **KeyScore**, a caption-aware frame scoring method that integrates semantic relevance, temporal diversity, and drop impact to select keyframes aligned with video captions.
- We introduce **STACFP** (Spatio-Temporal Adaptive Clustering for Frame Proposals), a lightweight yet effective sampling strategy that selects diverse candidate frames while preserving important content.
- We show that KeyScore improves task performance while significantly reducing computational cost—achieving up to 99% frame reduction compared to processing all frames, and outperforming standard sparse sampling strategies (e.g., uniform 8-frame inputs) by focusing on caption-relevant content and filtering out uninformative frames.

2. Related Works

2.1. Keyframe Selection and Video Summarization

Keyframe selection and video summarization aim to extract the most informative or representative frames from a video, thereby reducing redundancy while preserving essential content. Traditional approaches rely on low-level features such as motion, color histograms, or temporal differences to identify representative or diverse frames [16, 55, 57]. Katna [23], for instance, applies K-means clustering on frame histograms and selects the sharpest frame (via Laplacian variance) from each cluster, further filtering based on LUV color differences, brightness, and contrast. While effective, such methods are highly sensitive to feature design and hyperparameter tuning.

Recent learning-based methods have shifted toward supervised or unsupervised frame importance prediction using deep visual features [27, 41–43]. However, these approaches often lack semantic grounding from natural language annotations (e.g., captions), which limits their ability to select frames relevant to higher-level video-language tasks. Attention-based video transformers [6] and reinforcement learning strategies [29] have also been explored, but a consistent limitation is the absence of standardized, semantically informed evaluation criteria—making comparisons across methods less meaningful.

2.2. Frame Sampling and Proposal Methods

Uniform sampling is widely used in Video-LLMs [3, 19, 28, 44, 56] for its simplicity, but often overlooks dynamic moments and yields redundant frames in static regions.

Clustering-based methods such as VSUMM [11] and Katna [23] improve diversity but ignore temporal structure and require predefining the number of clusters. Adaptive variants incorporate silhouette scores [13] or use segmentation-based strategies such as KTS [2]. LMSKE [40] applies per-shot clustering with vision-language features, while TSDPC [42] leverages density peak clustering over temporal segments. Despite their improvements, these methods remain limited by their lack of semantic integration.

In contrast, our **STACFP** sampler performs lightweight global spatio-temporal clustering with automatic k selection, relying on scene transitions rather than caption information. This generates proposals that are temporally diverse and structurally coherent, establishing a strong foundation for subsequent caption-aware scoring and video-language tasks.

2.3. Semantic & Embedding-Aware Frame Scoring

With the rise of vision-language pretraining, frame selection has increasingly leveraged semantic alignment with text. KeyVideoLLM [27] uses CLIP-based text–frame

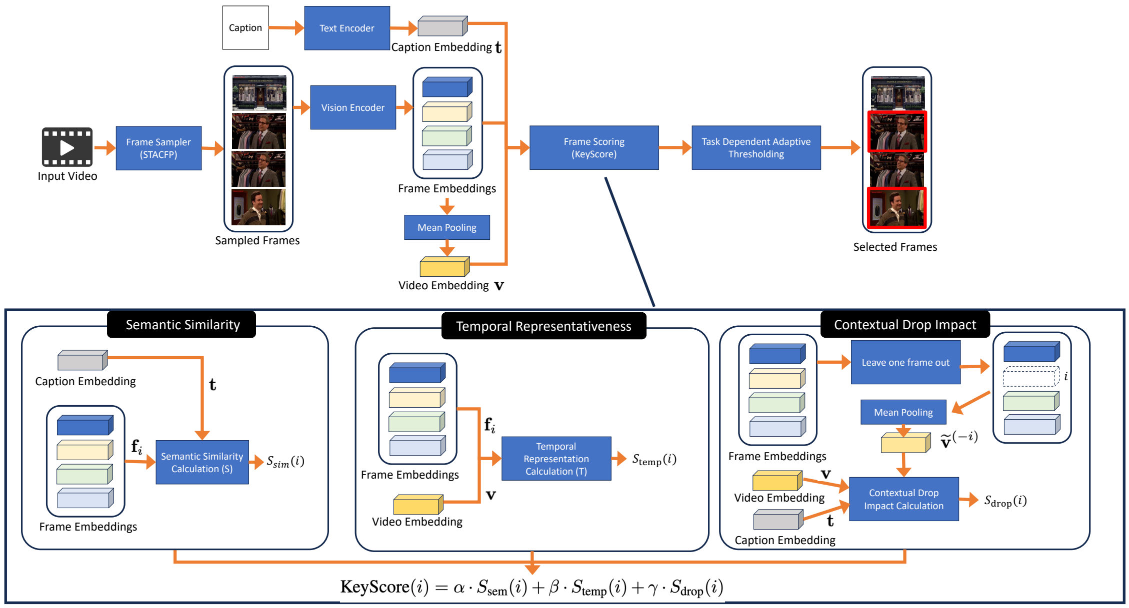

Figure 2. End-to-end pipeline of our proposed approach. STACFP first generates candidate keyframes from the input video. Caption and frame embeddings are then extracted using a text encoder and a vision encoder. The frame scoring module (**KeyScore**) integrates semantic similarity, temporal representation, and contextual drop impact to assign scores to each frame. Finally, task-dependent adaptive thresholding selects the most representative frames for downstream tasks such as retrieval, classification, or summarization.

similarity to achieve high compression while enhancing video QA. AKS [43] formulates keyframe selection as prompt-aware optimization, balancing semantic relevance with temporal coverage. Logic-in-Frames [17] integrates visual–logical dependencies (e.g., causality, spatial relations) to extract semantically rich frames from long videos.

These approaches demonstrate the promise of embedding-aware selection, but most rely on a single criterion—semantic similarity, temporal coverage, or logical reasoning—limiting their ability to generalize across diverse tasks.

Our **KeyScore** addresses this by introducing a hybrid scoring scheme that combines three complementary signals: (1) **semantic similarity**, measuring alignment with caption embeddings; (2) **temporal distinctiveness**, encouraging diverse event coverage over time; and (3) **drop impact**, penalizing redundant or low-utility frames.

This multi-faceted scoring provides a richer assessment of frame importance, yielding more balanced and context-aware selection for downstream retrieval, classification, and summarization tasks.

3. Method Overview

Given a raw video, our method aims to efficiently select a small set of semantically informative and temporally diverse keyframes for downstream video-language tasks. The pipeline consists of two main stages: (1) **STACFP** for frame proposal via spatio-temporal adaptive clustering and (2) **KeyScore** for fine-grained frame scoring based on semantic and structural cues.

As illustrated in Figure 2, a video is first processed by STACFP to generate candidate frames. These frames are then encoded and evaluated by KeyScore, which integrates semantic similarity, temporal contribution, and drop impact to assign importance scores. A task-dependent thresholding step selects the final keyframes used for retrieval, classification, or summarization.

3.1. Spatio-Temporal Adaptive Clustering for Frame Proposal (STACFP)

Long videos contain thousands of redundant or irrelevant frames, making full-frame processing computationally costly and unnecessary. We propose **STACFP**, a lightweight unsupervised method that selects a compact set of visually diverse and temporally distributed frames for downstream scoring or inference.

Unlike uniform sampling or prior clustering-based methods like Katna [23] and VSUMM [11], STACFP encodes both appearance and time in its clustering space. For each sampled frame f_i, we extract a low-level visual feature vector $\mathbf{v}_i$ based on color histograms computed in HSV color space, which is more perceptually aligned than RGB. This histogram is flattened into a vector of fixed dimension d. To encourage temporal dispersion in the clustering process, we also encode the normalized timestamp of each frame $t_i = \frac{i}{N-1}$, where i is the index of the frame among N

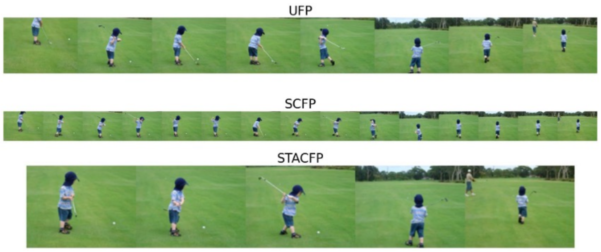

Figure 3. **Qualitative comparison of frame proposal methods.** UFP samples uniformly, leading to redundancy. SCFP enhances visual diversity but overlooks temporal cues, often oversampling static segments. STACFP jointly models spatial and temporal information, capturing representative moments (e.g., the start, peak, and follow-through of a golf swing) with **fewer yet more informative frames**.

total sampled frames. This scalar is then scaled by a hyperparameter γ_{time} and concatenated with the visual feature:

$$\mathbf{x}_i = [v_i;\ \gamma_{\text{time}} \cdot t_i]$$

This results in a $(d+1)$-dimensional feature vector $\mathbf{x}_i$ for each frame. The hyperparameter $\gamma_{\text{time}} \in [3, 15]$ controls the influence of temporal position relative to visual appearance in the clustering process.

We perform k-means clustering over these spatio-temporal features and automatically select the optimal number of clusters k^* via silhouette score maximization [38]:

$$k^* = \arg\max_k \text{Silhouette}(X, \text{KMeans}(k))$$

This adaptive strategy allocates fewer proposals to static scenes and more to dynamic content. The final frame proposals are chosen as the nearest frames to each cluster centroid.

Figure 3 compares UFP, SCFP, and our STACFP. STACFP more effectively captures key temporal transitions and semantically important moments, whereas UFP and SCFP tend to sample redundant or less informative frames.

3.2. Frame Scoring via KeyScore

Given a query caption C and a video $V = \{f_1, f_2, \ldots, f_T\}$ with T frames, our objective is to estimate the importance of each frame f_i in supporting video–caption alignment. We introduce **KeyScore**, a hybrid scoring framework that leverages a pretrained video–text model to embed frames and captions into a shared representation space.

Let $\mathbf{f}_i \in \mathbb{R}^D$ denote the embedding of frame f_i, $\mathbf{t} \in \mathbb{R}^D$ the embedding of caption C, and $\mathbf{v} \in \mathbb{R}^D$ the global video embedding (computed via mean pooling or text-guided attention over $\{\mathbf{f}_i\}$). All embeddings are ℓ_2-normalized.

Overall scoring. KeyScore assigns each frame f_i a weighted score:

$$\text{KeyScore}(i) = \alpha \cdot S_{\text{sem}}(i) + \beta \cdot S_{\text{temp}}(i) + \gamma \cdot S_{\text{drop}}(i) \quad (1)$$

where $\alpha + \beta + \gamma = 1$ and each $S_{\cdot}$ captures a complementary aspect of frame importance.

3.2.1. Semantic Similarity Score (S_{sem})

$$S_{\text{sem}}(i) = \cos(\mathbf{f}_i, \mathbf{t}) \quad (2)$$

S_{sem} measures how well a frame aligns with the caption. **Example:** For "a man riding a horse," frames showing the man on horseback obtain higher scores.

3.2.2. Temporal Representativeness Score (S_{temp})

$$S_{\text{temp}}(i) = \cos(\mathbf{f}_i, \mathbf{v}) \quad (3)$$

S_{temp} captures how representative a frame is of the overall video context, down-weighting outliers. **Example:** In a cooking tutorial, frames of the chef cooking are representative, while a shot of the wall clock is not.

3.2.3. Contextual Drop Impact Score (S_{drop})

$$S_{\text{drop}}(i) = \cos(\mathbf{v}, \mathbf{t}) - \cos(\tilde{\mathbf{v}}^{(-i)}, \mathbf{t}) \quad (4)$$

S_{drop} measures the *marginal contribution* of frame f_i by measuring how much video–text similarity degrades when the frame is removed. A high score indicates that the frame provides indispensable context for aligning the video with the caption, while redundant or uninformative frames yield near-zero impact. **Example:** For "a woman performs a ballet spin," excluding the spin frame sharply reduces alignment, revealing its critical role.

Implementation. All components are min–max normalized before combination. KeyScore can be efficiently computed with vectorized pooling, and returns both raw and weighted scores for downstream selection or ranking.

Figure 4 presents four qualitative examples of KeyScore applied to different video–caption pairs. In the prosthetic setup video (Fig. 4a), KeyScore focuses on frames that visually capture the medical procedure, while down-weighting irrelevant early frames. In the mountain scenes video (Fig. 4b), most frames align with the caption, and KeyScore identifies representative landscape shots without redundancy. The comedian actor example (Fig. 4c) highlights frames where the actor is clearly visible and contextually important, while the Minnie Mouse cartoon example (Fig. 4d) selects frames where the character appears prominently.

Across all cases, semantic similarity (S) and contextual drop impact (D) are the strongest contributors, ensuring semantic and contextual fidelity. Temporal representativeness (T), although less discriminative, provides complementary coverage by selecting recurring frames. Together, these signals enable KeyScore to select just 2–3 frames that faithfully capture the essential visual evidence described by the caption, while discarding redundant or irrelevant content.

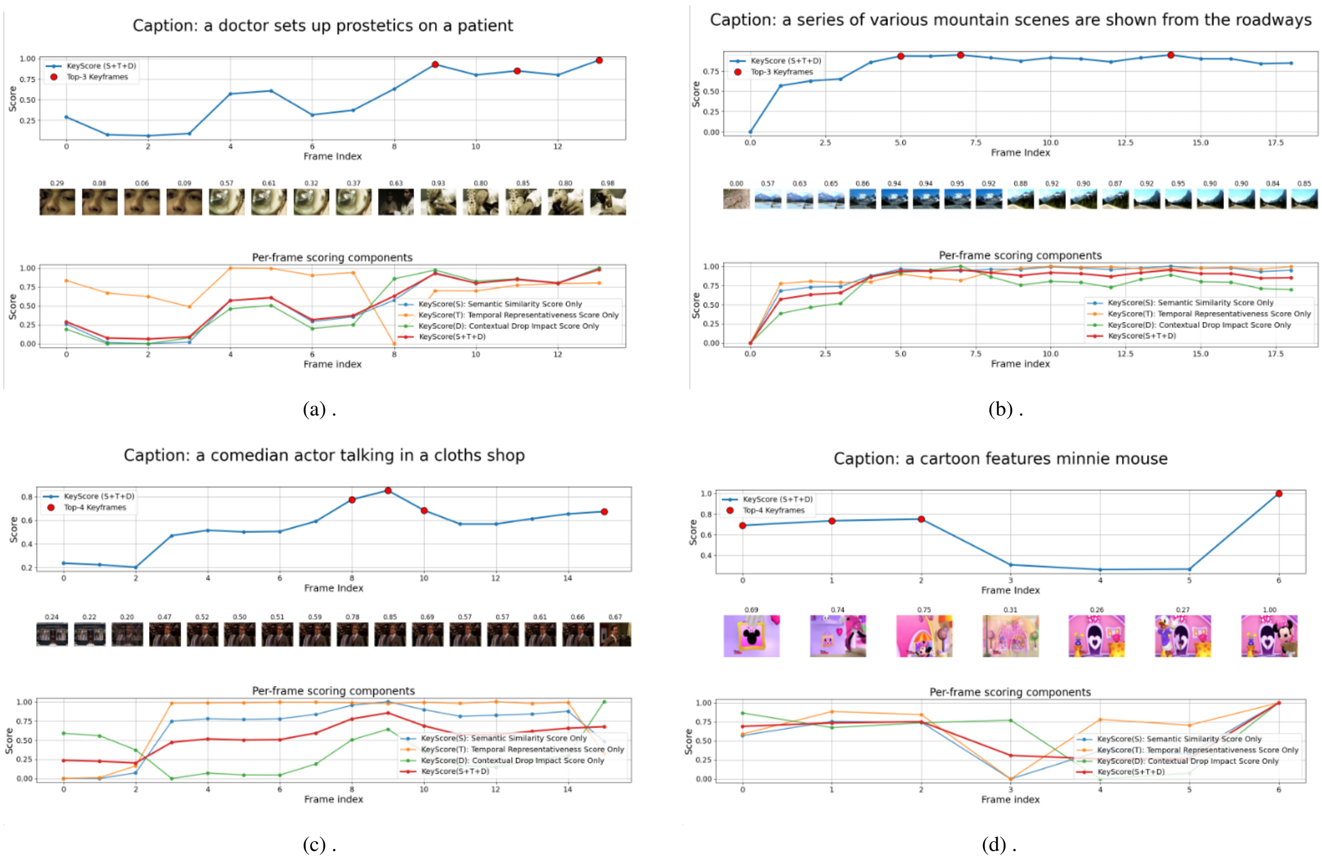

Figure 4. **Qualitative examples of KeyScore across diverse videos.** Each example shows (top) the overall KeyScore curve with top frames, (middle) sampled frames with scores, and (bottom) component contributions. **S** highlights caption-relevant moments, **T** ensures temporal coverage, and **D** preserves contextually critical evidence. Their combination yields compact, semantically grounded, and temporally diverse keyframes.

4. Experiments

We evaluate KeyScore on three representative tasks—video–text retrieval, keyframe extraction, and zero-shot action classification—across multiple public benchmarks.

4.1. Zero-Shot Video-Text Retrieval

We evaluate KeyScore across four aspects: (1) the impact of frame sampling strategies, (2) encoder compatibility, (3) comparison with state-of-the-art models, and (4) frame compression efficiency.

Setup. We follow standard protocols, reporting Recall@K (R@1/5/10) for text-to-video (T2V) and video-to-text (V2T) retrieval.

Backbone. Unless specified, we use the Perception Encoder (PE) [7] as the vision–language backbone. Each video is represented by keyframes from the frame proposal module; when enabled, KeyScore re-ranks and selects the final subset.

Datasets. Experiments are conducted on MSR-VTT [53], MSVD [9], and DiDeMo [4] following standard splits and evaluation protocols.

4.1.1. Frame Proposal Strategies

We evaluate four frame proposal strategies under a controlled retrieval setup:

- **UFP:** Uniform fixed-interval sampling (typically 8 frames); simple and efficient but prone to redundancy and sensitive to frame count.
- **SCFP (Kanta [23]):** K-means clustering in visual space with a fixed number of clusters; reduces redundancy but ignores temporal continuity.
- **SACFP (LMSKE [40]):** Spatial Adaptive Clustering Frame Proposal, equivalent to the LMSKE variant (K-means with silhouette-based cluster estimation). It adaptively determines cluster count based on clustering quality but remains spatial-only.
- **STACFP (ours):** Spatio-Temporal Adaptive Clustering guided by silhouette analysis, jointly modeling spatial and temporal cues for compact, representative frame selection.

Table 1. **Comparison of frame sampling strategies on retrieval performance.** We compare UFP, SCFP (Kanta [23]), SACFP (LMSKE [40]), and our proposed STACFP, each paired with the same encoder [7]. STACFP achieves competitive or superior accuracy with significantly fewer frames, demonstrating efficiency and robustness for video–text retrieval. T2V/V2T: Recall@1 (%); ASF: average sampled frames.

Frame Sampler	MSR-VTT			MSVD		
	T2V	V2T	ASF	T2V	V2T	ASF
UFP	50.0	47.5	8.0	60.4	**82.9**	8.0
SCFP (Kanta [23])	49.4	45.1	16.0	59.9	82.3	10.7
SACFP (LMSKE [40]	49.6	46.3	9.2	60.1	82.4	7.8
STACFP	**49.7**	**48.2**	**6.0**	**60.4**	82.3	**5.6**

As shown in Table 1, all four methods achieve comparable retrieval accuracy on MSR-VTT and MSVD with the same encoder (PE_{core}G [7]). However, STACFP matches or surpasses others with substantially fewer frames—6 and 5.6 per video, versus 8 for UFP, 9.2 for SACFP, and 16 for SCFP—demonstrating superior sampling efficiency. While UFP relies on uniform spacing and SCFP/SACFP perform purely spatial clustering, STACFP adaptively balances spatial diversity and temporal coverage, achieving the best trade-off between accuracy and efficiency for scalable video–language modeling.

Ablation on Timestamp Normalization. STACFP uses normalized timestamps to balance spatial–temporal distances during clustering. Removing normalization ($t_i{=}i$) biases clustering toward later frames, reducing accuracy (T2V 49.7→47.9, V2T 48.2→46.5) and increasing ASF (6.0→8.4). Normalization is thus crucial for stable temporal diversification across videos of varying lengths.

Ablation on Effect of Fixed Cluster Count. We examine how the number of selected clusters ($K \in \{1,3,5\}$) affects retrieval accuracy on MSR-VTT (Table 2). Across all fixed settings and both directions (T2V/V2T), STACFP consistently outperforms UFP, SCFP (Kanta), and SACFP (LMSKE), with the largest gains under tighter budgets ($K{=}1$). As K increases, all methods improve and performance gaps narrow, yet STACFP remains the best performer while staying below the maxima reported in Table 1. This confirms that STACFP's spatio-temporal clustering produces more representative frames even without adaptive K, and its advantage is most pronounced when only a few keyframes are allowed.

4.1.2. KeyScore: Frame Scoring and Selection

Given initial frame proposals from STACFP, we further score each frame using **KeyScore**, a weighted combination of three complementary cues: semantic similarity (S), temporal representativeness (T), and contextual drop impact (D).

Table 3 presents an ablation across MSR-VTT [53], MSVD [9], and DiDeMo [4], comparing individual and joint scoring signals. The PE-only baseline uses 6–11 frames per video and yields modest retrieval performance. Adding KeyScore significantly improves retrieval accuracy while substantially reducing the number of frames.

Table 2. **Fixed cluster count ablation on MSR-VTT (R@1, %).** We fix $K \in \{1,3,5\}$ across all videos and compare UFP, SCFP, SACFP, and STACFP using the same encoder [7]. All fixed-K results remain below each method's main-table maxima.

Method	T2V			V2T		
	$K{=}1$	$K{=}3$	$K{=}5$	$K{=}1$	$K{=}3$	$K{=}5$
UFP	33.5	43.8	46.5	44.1	46.0	47.0
SCFP (Kanta [23])	36.0	44.5	47.0	44.8	45.0	45.1
SACFP (LMSKE [40])	37.2	45.3	48.0	45.5	45.9	46.3
STACFP (ours)	**38.1**	**46.0**	**48.5**	**46.7**	**47.5**	**48.2**

Table 3. **Ablation of KeyScore components.** Text-to-video (T2V) / video-to-text (V2T) R@1 (%) and average selected frames (ASF). S: semantic, T: temporal, D: contextual drop impact.

Method	MSR-VTT			MSVD			DiDeMo		
	T2V	V2T	ASF	T2V	V2T	ASF	T2V	V2T	ASF
PE_{core}G-Video	49.7	48.2	6	60.4	82.3	5.6	45.1	46.1	11.3
+ KeyScore (S)	63.2	60.0	2	88.5	86.5	5	57.8	59.0	3
+ KeyScore (T)	49.8	48.9	8	84.6	86.1	4	48.5	50.1	2
+ KeyScore (D)	62.6	59.4	3	85.8	86.5	3	57.2	58.0	2
+ KeyScore (S+T)	61.3	59.5	3	87.9	88.6	2	59.4	60.3	2
+ KeyScore (D+T)	61.4	59.1	2	87.9	89.2	4	59.7	60.1	2
+ KeyScore (S+D)	63.5	60.3	2	89.1	89.7	2	59.8	60.3	2
+ KeyScore (S+T+D)	**63.9**	**60.5**	**2.5**	**89.2**	**89.2**	**2**	**60.4**	**60.3**	**2**

Among single signals, semantic similarity (S) and contextual drop impact (D) are the most effective, boosting MSR-VTT T2V R@1 above 62 and DiDeMo above 77. Temporal representativeness (T) alone contributes little, but enhances performance when combined with other signals. Pairwise combinations like KeyScore(S+D) already deliver strong gains across datasets.

The best results are obtained with the full combination KeyScore(S+T+D), achieving 63.9/60.5 R@1 on MSR-VTT, 89.2/89.2 on MSVD, and 60.4/60.3 on DiDeMo — all while using only 2–2.5 frames on average. This demonstrates KeyScore's ability to balance semantic, temporal, and contextual factors for compact yet informative frame selection.

4.1.3. Comparison with State of the Art

Integrating KeyScore into the retrieval pipeline substantially boosts performance by filtering redundant frames and retaining the most informative ones, leading to stronger visual–text alignment across encoders and datasets.

Table 4 reports Recall@1 (R@1) for text-to-video (T2V) and video-to-text (V2T) retrieval on MSR-VTT, MSVD, and DiDeMo. Beyond PE_{core}G-Video, KeyScore also im-

Table 4. **Zero-shot video–text retrieval (R@1)** on MSR-VTT, MSVD, and DiDeMo. Results are reported for text-to-video (T2V) and video-to-text (V2T). KeyScore consistently improves both ViCLIP [50] and PE_{core}G-Video [7], demonstrating encoder-agnostic scalability and state-of-the-art results.

Model	MSR-VTT		MSVD		DiDeMo	
	T2V	V2T	T2V	V2T	T2V	V2T
CLIP4Clip [31]	32.0	–	45.2	48.4	–	–
X-CLIP [32]	49.3	48.9	50.4	66.8	47.8	47.8
UMT-L [25]	40.7	37.1	49.0	74.5	49.9	59.7
SigLIP2-L/16 [46]	41.5	31.4	53.7	74.2	18.4	–
InternVL [10]	44.7	40.2	43.4	67.6	–	–
InternVideo2 [51]	51.9	50.9	–	–	57.9	57.1
VideoPrism-g [58]	39.7	71.0	58.1	83.3	–	–
SigLIP2-g-opt [46]	43.1	34.2	55.8	74.6	–	–
PE_{core}G-Image [7]	44.3	35.2	54.3	73.9	–	–
ViCLIP [50]	42.4	41.3	49.1	75.1	31.5	31.5
ViCLIP + KeyScore	**51.3**	**49.8**	**57.9**	**83.4**	**41.2**	**40.9**
PE_{core}G-Video [7]	51.2	49.9	59.7	85.4	43.1	45.1
PE_{core}G-Video + KeyScore	**63.9**	**60.5**	**89.2**	**89.2**	**60.4**	**60.3**

proves **ViCLIP** [50], yielding gains of about $+9\sim10$ R@1 (T2V) and $+8$ (V2T) across benchmarks—demonstrating encoder-agnostic generalization.

ViCLIP + KeyScore achieves 51.3/49.8 (T2V/V2T) on MSR-VTT and 57.9/83.4 on MSVD, while PE_{core}G-Video + KeyScore reaches competitive with recent large models while using only 2–3 frames results: 63.9/60.5 on MSR-VTT, 89.2/89.2 on MSVD, and 60.4/60.3 on DiDeMo. These consistent improvements confirm that KeyScore generalizes across architectures and enhances retrieval robustness without any retraining.

4.1.4. Frame Reduction Analysis

To quantify the efficiency of KeyScore, we measure the proportion of frames it discards relative to standard baselines. We define the *Frame Reduction Rate* (FRR) as:

$$\text{FRR-UFP} = 1 - \frac{N_{\text{sel}}}{N_{\text{UFP}}}, \qquad \text{FRR-Avg} = 1 - \frac{N_{\text{sel}}}{N_{\text{avg}}},$$

where N_{sel} is the number of frames selected by KeyScore, N_{UFP}=8 corresponds to uniform fixed sampling, and N_{avg} denotes the dataset-specific average frame count. A higher FRR indicates greater efficiency (i.e., more frames saved).

Dataset-Level Frame Savings. Table 5 reports the average selected frames (ASF), and frame reduction rates (FRR-UFP and FRR-Avg) across three datasets. On MSR-VTT (avg. 408 frames), KeyScore retains only 2–3 frames (**FRR-UFP = 0.69**, **FRR-Avg = 0.99**), achieving over a 99% reduction relative to the dataset average. On MSVD (avg. 275 frames), similar efficiency is observed (**FRR-UFP = 0.75**, **FRR-Avg = 0.99**), while on DiDeMo, KeyScore reduces 11 sampled frames to just 2–3 (**FRR-UFP = 0.63–0.75**, **FRR-Avg = 0.99**). These results confirm KeyScore's consistent ability to maintain high retrieval accuracy, even under extreme frame reduction.

Discussion. Across datasets, KeyScore consistently saves **70–75% of frames relative to UFP** and nearly **99% relative to raw video averages**, while preserving or improving retrieval performance. The S+D+T configuration achieves the optimal trade-off between semantic coverage and efficiency, demonstrating the complementarity of its three cues.

4.2. Keyframe Extraction

We evaluate KeyScore on two widely used keyframe extraction benchmarks: TVSum20 [39] and SumMe [18]. For TVSum, we pair KeyScore with CLIP-ViT-H/14 [36], while for SumMe, we use PE_{core}G-Video [7] with KeyScore. Following the evaluation protocol of [8], we report F1 scores computed using frame-level color histogram similarity. As shown in Table 6, KeyScore and its variants achieve strong results, outperforming $\text{TRIPSS}_{\text{semantic}}$ and several recent baselines, despite relying solely on semantic alignment.

4.3. Runtime & Frame Efficiency Analysis (TVSum20)

We further evaluate sampling efficiency on **TVSum20**, which contains 20 videos with 2.5k–6.9k frames each. Uniform and SCFP [23] sample 8 frames per video, while STACFP adaptively selects 5–8 frames (typically 8).

Table 7 summarizes per-video runtime and frame reduction rates. UFP is the fastest but lacks adaptivity. SCFP incurs heavy clustering cost over all frames, whereas STACFP achieves a strong balance—processing long videos **3**$\times$ faster than SCFP while retaining comparable coverage.

Conclusion. STACFP achieves near-identical frame reduction to static methods ($\sim$99.8%) while reducing runtime by over **68**% compared to SCFP, demonstrating that adaptive clustering delivers both efficiency and scalability for long videos.

4.4. Zero-Shot Video Action Classification

We further evaluate our frame proposal and scoring strategies on the HMDB-51 [24] benchmark, which contains 51 human action categories. Following Qwen-2.5-VL [44], we first generate captions for each video clip and use them to guide KeyScore-based frame scoring. For classification, we employ the PE_{core}G-Video [7] frame-based video encoder. Frames are selected according to score thresholds, and for scoring-based methods we report the best F1 obtained across thresholds.

Table 8 presents zero-shot video action classification results on HMDB51. Among the baseline models, InternVL [49], InternVideo2 [51], and SigLIP2-g-opt [45] achieve F1 scores in the 0.518–0.555 range with FRR-Avg

Table 5. **KeyScore frame reduction across datasets.** We report average selected frames (ASF), FRR-UFP, and FRR-Avg. Combining semantic (S), temporal (T), and drop-impact (D) cues yields the best balance between efficiency and robustness. Note FRR can be negative when a variant uses more than 8 frames

Frame Scoring	MSR-VTT (avg. 408)			MSVD (avg. 275)			DiDeMo (avg. 1728)		
	ASF	FRR-UFP↑	FRR-Avg↑	ASF	FRR-UFP↑	FRR-Avg↑	ASF	FRR-UFP↑	FRR-Avg↑
PE$_{\text{core}}$G-Video + KeyScore(S)	2.00	0.75	0.99	5.00	0.38	0.98	3.00	0.63	0.99
PE$_{\text{core}}$G-Video + KeyScore(T)	8.20	-0.03	0.98	4.00	0.50	0.99	2.00	0.75	0.99
PE$_{\text{core}}$G-Video + KeyScore(D)	3.00	0.63	0.99	6.00	0.25	0.98	2.00	0.75	0.99
PE$_{\text{core}}$G-Video + KeyScore(S+T)	3.30	0.59	1.00	2.00	0.75	0.99	2.00	0.75	0.99
PE$_{\text{core}}$G-Video + KeyScore(D+T)	2.57	0.68	0.99	2.00	0.50	0.99	2.00	0.75	0.99
PE$_{\text{core}}$G-Video + KeyScore(S+D)	2.69	0.66	0.99	2.00	0.75	0.99	2.00	0.75	0.99
PE$_{\text{core}}$G-Video + KeyScore(S+D+T)	**2.50**	**0.69**	**0.99**	**2.00**	**0.75**	**0.99**	**2.00**	**0.75**	**0.99**

Table 6. **F1 scores on TVSum20 [39] and SumMe [18].** KeyScore with CLIP/PE outperforms or matches prior baselines.

TVSum20		SumMe	
Method	**F1↑**	**Method**	**F1↑**
HistDiff [37]	0.338	H-MAN [30]	0.518
VS-UID [14]	0.462	SUM-GDA [26]	0.528
GMC [15]	0.483	STVS [22]	0.536
VSUMM [11]	0.489	TAC-SUM [20]	0.545
KMKey [33]	0.504	PGL-SUM [5]	0.556
LBP-Shot [34]	0.505	SMN [48]	0.583
VS-Inception [14]	0.517	AugFusion [35]	0.584
LMSKE [40]	0.531	Ldpp-c [21]	0.588
TRIPSS	0.610	TRIPSS	0.590
CLIP [36] + KeyScore	**0.539**	**PE [7] + KeyScore**	**0.655**

Table 7. **Runtime and frame reduction on TVSum20.** FRR-Avg: ratio of discarded frames to total video length.

Method	Frames	Runtime (s)	FRR-Avg (%)
UFP (Uniform)	8	15.04	99.7
SCFP (Kanta [23])	8	178.95	99.7
STACFP (ours)	5–8	**56.20**	**99.8**

values of 0.915, reflecting strong but comparable performance across different architectures and resolutions. In contrast, PE$_{\text{core}}$G-Video + KeyScore delivers a substantial improvement, achieving an F1 of **0.675** and an FRR-Avg of **0.972**. This represents an absolute gain of +12.0 F1 points over the strongest baseline (InternVL), while simultaneously discarding a larger fraction of frames. The higher FRR-Avg demonstrates that KeyScore can aggressively reduce frame inputs while preserving the frames most critical for action understanding.

These results reveal two important trends. First, semantic- and context-aware scoring is more effective for action classification than dense uniform sampling, as KeyScore prioritizes frames aligned with action semantics rather than treating all frames equally. Second, KeyScore's ability to retain fewer frames yet improve accuracy highlights its efficiency, making it particularly suitable for large-

Table 8. Zero-shot video action classification results on **HMDB51** [24]. Our method (PE$_{\text{core}}$G-Video + KeyScore) achieves the best F1 with the highest FRR-Avg.

Model	Resolution	F1↑	FRR-Avg↑
InternVL [49]	224	0.555	0.915
InternVideo2 [51]	224	0.539	0.915
SigLIP2-g-opt [47]	384	0.518	0.915
PE$_{\text{core}}$G-Video [7] + KeyScore	**448**	**0.675**	**0.972**

scale video understanding tasks where both performance and computational cost are critical. Overall, the combination of PE$_{\text{core}}$G-Video with KeyScore establishes a new state of the art on HMDB51 under zero-shot evaluation by jointly optimizing recognition accuracy and frame efficiency.

5. Discussion & Limitations

KeyScore substantially reduces frame redundancy but currently relies on accompanying captions for semantic guidance. Future extensions could explore unsupervised or generative captioning to broaden applicability to unlabeled or streaming videos.

6. Conclusion

We introduced **KeyScore**, a caption-grounded frame scoring framework that integrates semantic, temporal, and contextual cues to select the most informative video frames. Across retrieval, summarization, and action recognition tasks, KeyScore improves accuracy while cutting frame usage by 70–99% versus full videos and 63–75% over 8-frame baselines. By converting video–caption pairs into frame-level importance, KeyScore enables efficient keyframe selection for video encoders and Video-LLMs. Future work will explore unsupervised or auto-captioned variants and integrate KeyScore into long-form and streaming multimodal systems for scalable video understanding.

References

[1] Moloud Abdar, Meenakshi Kollati, Swaraja Kuraparthi, Farhad Pourpanah, Daniel McDuff, Mohammad Ghavamzadeh, Shuicheng Yan, Abduallah Mohamed, Abbas Khosravi, Erik Cambria, et al. A review of deep learning for video captioning. *IEEE TPAMI*, 2024. 1

[2] Mohamed Afham, Satya Narayan Shukla, Omid Poursaeed, Pengchuan Zhang, Ashish Shah, and Sernam Lim. Revisiting kernel temporal segmentation as an adaptive tokenizer for long-form video understanding. In *CVPR*, 2023. 2

[3] Imtiaz Ahmed, Sadman Islam, Partha Protim Datta, Imran Kabir, Naseef Ur Rahman Chowdhury, and Ahshanul Haque. Qwen 2.5: A comprehensive review of the leading resource-efficient llm with potentioal to surpass all competitors. *Authorea Preprints*, 2025. 2

[4] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with natural language. In *ICCV*, 2017. 5, 6

[5] Evlampios Apostolidis, Georgios Balaouras, Vasileios Mezaris, and Ioannis Patras. Combining global and local attention with positional encoding for video summarization. In *ISM*, pages 226–234, 2021. 8

[6] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, 2021. 2

[7] Daniel Bolya, Po-Yao Huang, Peize Sun, Jang Hyun Cho, Andrea Madotto, Chen Wei, Tengyu Ma, Jiale Zhi, Jathushan Rajasegaran, Hanoona Rasheed, et al. Perception encoder: The best visual embeddings are not at the output of the network. *arXiv preprint arXiv:2504.13181*, 2025. 5, 6, 7, 8

[8] Mert Can Cakmak, Nitin Agarwal, and Diwash Poudel. Tripss: A tri-modal keyframe extraction framework using perceptual, structural, and semantic representations. *arXiv preprint arXiv:2506.05395*, 2025. 7

[9] David Chen and William B Dolan. Collecting highly parallel data for paraphrase evaluation. In *ACL*, 2011. 5, 6

[10] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *CVPR*, pages 24185–24198, 2024. 7

[11] Sandra Eliza Fontes De Avila, Ana Paula Brandao Lopes, Antonio da Luz Jr, and Arnaldo de Albuquerque Araújo. Vsumm: A mechanism designed to produce static video summaries and a novel evaluation method. *Pattern Recognition Letters*, 32(1):56–68, 2011. 2, 3, 8

[12] Xingning Dong, Zipeng Feng, Chunluan Zhou, Xuzheng Yu, Ming Yang, and Qingpei Guo. M2-raap: A multi-modal recipe for advancing adaptation-based pre-training towards effective and efficient zero-shot video-text retrieval. In *SIGIR*, 2024. 1

[13] Naveed Ejaz, Tayyab Bin Tariq, and Sung Wook Baik. Adaptive key frame extraction for video summarization using an aggregation mechanism. *Journal of Visual Communication and Image Representation*, 23(7):1031–1040, 2012. 2

[14] Luis C Garcia-Peraza-Herrera, Sebastien Ourselin, and Tom Vercauteren. Videosum: A python library for surgical video summarization. *arXiv preprint arXiv:2303.10173*, 2023. 8

[15] Hana Gharbi, Sahbi Bahroun, Mohamed Massaoudi, and Ezzeddine Zagrouba. Key frames extraction using graph modularity clustering for efficient video summarization. In *ICASSP*, pages 1502–1506, 2017. 8

[16] Boqing Gong, Wei-Lun Chao, Kristen Grauman, and Fei Sha. Diverse sequential subset selection for supervised video summarization. *NeurIPS*, 2014. 2

[17] Weiyu Guo, Ziyang Chen, Shaoguang Wang, Jianxiang He, Yijie Xu, Jinhui Ye, Ying Sun, and Hui Xiong. Logic-in-frames: Dynamic keyframe search via visual semantic-logical verification for long video understanding. *arXiv preprint arXiv:2503.13139*, 2025. 3

[18] Michael Gygli, Helmut Grabner, Hayko Riemenschneider, and Luc Van Gool. Creating summaries from user videos. In *ECCV*, 2014. 7, 8

[19] Bin Huang, Xin Wang, Hong Chen, Zihan Song, and Wenwu Zhu. Vtimellm: Empower llm to grasp video moments. In *CVPR*, 2024. 2

[20] Hai-Dang Huynh-Lam, Ngoc-Phuong Ho-Thi, Minh-Triet Tran, and Trung-Nghia Le. Cluster-based video summarization with temporal context awareness. In *Pacific-Rim Symposium on Image and Video Technology*, pages 15–28. Springer, 2023. 8

[21] Michail Kaseris, Ioannis Mademlis, and Ioannis Pitas. Exploiting caption diversity for unsupervised video summarization. In *ICASSP*, pages 1650–1654, 2022. 8

[22] Shamal Kashid, Lalit K Awasthi, Krishan Berwal, and Parul Saini. Stvs: Spatio-temporal feature fusion for video summarization. *TMM*, 2024. 8

[23] KeplerLab. Katna: Tool for automating video keyframe extraction, video compression, image autocrop and smart image resize tasks. https://github.com/keplerlab/katna, 2025. 1, 2, 3, 5, 6, 7, 8

[24] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *ICCV*, 2011. 7, 8

[25] Kunchang Li, Yali Wang, Yizhuo Li, Yi Wang, Yinan He, Limin Wang, and Yu Qiao. Unmasked teacher: Towards training-efficient video foundation models. In *CVPR*, 2023. 7

[26] Ping Li, Qinghao Ye, Luming Zhang, Li Yuan, Xianghua Xu, and Ling Shao. Exploring global diverse attention via pairwise temporal relation for video summarization. *Pattern Recognition*, 111:107677, 2021. 8

[27] Hao Liang, Jiapeng Li, Tianyi Bai, Xijie Huang, Linzhuang Sun, Zhengren Wang, Conghui He, Bin Cui, Chong Chen, and Wentao Zhang. Keyvideollm: Towards large-scale video keyframe selection. *arXiv preprint arXiv:2407.03104*, 2024. 1, 2

[28] Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. *arXiv preprint arXiv:2311.10122*, 2023. 2

CVPR #22338

[29] Tianrui Liu, Qingjie Meng, Jun-Jie Huang, Athanasios Vlontzos, Daniel Rueckert, and Bernhard Kainz. Video summarization through reinforcement learning with a 3d spatio-temporal u-net. *IEEE TIP*, 31:1573–1586, 2022. 2

[30] Yen-Ting Liu, Yu-Jhe Li, Fu-En Yang, Shang-Fu Chen, and Yu-Chiang Frank Wang. Learning hierarchical self-attention for video summarization. In *ICIP*, 2019. 8

[31] Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. Clip4clip: An empirical study of clip for end to end video clip retrieval and captioning. *Neurocomputing*, 508:293–304, 2022. 7

[32] Yiwei Ma, Guohai Xu, Xiaoshuai Sun, Ming Yan, Ji Zhang, and Rongrong Ji. X-clip: End-to-end multi-grained contrastive learning for video-text retrieval. In *ACM MM*, 2022. 7

[33] Bilyamin Muhammad, Bashir Sadiq, Ime Umoh, and H Bello-Salau. A k-means clustering approach for extraction of keyframes in fast-moving videos. *International Journal of Information Processing and Communication (IJIPC)*, 9 (1&2):147–157, 2020. 8

[34] HM Nandini, HK Chethan, and BS Rashmi. Shot based keyframe extraction using edge-lbp approach. *Journal of King Saud University-Computer and Information Sciences*, 34(7):4537–4545, 2022. 8

[35] Theodoros Psallidas and Evaggelos Spyrou. Video summarization based on feature fusion and data augmentation. *Computers*, 12(9):186, 2023. 8

[36] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 7, 8

[37] Jorge Michel Diaz Rodriguez, Pin Yao, and Wanggen Wan. Selection of key frames through the analysis and calculation of the absolute difference of histograms. In *ICALIP*, pages 423–429, 2018. 8

[38] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987. 4

[39] Yale Song, Jordi Vallmitjana, Amanda Stent, and Alejandro Jaimes. Tvsum: Summarizing web videos using titles. In *CVPR*, 2015. 7, 8

[40] Kailong Tan, Yuxiang Zhou, Qianchen Xia, Rui Liu, and Yong Chen. Large model based sequential keyframe extraction for video summarization. In *CMLD*, 2024. 2, 5, 6, 8

[41] Reuben Tan, Ximeng Sun, Ping Hu, Jui-hsien Wang, Hanieh Deilamsalehy, Bryan A Plummer, Bryan Russell, and Kate Saenko. Koala: Key frame-conditioned long video-llm. In *CVPR*, 2024. 1, 2

[42] Hao Tang, Lei Ding, Songsong Wu, Bin Ren, Nicu Sebe, and Paolo Rota. Deep unsupervised key frame extraction for efficient video classification. *ACM TOMCCAP*, 19(3):1–17, 2023. 1, 2

[43] Xi Tang, Jihao Qiu, Lingxi Xie, Yunjie Tian, Jianbin Jiao, and Qixiang Ye. Adaptive keyframe sampling for long video understanding. In *CVPR*, 2025. 2, 3

[44] Qwen Team. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024. 2, 7

[45] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025. 7

[46] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025. 7

[47] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025. 8

[48] Junbo Wang, Wei Wang, Zhiyong Wang, Liang Wang, Dagan Feng, and Tieniu Tan. Stacked memory network for video summarization. In *Proceedings of the 27th ACM international conference on multimedia*, pages 836–844, 2019. 8

[49] Yi Wang, Kunchang Li, Yizhuo Li, Yinan He, Bingkun Huang, Zhiyu Zhao, Hongjie Zhang, Jilan Xu, Yi Liu, Zun Wang, et al. Internvideo: General video foundation models via generative and discriminative learning. *arXiv preprint arXiv:2212.03191*, 2022. 7, 8

[50] Yi Wang, Yinan He, Yizhuo Li, Kunchang Li, Jiashuo Yu, Xin Ma, Xinhao Li, Guo Chen, Xinyuan Chen, Yaohui Wang, et al. Internvid: A large-scale video-text dataset for multimodal understanding and generation. *arXiv preprint arXiv:2307.06942*, 2023. 7

[51] Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Guo Chen, Baoqi Pei, Rongkun Zheng, Zun Wang, Yansong Shi, et al. Internvideo2: Scaling foundation models for multimodal video understanding. In *ECCV*, pages 396–416, 2024. 7, 8

[52] Wenhao Wu, Haipeng Luo, Bo Fang, Jingdong Wang, and Wanli Ouyang. Cap4video: What can auxiliary captions do for text-video retrieval? In *CVPR*, 2023. 1

[53] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *CVPR*, 2016. 5, 6

[54] Hongwei Yin, Richard O Sinnott, and Glenn T Jayaputera. A survey of video-based human action recognition in team sports. *Artificial intelligence review*, 57(11):293, 2024. 1

[55] Bingqing Zhang, Zhuo Cao, Heming Du, Xin Yu, Xue Li, Jiajun Liu, and Sen Wang. Tokenbinder: Text-video retrieval with one-to-many alignment paradigm. In *WACV*, 2025. 2

[56] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*, 2023. 2

[57] Ke Zhang, Wei-Lun Chao, Fei Sha, and Kristen Grauman. Summary transfer: Exemplar-based subset selection for video summarization. In *CVPR*, 2016. 2

[58] Long Zhao, Nitesh B Gundavarapu, Liangzhe Yuan, Hao Zhou, Shen Yan, Jennifer J Sun, Luke Friedman, Rui Qian, Tobias Weyand, Yue Zhao, et al. Videoprism: A foundational visual encoder for video understanding. *arXiv preprint arXiv:2402.13217*, 2024. 7