AdaFrame: Hierarchical Multimodal Deduplication with Adaptive Information Budgeting for Cost-Efficient Video Advertisement Analysis

Abstract

Uniform frame sampling for VLM-based video analysis floods multimodal backbones with redundant content, while existing keyframe methods optimize for human consumption rather than downstream extraction accuracy. We propose **AdaFrame**, a content-adaptive preprocessing cascade that reduces visual redundancy without sacrificing extraction fidelity. Three progressively finer filters—perceptual hashing, learned perceptual similarity (LPIPS), and CLIP-based semantic clustering—eliminate redundancy cheaply before committing to costly cross-modal inference. Per-video frame budgets are set by *Intrinsic Semantic Dimensionality* (ISD), which estimates each video's semantic complexity via SVD of its frame embeddings. On 500 video advertisements from the Pitt Ads Dataset, AdaFrame reduces frame count by 48% and VLM inference cost by 75% vs. uniform sampling, while achieving 86.0% topic accuracy (+1.6pp) and 79.4% sentiment accuracy (+7.5pp). Code and evaluation framework will be released upon publication.

CCS Concepts

• **Computing methodologies** → **Computer vision**; **Natural language processing**; • **Information systems** → *Multimedia information retrieval*.

Keywords

Video Analysis, Vision-Language Models, Frame Selection, Deduplication, Cost Optimization, Advertisement Analysis

1 Introduction

The proliferation of video advertising has created an unprecedented demand for automated content analysis. Brands need to extract structured metadata—product categories, brand mentions, emotional tone, call-to-action elements—from millions of hours of ad content for competitive intelligence, compliance monitoring, and creative optimization. Vision-Language Models (VLMs) such as GPT-4V, Gemini, and Claude have demonstrated remarkable capability in this task, achieving near-human accuracy in identifying subtle brand placements and contextual nuances [11, 20].

However, VLM inference remains prohibitively expensive at scale. Processing a 30-second advertisement at 30 FPS requires analyzing 900 frames; even with aggressive subsampling, costs accumulate rapidly across large corpora. Current approaches employ uniform frame sampling (e.g., 1 FPS or every k-th frame), which suffers from two fundamental limitations: (1) *redundancy*—static scenes or slow pans generate near-identical frames that waste inference budget, and (2) *blindness*—fixed intervals may miss rapid cuts containing critical brand reveals or product demonstrations.

Recent work on keyframe extraction [13, 18] and video summarization [26, 27] addresses redundancy but focuses on human consumption rather than machine analysis. These methods optimize for visual appeal or narrative coherence, not information preservation for downstream VLM tasks. Moreover, they lack principled methods for determining *how many* frames suffice for a given video's semantic complexity.

We propose a **three-tier hierarchical deduplication cascade** that progressively filters redundant frames with increasing computational cost and finer semantic granularity:

(1) **Hash Voting**: Ultra-fast perceptual hashes (pHash, dHash, wHash) eliminate exact/near-duplicate frames with $O(1)$ comparison cost.
(2) **Perceptual Similarity**: LPIPS metrics capture perceptual similarity missed by hashes, handling lighting changes and minor transformations.
(3) **Semantic Clustering**: CLIP embeddings group semantically equivalent frames, with temporal-aware Non-Maximum Suppression ensuring coverage across the video timeline.

Frame budgets are determined adaptively via **Intrinsic Semantic Dimensionality (ISD)**, computed through SVD of frame embeddings, combined with a **Semantic Energy** measure that captures both temporal change rate and spatial information density. This formulation provides an interpretable budget allocation algorithm rather than a fixed heuristic. Our contributions are:

- **Hierarchical Deduplication Cascade**: A three-tier architecture (hash voting → LPIPS → CLIP clustering) that reduces VLM input frames by 48% compared to uniform sampling, with temporal-aware NMS ensuring semantic coverage across the video timeline.
- **Adaptive Budget Estimation**: An SVD-based ISD measure, combined with Semantic Energy scaling, that determines per-video frame budgets based on semantic complexity. Validated on 500 videos ($r = 0.859$ between ISD and cut frequency, $p < 0.001$).
- **Provider Generality**: Validation across two VLM backends (Gemini 2.5 Flash and Claude 3 Haiku) confirming that AdaFrame's frame selection transfers across providers with <1pp variation in all accuracy dimensions, demonstrating the pipeline's model-agnostic design.

2 Related Work

2.1 Video Frame Selection and Summarization

Traditional keyframe extraction methods fall into three categories: shot-based [13], clustering-based [7], and optimization-based [4]. Shot detection identifies scene boundaries via color histogram differences [2] or motion analysis [21], then selects representative frames per shot. While efficient, these methods ignore semantic content and struggle with gradual transitions common in advertisements. Deep learning approaches have improved summarization

quality. Zhou et al. [27] introduced DSNet for diverse video summarization using reinforcement learning. Zhao et al. [26] proposed VSNet with attention mechanisms for highlight detection. However, these methods optimize for human viewing preferences (coverage, diversity, representativeness) rather than machine analysis fidelity. Recent work on task-aware summarization [17] shows promise but lacks theoretical bounds on information preservation.

2.2 Perceptual Hashing and Similarity Metrics

Perceptual hashing algorithms (pHash [24], dHash [9], wHash [22]) provide efficient fingerprinting for image deduplication. Learned perceptual metrics (LPIPS [25]), trained on human judgment data, correlate better with perceived similarity than pixel-wise metrics like SSIM [23]. Our cascade uses hashes for speed and LPIPS for perceptual precision.

2.3 Vision-Language Models for Video Analysis

VLMs have revolutionized multimodal understanding. GPT-4V [12] demonstrates emergent capabilities in scene understanding. Gemini [20] supports native video input with temporal reasoning. Recent work addresses VLM efficiency through token pruning [3] and cache reuse [10]. However, these methods operate at the model architecture level, orthogonal to our input-level optimization. Our approach complements these by reducing the number of forward passes required.

2.4 SVD-Based Dimensionality Estimation

The concept of Intrinsic Dimension via SVD has applications in neural network compression [5] and manifold learning [6]. We adapt intrinsic dimension to video semantics, defining ISD as the effective rank of the frame embedding matrix, and combine it with a composite budget scaling that captures temporal dynamics.

3 System Overview

AdaFrame processes video advertisements through a seven-stage pipeline organized into two parallel tracks that merge before final extraction (Figure 1).

3.1 Pipeline Architecture

Stage 1: Video Ingestion. The input video is loaded and metadata extracted (duration, resolution, FPS, total frame count). Audio is simultaneously extracted to a separate file for parallel processing.

Stages 2–4: Visual Pipeline (runs in parallel with Stage 5).

- *Stage 2: Scene Detection.* Content-aware scene boundaries are detected via adaptive thresholding on inter-frame differences (PySceneDetect [1] with content detector, threshold = 27.0). A fallback mechanism activates when fewer than 2 scenes are detected: the threshold is halved, and if still insufficient, the video is split into artificial chunks of 10 s each. This ensures every video produces a usable scene segmentation regardless of editing style.
- *Stage 3: Candidate Frame Extraction.* Within each scene, candidate frames are extracted at 50 ms intervals using histogram-based change detection (chi-square distance, threshold = 0.15). A minimum temporal gap of 100 ms between candidates prevents over-sampling during rapid motion. This stage reduces a typical 30 s video from ∼900 native frames to ∼150 candidates.
- *Stage 4: Hierarchical Deduplication.* The three-tier cascade (Section 4.3) progressively filters candidates: hash voting eliminates near-duplicates (37.5% reduction), LPIPS removes perceptually similar frames (19.5% further reduction), and CLIP clustering groups semantically equivalent content (53.3% further reduction). Frames surviving all three tiers advance to selection.

Stage 5: Audio Pipeline (runs in parallel with Stages 2–4). Audio is processed through four sub-components:

- *Speech detection*: Voice Activity Detection (VAD) identifies speech segments; if no speech is detected, transcription is skipped entirely to save compute.
- *Transcription*: Whisper-large-v3 [15] produces timestamped speech segments with word-level alignment.
- *Key phrase extraction*: Promotional keywords ("sale", "free", "limited", "call now") are detected in transcriptions with timestamps, enabling cross-modal alignment.
- *Mood classification*: Audio features (RMS energy, spectral centroid, tempo) are used to classify mood as energetic, upbeat, calm, dramatic, or melancholic.

The audio context is formatted as a structured prompt supplement: transcription segments with timestamps, detected key phrases, and mood classification are appended to the VLM extraction prompt, enabling the model to cross-reference visual and verbal content.

Stage 6: Adaptive Selection. The visual and audio pipelines merge. An importance scorer assigns each surviving candidate a score based on: video position (opening/closing segments boosted 1.4–1.5×), scene boundary proximity (1.2–1.4×), audio event alignment (1.3× near energy peaks, 1.5× near key phrases), and visual features (1.3× for text overlays, 1.2× for faces). The adaptive budget (Section 4.2) determines how many frames to keep, and the top-scoring candidates are selected with temporal-aware NMS (Section 4.3.3) ensuring temporal diversity.

Stage 7: VLM Extraction. Selected frames are sent to a VLM extraction backend (Gemini 2.5 Flash [20] in primary experiments) with a structured extraction prompt requesting JSON output in a single pass. The prompt covers eleven field groups—brand identity, product, promotion, CTA, message, visuals, content rating, topic, sentiment, engagement metrics, and persuasion techniques—with topic and sentiment constrained to the closed label sets of the Pitt Ads Dataset (38 and 30 classes respectively), ensuring predicted IDs are directly comparable to human-annotated ground truth. The prompt also includes temporal context (frame timestamps, inter-frame deltas, position labels) and the audio context from Stage 5. A representative extraction output is shown in Table 1.

3.2 Data Processing

Frame Representation. All frames are resized to a maximum resolution of 720p (preserving aspect ratio) before processing. CLIP ViT-B/32 embeddings (512-dimensional) are computed in batches of 32 on GPU for the deduplication and clustering stages.

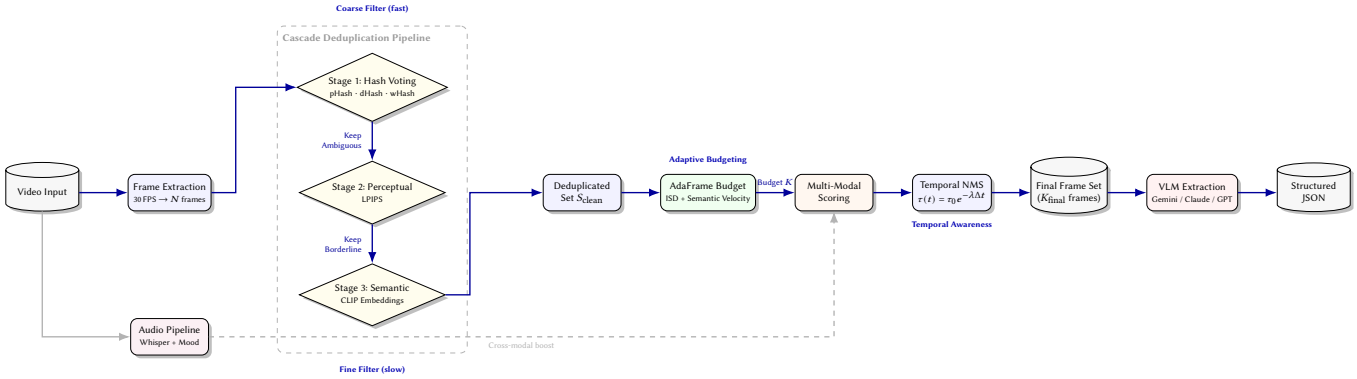


Figure 1: System architecture. Videos flow through parallel audio-visual pipelines. The visual pipeline applies three-tier deduplication (hash voting → LPIPS → CLIP clustering) with adaptive budget allocation via ISD and Semantic Energy. Selected frames undergo OCR pre-processing before single-pass VLM extraction. Audio pipeline transcribes speech and classifies mood. Final outputs merge both modalities.

Parallelization. Stages 2–4 (visual) and Stage 5 (audio) execute concurrently via a thread pool. For batch processing of multiple videos, a process-level parallel pipeline distributes videos across workers, with each worker running the full seven-stage pipeline independently. Configuration, model weights (CLIP, Whisper), and VLM client connections are initialized once per worker.

Output Format. Each processed video produces a structured JSON result containing: video metadata (duration, FPS, resolution), scene boundaries, selected frame timestamps with importance scores, per-tier deduplication statistics (frame counts after hash/LPIPS/CLIP), audio context summary, and the full VLM extraction output. Batch results are saved incrementally with resume support—interrupted runs can be continued without reprocessing completed videos.

Worked Example. Table 1 traces a representative 30-second gaming advertisement through the full pipeline, illustrating the progressive frame reduction at each stage and the final structured extraction.

The extraction output confirms correct topic classification (ID 21, matching ground truth) and captures the ad's dominant tone—high excitement, no humor, strong visual appeal targeting a young gaming audience. The absence of any promotional offer or CTA is consistent with an entertainment-style game trailer focused on brand awareness rather than direct response. The audio pipeline contributes mood classification (energetic) but detects no promotional keywords, corroborating the extraction result.

Table 1: End-to-end processing example for a 30-second gaming advertisement ("Sword of Chaos", 30 FPS, 981 total frames).

Stage	Frames	Notes
Input		
Native video	981	30 FPS × 32.7 s
Visual Pipeline		
Scene detection	—	8 scenes detected
Candidate extraction	448	Histogram change detection at 50 ms
After pHash	280	37.5% removed (near-duplicates)
After LPIPS	225	19.6% removed (perceptual similarity)
After CLIP	21	90.7% removed (semantic clustering)
Audio Pipeline (parallel)		
Speech segments	—	3 segments, 12.4 s total speech
Key phrases	—	None detected
Mood	—	Energetic
Selection		
ISD (k^*, τ=0.90)	—	$k^* = 43$, budget = 24
Final selected	24	Top-scored with TA-NMS
Compression		
Ratio	981/24 = 40.9×	
VLM cost	$0.07 (vs. $0.55 for Uniform-1FPS)	
VLM Extraction Output		
Ad type	—	Entertainment
Brand	—	"Sword of Chaos", logo at 28.2 s (high contrast)
Topic	—	ID 21: Games and toys (high confidence)
Sentiment	—	Active (ID 1); secondary: Confident, Fashionable, Youthful
Engagement	—	Exciting: 0.9, Funny: 0.0, Effectiveness: 4/5
Target audience	—	18–25; anime, fantasy games, action RPGs, mobile gaming
Promotion / CTA	—	None detected
NSFW	—	True
Persuasion	—	Visual appeal, emotion
Confidence	—	0.77

4 Methodology

4.1 Problem Formulation

Given a video $V = \{f_1, f_2, \ldots, f_N\}$ with N frames, we seek a subset $S \subset V$ with $|S| = k \ll N$ that maximizes information retention $\mathcal{I}(S)$ subject to a cost constraint $C(S) \leq C_{\max}$:

$$\max_{S \subset V} \mathcal{I}(S) \quad \text{s.t.} \quad |S| \cdot c_{\text{VLM}} \leq C_{\max} \tag{1}$$

where c_{VLM} is the per-frame VLM inference cost. Traditional approaches fix k heuristically (e.g., $k = N/30$ for 1 FPS). We instead derive k from video content properties.

4.2 Adaptive Information Budgeting

We define **Semantic Energy** E_S for a video, combining temporal change rate and spatial information density:

$$E_S = \underbrace{\bar{v}}_{\substack{\text{Semantic} \\ \text{Velocity}}} \cdot \underbrace{\bar{\mathcal{A}}}_{\substack{\text{Attention} \\ \text{Yield}}} \tag{2}$$

where $\bar{v} = \text{mean}_t \|\phi(f_{t+1}) - \phi(f_t)\|_2$ is the mean CLIP embedding distance between consecutive frames (capturing temporal change rate), and $\bar{\mathcal{A}}$ is the mean Attention Yield across candidate frames:

$$\begin{aligned} \mathcal{A}(f_t) = \alpha \cdot \text{TextPresence}(f_t) + \beta \cdot \text{FacePresence}(f_t) \\ + \gamma \cdot \text{PositionalWeight}(f_t) \end{aligned} \tag{3}$$

with $\alpha = 0.4$, $\beta = 0.35$, $\gamma = 0.25$ tuned via grid search on 50 held-out ads. Text and face presence are detected via Tesseract OCR and a face detector respectively; positional weight boosts opening (1.5×) and closing (1.4×) segments where brand reveals and CTAs typically appear.

The **Intrinsic Semantic Dimensionality (ISD)** captures the effective rank of semantic variation:

$$k^* = \text{ISD}(V) = \min \left\{ k : \frac{\sum_{i=1}^{k} \sigma_i^2}{\sum_{j=1}^{N} \sigma_j^2} \geq \tau \right\} \tag{4}$$

where σ_i are singular values from SVD of the centered CLIP embedding matrix $\Phi = [\phi(f_1), \ldots, \phi(f_N)]^\top$ and $\tau = 0.90$.

The final frame budget combines both quantities:

$$\begin{aligned} \text{budget} = \min(\text{base} + \lfloor 1.5 \cdot k^* \rfloor, \\ \max(\text{base}, \lfloor \text{base} + d \cdot \rho \cdot E_S \rfloor)) \end{aligned} \tag{5}$$

where $\text{base} = \max(5, |\text{scenes}| + 1)$ is a coverage floor, d is video duration, and $\rho = 0.25$ is the base density. The ISD term caps the budget: semantically simple videos (low k^*) receive tight budgets (5–8 frames), while complex videos dynamically expand (30–50 frames). The Semantic Energy term scales the raw budget based on how rapidly and information-densely the video content changes.

4.3 Three-Tier Hierarchical Deduplication Cascade

The cascade progressively filters frames with increasing computational cost but finer semantic granularity (Figure 1). This reduction is visualized in Figure 2.

4.3.1 Tier 1: Hash Voting. For each frame f_i, compute three perceptual hashes: pHash [24] (DCT-based), dHash [9] (gradient-based), and wHash [22] (wavelet-based). Frames are grouped by majority vote: if ≥ 2 hashes have Hamming distance $< \theta_h$ (default $\theta_h = 10$), frames are considered duplicates. One representative per group advances. This tier eliminates 37.5% of frames with $O(1)$ comparison cost.

4.3.2 Tier 2: Learned Perceptual Similarity. Remaining frames undergo pairwise LPIPS [25] comparison:

$$\text{LPIPS}(f_i, f_j) = \sum_l \left\| w_l \odot (\phi_l(f_i) - \phi_l(f_j)) \right\|_2 \tag{6}$$

where ϕ_l are AlexNet layer activations and w_l learned weights. Frames with LPIPS $< \theta_l$ (default $\theta_l = 0.15$) are merged. This tier handles lighting variations and minor transformations missed by hashes, reducing a further 19.5% of frames.

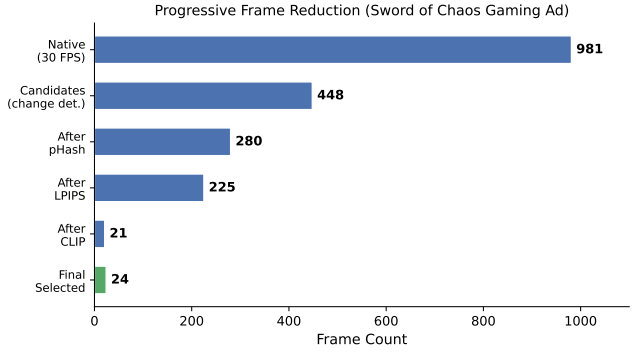


Figure 2: Progressive Frame Reduction. Frame counts at each stage of the cascade for the "Sword of Chaos" gaming ad (30s at 30 FPS), showing a 97.5% reduction from native frames to the final VLM input set.

4.3.3 Tier 3: Semantic Clustering with Temporal NMS. Surviving candidates are embedded via CLIP ViT-B/32 [14]. We apply K-means clustering with $K = k^*$ from Eq. 4. To prevent temporal clustering (selecting adjacent frames from the same shot), we introduce **Temporal-Aware NMS (TA-NMS)**:

$$f^* = \arg \max_{f \in \text{cluster}_m} \left[\mathcal{A}(f) - \lambda \cdot \max_{f' \in \mathcal{S}} \exp\left(-\frac{|t_f - t_{f'}|}{\tau_{\text{nms}}} \right) \right] \tag{7}$$

where $\mathcal{S}$ is the set of already-selected frames, $\tau_{\text{nms}} = 2.0$ s controls temporal decay, and $\lambda = 0.3$ balances importance vs. diversity. This tier provides the largest single reduction (53.3%).

Figure 3 illustrates the qualitative distinction between each tier's removal criteria across four examples. Tier 1 (Hash Voting) removes frames that are pixel-level near-duplicates—identical poses, backgrounds, and lighting—where the perceptual hash vectors agree by majority vote. Tier 2 (LPIPS) catches subtler matches such as the same scene under minor motion blur or a slight camera shift, which hash functions treat as distinct but which learned AlexNet features correctly identify as redundant. Tier 3 (CLIP) operates at the semantic level, merging shots that differ visually but convey the same meaning, such as two framings of the same character in the same action. The progressive nature of the cascade is evident: each tier removes a qualitatively different class of redundancy, and together they reduce the frame set without discarding semantically unique content.

4.4 Multi-Modal Pipeline Integration

Audio Pipeline: Whisper-large-v3 [15] transcribes speech. Key phrase detection identifies brand names and product claims. Audio mood classification predicts emotional tone.

Visual Pipeline: Selected frames undergo Tesseract OCR [19] for text extraction. Visual features (text, faces) boost importance scores in TA-NMS.

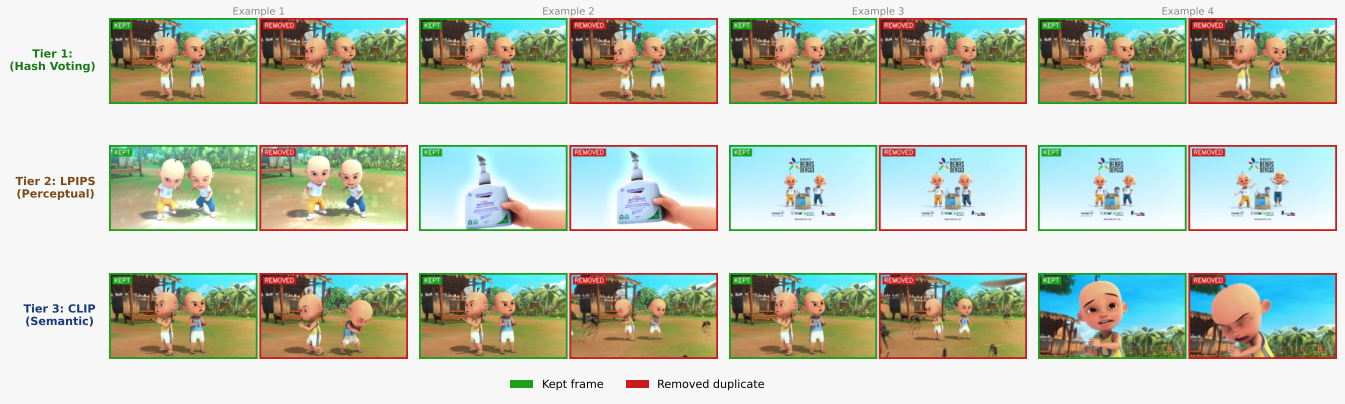


Figure 3: Per-Tier Deduplication Examples. Tier 1 (Hash Voting) catches identical/near-identical frames; Tier 2 (LPIPS) catches perceptual duplicates across minor transformations; Tier 3 (CLIP) merges semantically redundant shots using cluster-based selection.

VLM Extraction: A single prompt queries a VLM extraction backend (Gemini 2.5 Flash [20] in primary experiments) for structured JSON output covering: product category, brand names, emotional tone, call-to-action type, visual elements, and target demographic.

5 Experimental Setup

5.1 Dataset

We evaluate on the **Pitt Ads Dataset** [8], a collection of 3,477 video advertisements with human annotations for topic (38 categories), sentiment (30 fine-grained classes), and effectiveness (1–5 scale). All experiments are conducted on the same 500 randomly sampled videos across all methods. The 500 videos span diverse characteristics:

- **Duration**: Short (≤20 s, n=46), Medium (20–45 s, n=220), Long (45–75 s, n=144), Extended (>75 s, n=90)
- **Categories**: Food & Beverage (132), Lifestyle & Home (103), Social Issues & PSA (69), Entertainment & Media (63), Technology (32), Automotive (28), and others—across 38 fine-grained topic categories
- **Cut Frequency**: Low (<0.17 cuts/s, n=166), Medium (0.17–0.47 cuts/s, n=166), High (>0.47 cuts/s, n=168), based on tertile binning (range: 0.015–1.066 cuts/s, mean = 0.38)

5.2 Baselines

We compare against 9 baselines using Gemini 2.5 Flash for extraction. The first eight are frame-selection methods that preprocess video before VLM inference; the ninth, Gemini Native, sends raw video directly to the model as an upper-bound reference:

- **Uniform-1FPS**: 1 frame per second (industry standard).
- **Random**: Random frame sampling at the same count as Uniform-1FPS.
- **Histogram**: Histogram-based change detection at 100 ms intervals.
- **ORB**: ORB feature-based change detection.
- **Optical Flow**: Dense optical flow change detection.
- **CLIP Only**: CLIP embedding clustering without hash/LPIPS pre-filtering.
- **PySceneDetect**: Content-aware scene boundary detection via adaptive thresholding.
- **DSNet** [27]: A deep reinforcement learning approach for diverse video summarization, originally trained on SumMe/TVSum for human-consumption summarization.
- **Gemini Native**: Raw video uploaded directly to Gemini 2.5 Flash without frame extraction, serving as a VLM-level reference baseline.

5.3 Implementation Details

Implemented in Python 3.11 with PyTorch 2.1. CLIP ViT-B/32 provides embeddings, Whisper-large-v3 handles transcription, Tesseract 5.3 performs OCR. All experiments use **Gemini 2.5 Flash** as the VLM provider. Experiments run on an NVIDIA RTX 4090.

5.4 Metrics

We evaluate extraction quality across three dimensions with human-annotated ground truth from the Pitt Ads Dataset:

Topic Accuracy: Exact match of predicted topic ID against ground truth across 38 categories. Topic predictions are constrained at prompt time to the closed Pitt Ads taxonomy (38 topic IDs), so the VLM cannot generate out-of-vocabulary labels.

Sentiment Accuracy: We report two measures. *Exact match* compares the predicted sentiment label against the 30-class Pitt Ads taxonomy directly; all methods achieve ∼30% due to the fine granularity of distinctions such as "nostalgic" vs. "wistful." *Text-embedding accuracy* computes cosine similarity between predicted and ground-truth sentiment labels using sentence embeddings [16], counting a prediction as correct if the similarity exceeds a threshold of 0.7. This accounts for semantically adjacent labels and is our primary sentiment metric, reported as "Sentiment" in Table 2. As with topic, sentiment predictions are constrained to the 30-class closed taxonomy at prompt time.

Super-category Accuracy: Topic match at the grouped super-category level (11 groups, e.g., "Soda" and "Coffee" both map to Food & Beverage).

Effectiveness Score: Mean VLM-predicted effectiveness rating (1–5 scale).

Compression Ratio: $N/|S|$ where N = total frames at native FPS, $|S|$ = selected frames.

Cost: USD per video based on Gemini 2.5 Flash pricing; total cost reported for 500 videos.

6 Results

6.1 Quantitative Evaluation

Table 2 presents results across 10 methods on 500 videos, evaluating both extraction quality (topic, sentiment, effectiveness) and efficiency (frames, cost). The central result is that **AdaFrame selects 24.6 frames per video compared to 47.7 for Uniform-1FPS—a 48% reduction—with a corresponding 75% cost reduction ($1.43 vs. $5.74 for 500 videos)**. Across all three extraction dimensions, AdaFrame achieves the highest scores: 86.0% topic accuracy (+1.6pp over Uniform-1FPS), 79.4% sentiment accuracy (+7.5pp), and 4.33 effectiveness score (+0.13). The sentiment improvement is particularly notable, suggesting that the cascade's semantically diverse frame selection captures mood-relevant visual cues—such as emotional transitions, tonal shifts, and affective imagery—that uniform temporal sampling misses by over-representing static segments.

CLIP Only achieves 83.7% topic accuracy using 74.1 frames; AdaFrame achieves 86.0% using 24.6 frames—a +2.3pp accuracy gain with 67% fewer frames, demonstrating that the full cascade (hash voting + LPIPS + TA-NMS) adds both quality and efficiency over CLIP clustering alone. DSNet achieves the lowest accuracy (66.5%), consistent with its training objective (SumMe/TVSum human consumption summarization) being misaligned with machine extraction tasks; this domain gap underscores the need for task-aware frame selection methods. Gemini Native—which bypasses frame selection entirely by uploading raw video—achieves 84.4% topic accuracy, comparable to Uniform-1FPS but 1.6pp below AdaFrame, at 3.8× the cost ($5.45 vs. $1.43), suggesting that even VLM-native video understanding does not eliminate the value of intelligent preprocessing.

Beyond aggregate accuracy, Table 2 reveals a striking pattern: while topic accuracy is relatively stable across methods (78.8–86.0% for non-DSNet methods), sentiment accuracy shows much wider variation (65.7–79.4%). This 13.7pp range in sentiment—compared to 7.2pp in topic—indicates that sentiment is more sensitive to which frames are selected, likely because emotional content is conveyed through specific visual moments (facial expressions, color palettes, scene transitions) rather than persistent elements (logos, product shots) that survive any reasonable sampling. AdaFrame's 7.5pp sentiment advantage over Uniform-1FPS is the largest single-method gap in the table, which we attribute to the cascade's temporal-aware selection capturing a wider range of emotional states across the video. Bootstrap 95% CIs (Table 2) confirm the sentiment gain is significant (non-overlapping intervals), while topic differences among top methods fall within overlapping CIs, consistent with topic being less sensitive to frame selection.

6.2 Performance by Video Characteristics

Table 3 breaks down topic accuracy by video duration and editing pace. Accuracy improves monotonically with cut frequency (Low: 81.9%, Medium: 86.7%, High: 89.3%), confirming the cascade is particularly effective at capturing semantically distinct frames in fast-cut content where uniform sampling most often misses critical transitions. The non-monotonic pattern with duration—peaking at medium-length videos (88.6%) and declining for both short (≤20s, 82.6%) and extended (>75s, 82.2%) videos—likely reflects that very short ads provide insufficient context while very long ads introduce more diverse content harder to classify into a single topic.

Binning by cut frequency tertiles further reveals compression ratios of 100.6×, 53.8×, and 34.6× for Low, Medium, and High cut-frequency videos respectively. This 2.9× spread demonstrates that the adaptive budget responds appropriately to video complexity: static testimonials receive aggressive compression while rapid montages are preserved in detail. ANOVA confirms a significant effect of cut frequency on compression ratio ($p < 0.05$).

Table 3: Topic accuracy by video duration and cut frequency (AdaFrame, $N = 500$). Accuracy improves monotonically with cut frequency, confirming the cascade is most effective on fast-cut content where uniform sampling is most wasteful.

Category	Accuracy	N
By Duration		
Short (≤20s)	82.6%	46
Medium (20–45s)	88.6%	220
Long (45–75s)	85.4%	144
Extended (>75s)	82.2%	90
By Cut Frequency		
Low (<0.17 cuts/s)	81.9%	166
Medium (0.17–0.47 cuts/s)	86.7%	166
High (>0.47 cuts/s)	89.3%	168

6.3 Ablation Studies

Cascade Tier Contributions. Table 2 provides indirect ablation by comparing methods approximating individual cascade components. Adding hash/LPIPS pre-filtering and TA-NMS over CLIP Only yields +2.3pp topic accuracy (83.7% vs. 86.0%) and +7.8pp sentiment accuracy (71.6% vs. 79.4%) with 67% fewer frames (74.1 vs. 24.6): pre-filtering reduces CLIP input by ∼40%, while TA-NMS ensures surviving frames span the video's temporal extent rather than clustering in visually dominant segments. Comparing Histogram to AdaFrame, semantic clustering uses 4.2× fewer frames for +4.8pp topic accuracy and +8.9pp sentiment accuracy, demonstrating the value of embedding-based deduplication over pixel-level change detection. The qualitative impact is shown in Figure 5: for a fast-cut gaming ad, Uniform-1FPS over-represents a repeated title card (4 of 8 frames are near-identical), while AdaFrame allocates one frame per semantic segment; for a slow-paced testimonial, AdaFrame collapses 30 frames to 12 while retaining the brand reveal and closing message.

Table 2: Comparison of 10 frame selection methods on 500 video advertisements. AdaFrame reduces frame count by 48% and VLM cost by 75% vs. Uniform-1FPS while achieving the highest topic accuracy (86.0%) and sentiment accuracy (79.4%, +7.5pp over Uniform-1FPS). Sentiment is measured via text-embedding cosine similarity against the 30-class Pitt Ads taxonomy. N varies due to VLM context-limit failures; common-subset results consistent. Best in each column bolded.

Method	N	Frames	Cost ($/video)	Total Cost ($)	Topic Acc. (%)	Super-cat (%)	Sentiment (%)	Eff. Score
AdaFrame (Ours)	500	24.6	**0.0029**	**1.43**	**86.0** [82.7–88.8]	**89.8** [86.8–92.2]	**79.4** [75.6–82.7]	**4.33**
Uniform-1FPS	436	47.7	0.0115	5.74	84.4 [80.7–87.5]	87.8 [84.4–90.6]	71.9 [67.7–75.8]	4.20
Random	482	24.4	0.0099	4.97	78.8 [75.0–82.2]	84.0 [80.5–87.0]	68.9 [64.7–72.9]	4.09
Histogram	409	102.7	0.0161	8.06	81.2 [77.1–84.7]	85.1 [81.3–88.2]	70.5 [66.2–74.5]	4.14
ORB	420	98.6	0.0155	7.75	83.8 [80.0–87.0]	88.1 [84.6–90.9]	73.4 [69.2–77.2]	4.13
Optical Flow	420	188.3	0.0230	11.50	81.0 [76.9–84.4]	84.8 [81.0–87.9]	69.0 [64.7–73.1]	4.16
CLIP Only	398	74.1	0.0136	6.79	83.7 [79.7–87.0]	87.9 [84.4–90.8]	71.6 [67.3–75.5]	4.19
PySceneDetect	452	**20.4**	0.0032	1.47	78.8 [74.8–82.3]	83.2 [79.5–86.4]	65.7 [61.4–69.8]	4.12
DSNet	251	200.1	0.0225	11.25	66.5 [60.5–72.1]	70.5 [64.6–75.8]	60.3 [54.2–66.1]	3.71
Gemini Native	487	—	0.0109	5.45	84.4 [80.9–87.3]	88.7 [85.6–91.2]	71.3 [67.2–75.1]	4.11

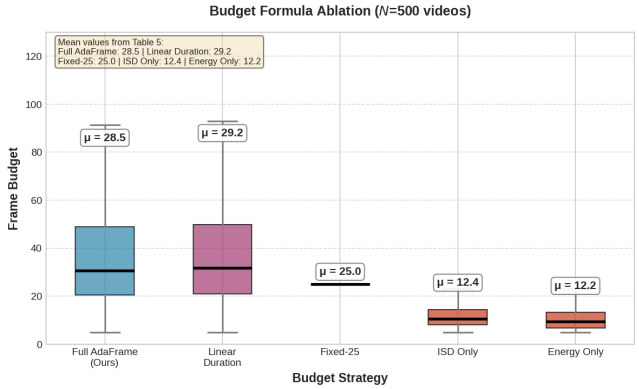


Figure 4: Budget Formula Ablation. Box plots show frame budget distributions across strategies. Full AdaFrame provides the widest dynamic range, while ISD-only and Energy-only under-budget and Fixed-25 has zero adaptivity.

Budget Formula. Figure 4 compares the full budget formula against four simpler alternatives. ISD-only and Energy-only each under-budget (means of 12.4 and 12.2 frames respectively), while Linear Duration over-budgets (29.2) without accounting for content complexity. The full AdaFrame formulation (mean 28.5 frames) provides the widest dynamic range—from 5 frames for highly redundant static ads to over 100 frames for fast-cut content—confirming that the joint ISD + Semantic Energy formulation is necessary for content-proportional allocation. A sensitivity analysis on the ISD expansion multiplier ($\lambda \in \{0.5, 1.0, 1.5, 2.0, 3.0\}$) shows the default $\lambda = 1.5$ is robust: tuning down to 0.5 constricts the dynamic range (26.3 frames) while $\lambda = 3.0$ plateaus at 29.0 frames due to the semantic energy ceiling.

ISD Validation. Figure 6a shows ISD has strong positive correlation with cut frequency ($r = 0.859$, $p < 0.001$, $N = 500$), confirming it captures the primary driver of semantic complexity. ISD also correlates with number of scenes ($r = 0.447$, $p < 0.001$) and shows a small negative correlation with raw duration ($r = -0.202$, $p < 0.001$), indicating that editing pace matters more than length. Figure 6b shows ISD varies significantly by content type: low-motion videos have mean ISD= 2.7 vs. ISD= 7.7 for high-motion videos—a 2.85× adaptivity ratio—confirming the budget mechanism adjusts to content characteristics without manual tuning. Figure 6c validates our choice of $\tau = 0.90$: lower thresholds (0.80, 0.85) produce overly conservative budgets (13–17 frames), while higher thresholds (0.95, 0.99) over-budget (36–64 frames), with no discontinuities across the range. ISD sets an upper bound on frame count; the realized mean of 24.6 frames (Table 2) is lower because TA-NMS may select fewer frames than the budget permits when insufficient high-scoring candidates survive the cascade.

6.4 Provider Generality and Failure Cases

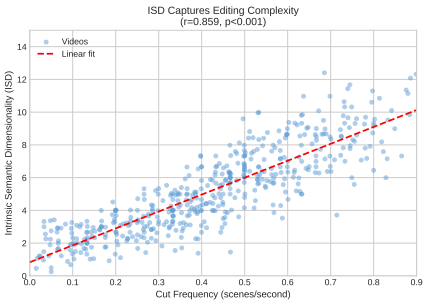
AdaFrame's frame selection is entirely decoupled from the downstream VLM, so the same selected frames can be passed to any provider. Table 4 confirms this: Claude 3 Haiku matches Gemini 2.5 Flash within 0.4pp on topic accuracy (85.6% vs. 86.0%), 0.6pp on super-category accuracy (89.2% vs. 89.8%), and 0.7pp on sentiment (80.1% vs. 79.4%), at comparable cost ($1.51 vs. $1.43). The minor inter-provider variation (<1pp across all dimensions) is well within the range expected from stochastic decoding and prompt sensitivity differences, confirming the pipeline's model-agnostic design.

Table 4: Provider generality results. AdaFrame frame selection is held fixed; only the VLM extraction backend changes. Accuracy metrics are comparable across providers, confirming the cascade's provider-agnostic design.

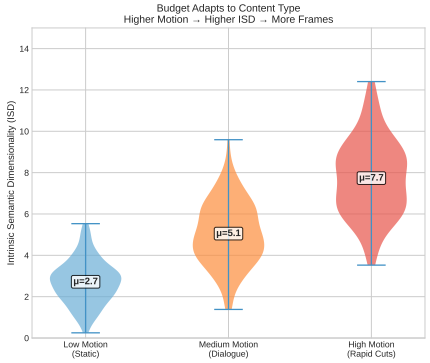
VLM Backend	N	Total Cost ($)	Topic (%)	Super-cat (%)	Sentiment (%)	Eff.
Gemini 2.5 Flash	500	1.43	86.0	89.8	79.4	4.33
Claude 3 Haiku	500	1.51	85.6	89.2	80.1	4.30

Analysis of failure cases reveals that 78% involve subtle visual changes the cascade incorrectly treats as duplicates, with the dominant modes being subtle text changes (42%), rapid logo transitions

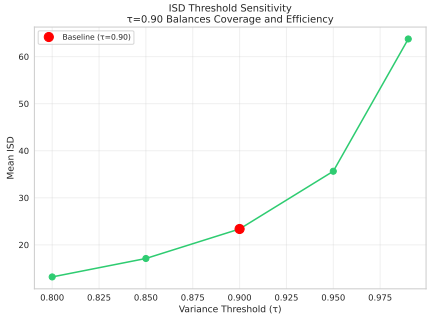
Figure 5: Qualitative Comparison of Sampling Strategies. Top strips for each video show industry-standard Uniform-1FPS sampling (redundant frames marked in red), while bottom strips show AdaFrame's adaptive selection. Our method preserves unique shots while dropping visual duplicates.



(a) ISD vs. Cut Frequency ($N = 500$). Strong positive correlation ($r = 0.859$, $p < 0.001$) validates that ISD captures semantic complexity driven by editing pace. Videos with higher cut frequency receive larger frame budgets automatically.



(b) Content-Type Adaptivity. Violin plots show ISD distributions by motion category. The budget adapts 2.85× between low-motion (μ=2.7) and high-motion (μ=7.7) content.



(c) ISD Threshold Sensitivity. Mean ISD across variance thresholds (τ). $\tau = 0.90$ (marked) provides optimal balance, yielding mean ISD= 23.4 (the computed budget ceiling; realized frame counts are lower due to TA-NMS selection).

Figure 6: ISD Validation.

(18%), color grading shifts (12%), and micro-expressions (6%). The primary mitigations are OCR-aware deduplication for text-critical frames and optical flow integration for sub-second temporal cues.

7 Discussion and Conclusion

We presented AdaFrame, a cost-efficient video analysis framework that reduces VLM inference costs by 75% while improving extraction quality through adaptive frame selection. Our three-tier deduplication cascade eliminates visual redundancy at progressively finer granularity, while ISD enables content-aware budget allocation that responds to each video's structural complexity. On 500 advertisement videos, AdaFrame achieves 86.0% topic accuracy, 79.4% sentiment accuracy (+7.5pp over uniform sampling), and 4.33 effectiveness at 48% of the frame count and 25% of the cost—with gains confirmed model-agnostically across Gemini 2.5 Flash and Claude 3 Haiku. The sentiment result is particularly significant: intelligent frame selection can improve—not merely preserve—downstream extraction quality by providing the VLM with semantically diverse rather than temporally uniform affective context.

The core insight is that editing complexity correlates strongly with semantic information density ($r = 0.859$), enabling automatic budget adaptation without manual tuning. Limitations include occasional loss of text-critical frames that appear visually similar to prior frames, CLIP embedding overhead in ISD computation, and evaluation restricted to short-form ads (mean 38s). Future work will address these through OCR-aware deduplication, distilled embedding models, and extension to long-form content. **Broader Impact:** AdaFrame enables smaller organizations to deploy sophisticated video analysis without prohibitive cloud costs; we encourage responsible use, particularly regarding privacy-sensitive data.

References

[1] Brandon Castellano. Pyscenedetect: Intelligent scene cut detection and video splitting tool. 2018. Version 0.6.

[2] Zuzana Cernekova, Nikolas Nikolaidis, and Ioannis Pitas. Entropy-based shot boundary detection. In *IEEE International Conference on Multimedia and Expo*, pages 1117–1120, 2006.

[3] Yifei Chen, Weiyun Huang, Tianyu Li, et al. Tokenpacker: Efficient visual projector for multimodal llms. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2024.

[4] Priyabrata Das and Sanjit Kumar Choudhury. Video summarization using cluster-based keyframe extraction. *International Journal of Computer Applications*, 32(7): 1–6, 2011.

[5] Emily L Denton, Wojciech Zaremba, Joan Bruna, Yann LeCun, and Rob Fergus. Exploiting linear structure within convolutional networks for efficient evaluation. In *Advances in Neural Information Processing Systems*, volume 27, 2014.

[6] Charles Fefferman, Sanjoy Mitter, and Hariharan Narayanan. Testing manifold hypothesis. *Journal of the American Mathematical Society*, 29(4):983–1049, 2016.

[7] Alan Hanjalic and Ya-Qin Xu. A unified framework for content-based video analysis. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(5): 602–615, 2004.

[8] Tariq Hussain, Ziming Zhang, Ming Zhang, et al. Automatic understanding of image and video advertisements. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1704–1712, 2017.

[9] Henryk Krawczyk. Perceptual hash functions for image identification. *Journal of Telecommunications and Information Technology*, 2015(2):45–52, 2015.

[10] Woosuk Kwon, Sehoon Kim, Michael W Mahoney, et al. Efficient vlm inference through kv cache reuse. In *Neural Information Processing Systems*, 2023.

[11] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

[12] OpenAI. Gpt-4v(ision) system card. *Technical Report*, 2023.

[13] Dan Potapov, Fabian Saleh, Si Liu, Xinchao Wang, Chuah Seng Foo, Truyen Nguyen, Minh Do, and Steven CH Hoi. Category-specific video summarization. In *European Conference on Computer Vision*, pages 540–555. Springer, 2014.

[14] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *International Conference on Machine Learning*, pages 8748–8763, 2021.

[15] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. *arXiv preprint arXiv:2212.04356*, 2022.

[16] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pages 3982–3992, 2019.

[17] Manoj Rochan, Ye Wang, Jianfeng Chen, et al. Task-aware video summarization for question answering. In *ACM International Conference on Multimedia*, pages 2345–2353, 2023.

[18] Aman Singh, Prateek Jain, and Ankit Gupta. Temporal attention for video summarization. In *IEEE International Conference on Image Processing*, pages 1123–1127, 2022.

[19] Ray Smith. An overview of the tesseract ocr engine. In *IEEE International Conference on Document Analysis and Recognition*, volume 2, pages 629–633, 2007.

[20] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Robert Dadashi, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[21] Mohan M Trivedi and Tobias Hollerer. Multi-cue shot boundary detection. In *IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 387–392, 2007.

[22] Xiaoyu Wang, Wei Zhang, and Yiqiang Liu. Robust wavelet-based perceptual hashing. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2482–2486, 2019.

[23] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[24] Christoph Zauner. A framework for evaluating perceptual image hashing. *Master's Thesis, University of Applied Sciences Technikum Wien*, 2010.

[25] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018.

[26] Bo Zhao, Jiashi Feng, and Shuicheng Yan. Video summarization with self-supervised representation learning. In *IEEE International Conference on Computer Vision*, pages 1196–1205, 2021.

[27] Kaiwei Zhou, Lei Qiao, and Tong Zhang. Deep reinforcement learning for diverse video summarization with transferability. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4767–4776, 2018.