

Lightweight Temporal Attention for Video Retrieval over Frozen Image and Text Encoders

Vidit Khazanchi*

Adobe

Noida, UP, India

vkazanchi@adobe.com

Umang Sharma*

Adobe

Noida, UP, India

umsharma@adobe.com

Shabhareesh Komirishetty

Adobe

San Jose, CA, USA

skomirishetty@adobe.com

Abstract

Large-scale visual search platforms maintain production text-image embedding models trained on behaviorally-mined query-image pairs, where the text encoder is implicitly tuned to organisation-specific search vocabulary. Extending such a system to video retrieval is non-trivial: retraining risks disrupting hard-won query understanding, while naive temporal aggregation discards the sequential structure that distinguishes a door opening from a door closing. We add a lightweight temporal attention pooling head over independently encoded frame embeddings, keeping both encoders frozen and training only the pooling head and positional encodings. We train on a large-scale dataset from a commercial creative platform using bidirectional InfoNCE with a two-stage hard negative mining pipeline. Benchmarking against LoRA fine-tuned Gemma 3 (12B) and SmolVLM2, neither matches our approach despite substantially larger parameter counts. We also introduce an evaluation benchmark with graded relevance judgments annotated by a VLM judge, where each query is mapped to all relevant videos in a large corpus rather than a single ground-truth clip, avoiding protocols that greatly underestimate model quality. Our model improves nDCG@1 by 10.1% over mean-pooling overall and 25% on complex action queries. With Matryoshka Representation Learning, 256-dim embeddings retain 95.6% of full performance, and int8 quantization costs under 1% in retrieval quality.

1 Introduction

Large visual search platforms maintain production text-image embedding models trained on hundreds of millions of behaviorally-mined query-image pairs, with text encoders implicitly tuned to organisation-specific search vocabulary. Extending such systems to video is non-trivial: retraining

the text encoder is prohibitively expensive and risks disrupting alignment with real-user query vocabulary, while the simplest extension—mean pooling over frame embeddings—discards temporal order entirely. Averaging frames of someone picking up a key, walking to a door, and unlocking it yields a vector indistinguishable from the same frames in reverse, making queries like “a person entering a building” and “a person leaving a building” indistinguishable.

We address this by adding a learned temporal attention pooling head on top of the frozen image encoder, with positional encodings to preserve frame order. Both encoders remain frozen throughout, preserving existing text-image alignment and allowing drop-in deployment with no changes to query serving infrastructure. We train with bidirectional InfoNCE (van den Oord et al., 2018) and a two-stage hard negative mining pipeline, and benchmark against LoRA fine-tuned generative alternatives—Gemma 3 (12B) (Gemma Team, 2025a) and SmolVLM2 (Marafioti et al., 2025)—finding that, under matched adaptation on our production stack, these larger decoder-only models do not surpass the lightweight head; we treat this as evidence about this adaptation regime rather than a general scaling law, given the differing backbones. For evaluation, we construct a benchmark via two-stage candidate generation and Qwen2.5-VL-72B (Bai et al., 2025) annotation, producing graded relevance judgments with multiple ground-truth videos per query and avoiding the systematic underestimation of one-to-one protocols.

Our contributions are: (1) a temporal attention pooling architecture that improves nDCG@1 by 10.1% over mean pooling overall and 25% on complex action queries, (2) a two-stage hard negative mining pipeline with BLIP (Li et al., 2022) ITM reranking and margin-based false-negative filtering over 225K training clips, (3) an evaluation benchmark of 400+ queries with pooled graded

* Equal contribution.

relevance judgments across nine categories, and (4) a production-readiness analysis covering Ma-tryoshka embeddings (Kusupati et al., 2022), int8 quantization, and sparse-dense hybrid retrieval on a real-world corpus of 100K videos.

2 Related Work

Video-Text Retrieval. Since CLIP (Radford et al., 2021) established contrastive image-text pretraining at scale, the dominant strategy for video-text retrieval has been to adapt image encoders to video via temporal aggregation over per-frame features. CLIP4Clip (Luo et al., 2022) showed mean pooling is surprisingly competitive for short clips. Frozen in Time (Bain et al., 2021) trained a joint encoder end-to-end treating images as single-frame videos. InternVideo2 (Wang et al., 2024) scaled contrastive and masked pretraining to 6B parameters, Cap4Video (Wu et al., 2023) added auto-generated captions as training signal, and X-Pool (Gorti et al., 2022) explored query-conditioned frame pooling at inference time. Our work differs from all of these: rather than pre-training from scratch, we add a lightweight head on top of an existing production encoder, preserving its retrieval behaviour, a practically important distinction in industrial settings where retraining a production encoder is prohibitive. Parameter-efficient adaptations of frozen CLIP for video—RAP (Cao et al., 2024), Lin et al. (2022), and STAN (Liu et al., 2023a)—insert or adapt modules *within* the backbone, whereas our head sits strictly *on top* of a frozen proprietary encoder with no backbone access, a stricter constraint that enables drop-in deployment. Prior temporal probes and bias analyses—the shuffled-frame tests of SViT (Li et al., 2023a) and the single-frame-bias study of Lei et al. (2023)—motivate our judge-free shuffle/reversal probe (Section 5.4).

Temporal Modeling. I3D (Carreira and Zisserman, 2017), SlowFast (Feichtenhofer et al., 2019), TimeSformer (Bertasius et al., 2021), ViViT (Arnab et al., 2021), and Video Swin (Liu et al., 2022) each tackle spatiotemporal modeling at increasing complexity and parameter cost. TSM (Lin et al., 2019) showed that temporal channel shifting adds zero parameters yet meaningfully improves temporal understanding, VideoMAE (Tong et al., 2022) established that high-ratio tube masking forces strong spatiotemporal representations during pre-training. Our attention pooling head follows TSM’s

lightweight philosophy: minimal added parameters focused on preserving temporal position. Our pooling mechanism—a learnable query attending over frame embeddings as keys and values—is an instance of attention-based set aggregation (Lee et al., 2019), also used for video in X-Pool (Gorti et al., 2022) and CLIP4Clip (Luo et al., 2022); our contribution is not the operator but its use as a drop-in, frozen-encoder head that preserves a production text tower’s alignment, together with the deployment study and graded benchmark.

Hard Negatives in Contrastive Learning. Hard negative mining is well-established in metric learning (Schroff et al., 2015), dense retrieval (Karpukhin et al., 2020; Xiong et al., 2021), and vision-language pretraining (Radenovic et al., 2023; Ye et al., 2023). False negatives — semantically equivalent pairs mislabelled as negative — are a known failure mode (Chuang et al., 2020; Robinson et al., 2021) that is particularly acute in large video corpus where many clips cover identical scenes. Our margin-based filtering retains a candidate as a hard negative only if its ITM similarity to the anchor falls below the positive-pair score by a fixed margin, directly addressing this.

Flexible Embeddings and Sparse Retrieval. MRL (Kusupati et al., 2022) jointly optimises nested embedding sub-vectors, enabling one model to serve multiple dimensionalities without separate projection heads. SPLADE (Formal et al., 2021) and ColBERT (Khattab and Zaharia, 2020) offer complementary sparse and late-interaction trade-offs, we explore the former via a Residual Autoencoder baseline. To our knowledge, ours is the first application of MRL to video embeddings.

LLM-Based Evaluation. LLM-as-judge evaluation (Zheng et al., 2023; Liu et al., 2023b) is now widely used for open-ended tasks. We use Qwen2.5-VL-72B (Bai et al., 2025) to assign continuous relevance scores in $[0, 1]$, thresholded at 0.85 to form ground-truth sets, following graded judgment protocols from TREC Deep Learning (Craswell et al., 2021) that produce more stable system rankings than binary qrels — particularly important when a corpus contains many near-duplicate relevant documents.

3 Method

3.1 Problem Formulation

We want to learn functions f_V and f_T that embed videos and text queries into a shared metric space $\mathbb{R}^d$ such that $\text{sim}(f_V(V), f_T(q))$ is high when V is a good retrieval result for query q and low otherwise. A video V is represented as a sequence of N frames $\{I_1, \dots, I_N\}$ sampled at 2 FPS up to a maximum of 120 frames, covering up to 60 seconds of footage. Both functions produce L2-normalised vectors with $d = 768$ by default.

3.2 Architecture

The architecture (Fig. 1) is intentionally minimal: encode frames independently, add positional awareness, and learn to attend.

We build on two components of an in-house production text-image embedding system trained on hundreds of millions of behaviorally-mined query-image pairs from a large commercial creative platform: PROVE (Production Visual Encoder), a ViT-L/14-based image encoder, and PROTE (Production Text Encoder), its paired text tower implicitly tuned to organisation-specific search vocabulary. Both are kept frozen in all our experiments, and we refer to them by these names throughout the paper.

Given frames $\{I_1, \dots, I_N\}$, we compute per-frame embeddings using the frozen PROVE:

$$e_i = f_{\text{PROVE}}(I_i) \in \mathbb{R}^d, \quad i = 1, \dots, N \quad (1)$$

Learnable positional encodings $p_i \in \mathbb{R}^d$ are added to each frame embedding to preserve temporal order:

$$\tilde{e}_i = e_i + p_i \quad (2)$$

These position-augmented embeddings pass through L transformer encoder layers and are compressed to a single L2-normalised vector by an attention pooling step:

$$v = \text{normalize}(\text{AttnPool}(\{\tilde{e}_1, \dots, \tilde{e}_N\})) \in \mathbb{R}^d \quad (3)$$

We call it attention pooling as learnable query vectors attend over frames as keys/values to compress T frames into a single vector $[B, D]$, where B is batch size and D is the embedding dimension. The temporal ordering is injected via positional encodings, and the attention itself is used for intelligent aggregation.

The frozen PROTE produces query embeddings in the same space as v . Keeping both PROVE and

PROTE frozen ensures the video embedding never drifts away from the established text-image alignment, and makes the system a drop-in extension to existing retrieval infrastructure with no changes to query serving. We train with bidirectional InfoNCE (van den Oord et al., 2018), summing symmetric video-to-text and text-to-video losses over in-batch negatives augmented with the hard negatives described in Section 4.2, using a learnable temperature initialised to $\tau = 0.07$. We compare our temporal attention pooling head against the following alternative aggregation strategies: mean pooling over frame embeddings (the production baseline), video thumbnail embedding, and two generative alternatives—SmolVLM2 (2.2B parameters) and Gemma 3 (12B parameters)—each LoRA fine-tuned to produce fixed-size embeddings. These generative baselines are LoRA fine-tuned on the language decoder only, with all other weights frozen, using the same bidirectional InfoNCE loss as our model. For text, query tokens are passed directly to the decoder, for video, frame embeddings flow through the vision encoder and cross-modal projector before the decoder, with the final hidden state L2-normalised as the embedding in both cases. The generative models produce 3072-dimensional embeddings versus 768 for PROVE-based variants, at substantially higher inference cost and no retrieval gain.

3.3 Matryoshka Representation Learning

After the base model is fully trained at $d = 768$, we continue from the best checkpoint using MRL (Kusupati et al., 2022) over a nested dimension set $\mathcal{D} = \{768, 512, 256\}$:

$$\mathcal{L}_{\text{MRL}} = \frac{1}{|\mathcal{D}|} \sum_{d \in \mathcal{D}} \mathcal{L}_{\text{InfoNCE}}(v_{1:d}, t_{1:d}) \quad (4)$$

where $v_{1:d}$ is the first d dimensions of the full embedding. This jointly optimises all three dimensionalities in a single training pass, forcing the model to pack the most salient temporal information into the earliest dimensions. At inference, any prefix of the embedding is usable directly without a separate projection head.

4 Datasets and Evaluation Setup

4.1 Training Data

We train on roughly 225K videos drawn from a commercial creative video platform: around 213K commercial clips and around 12K generated clips

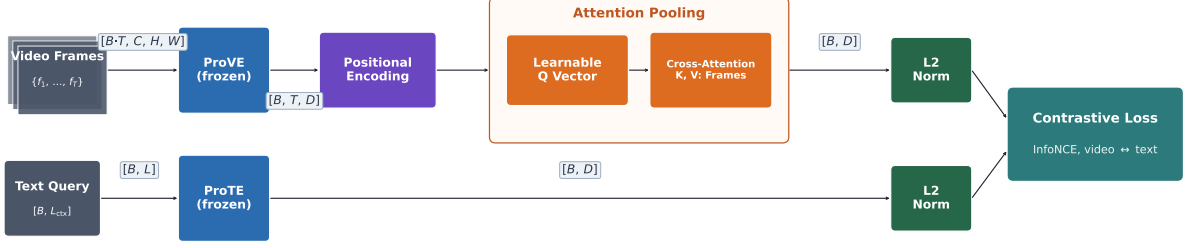


Figure 1: Architecture overview. Each video frame is independently encoded by the frozen PROVE image encoder. Learned positional encodings (PE) are added to preserve temporal order. A cross-attention pooling head then aggregates the frame sequence into a single video embedding $\mathbf{v}$: a learnable query vector $\mathbf{Q}$ attends over the frame embeddings as keys $\mathbf{K}$ and values $\mathbf{V}$, where $\mathbf{Q}$ is a trained parameter independent of the input while $\mathbf{K}$ and $\mathbf{V}$ are derived from the encoded frames. The frozen PROTE text encoder maps the query to embedding $\mathbf{t}$ in the same space. Bidirectional InfoNCE aligns the two embedding spaces during training.

from a text-to-video generation model on the same platform, each up to one minute long. Frames are sampled at 2 FPS, giving up to 120 frames per video. For each video, we generate annotations using Gemma 3n (Gemma Team, 2025b) in a two-stage pipeline: a dense complex_summary capturing action, scene, mood, and cinematography, followed by structured extraction of short prompts, multi-clause complex_prompts, intents, objects, and style, which serve as positive text for contrastive training. See Appendix A for some samples of the dataset.

4.2 Hard Negative Mining

Random in-batch negatives are easy. A random video from a library of hundreds of thousands is almost certainly not semantically similar to a given anchor, so the model learns nothing from it once basic categories are separated. We need negatives that are close enough to the anchor to force the model to learn fine-grained distinctions. Our pipeline runs in two stages (Fig. 2).

Stage 1: Dense Candidate Retrieval. For each anchor, we retrieve 128 candidate negatives from the training corpus using a fused embedding:

$$q_{\text{fused}} = [\alpha \cdot \text{norm}(u_T) \parallel \alpha \cdot \text{norm}(u_V)], \quad \alpha = 0.5 \quad (5)$$

where u_T comes from GTE-large-en-v1.5 (Li et al., 2023b) encoding the video’s complex_summary described in Section 4, and u_V comes from SigLIP-base-patch16 (Zhai et al., 2023) encoding a 60-frame grid of the video. We search a FAISS (Johnson et al., 2021) IndexFlatIP index over the full training video corpus.

Stage 2: Cross-Modal Reranking and Margin Filtering. The 128 candidates are reranked

with BLIP (Li et al., 2022) Image-Text Matching (ITM) scores. For anchor A (text T_A , video V_A) and candidate C (text T_C , video V_C), we compute:

$$S(A, C) = \frac{1}{2} [S_{\text{t2i}}(T_A, V_C) + S_{\text{i2t}}(T_C, V_A)] \quad (6)$$

The symmetric formulation catches asymmetric cases (two clips covering similar but not identical content) that a one-directional score would miss. We then apply a margin filter: a candidate is retained only if

$$S(A, C) < S_{\text{pos}}(A) - m, \quad m = 0.15 \quad (7)$$

where $S_{\text{pos}}(A)$ is the positive-pair ITM score for anchor A . The top-8 retained candidates become the hard negatives for that anchor. The margin prevents videos covering the same scene from being pushed apart during training.

4.3 Benchmark Construction

Standard evaluation practice links each query to exactly one correct video, which systematically underestimates quality on a large video corpus where dozens of clips are equally valid answers for a query like “a dog running in a park.” Retrieving the right kind of video but the wrong specific clip counts as a complete failure, consistently underestimating the quality of any reasonably good model.

Our benchmark contains 400+ queries covering nine semantic categories: test set queries (100), followed by synthetically generated complex multi-clause queries (100), action/activity recognition (50), subject/object detection (40), scene and setting recognition (40), cinematography (40), emotion and expression (25), theme and topic (25), and

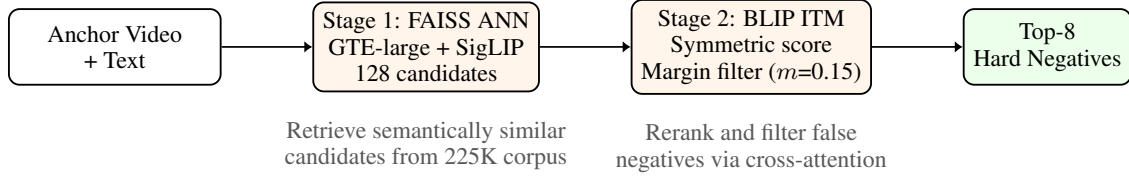


Figure 2: Two-stage hard negative mining pipeline. Stage 1 retrieves 128 candidate negatives per anchor using a fused text+vision query against the full training corpus via FAISS. Stage 2 reranks candidates with BLIP image-text matching and discards any that exceed the positive-pair score minus margin m , preventing semantically equivalent videos from being treated as negatives.

Table 1: Overall retrieval on the Golden Benchmark (14K test set). Bold: best per metric. @ K denotes nDCG@ K .

Model	@1	@3	@5	@10
ProVE Mean Pool	0.7729	0.7503	0.7372	0.7149
SmolVLM2-2.2B (LoRA)	0.6539	0.6305	0.6123	0.5929
Gemma3-12B (LoRA)	0.7282	0.6999	0.6870	0.6734
ProVE + Attn Pool (ours)	0.8509	0.8241	0.8122	0.7970

visual attributes (5). The categories are designed to stress-test specific aspects of video understanding.

For each query, we collect the top-100 videos retrieved across several different candidate text-to-video retrievers, forming a candidate pool. We then run Qwen2.5-VL-72B (Bai et al., 2025) on every (query, video) pair, asking the model to assign a relevance score in $[0, 1]$. Any video scoring above 0.85 is included in that query’s ground-truth set, with the score used as a relevance grade in the nDCG computation.

5 Experiments

5.1 Results on the Golden Evaluation Benchmark

Table 1 shows overall performance on the golden benchmark (14K test videos). Temporal attention pooling improves nDCG@1 by 10.1% over mean pooling. The two generative model baselines, SmolVLM2 (2.2B) and Gemma 3 (12B), both fall below mean pooling despite having far more parameters. Gemma 3 alone produces 3072-dimensional embeddings versus 768 for ProVE variants, at substantially higher inference cost and no retrieval gain. We read this as a statement about the adaptation regime—decoder-only LoRA with last-hidden-state pooling is not a competitive embedding recipe here—rather than a general scaling law, since backbone and adaptation differ across these models. See Appendix B for a detailed qualitative analysis of these results.

Table 2: Per-category nDCG@1 on the Golden Benchmark. Gain is relative to mean pooling. The nine benchmark categories omit two here: the general *test-set queries* bucket and *visual attributes* ($n=5$, too small for a stable per-category estimate; reported as indicative only in Appendix F).

Category	Mean Pool	Attn Pool	Gain
Complex Queries	0.5572	0.6965	+25.0%
Subject/Object Detect.	0.7890	0.9150	+16.0%
Scene/Setting Recogn.	0.8426	0.9588	+13.8%
Emotion/Expression	0.8675	0.9591	+10.6%
Action/Activity	0.8182	0.8713	+6.5%
Theme/Topic	0.8904	0.9213	+3.5%
Cinematography	0.8566	0.8714	+1.7%

5.2 Category-Level Analysis

Table 2 breaks down results by query category. The gains are largest where temporal ordering matters most. Complex multi-clause queries gain 25.0% in nDCG@1. Subject/object detection gains 16.0%, likely because attention can focus on the frames where a particular object is most visible rather than averaging its presence across the full clip. Scene and setting recognition gains 13.8%.

Cinematography queries (those specifying camera behavior like “a tracking shot” or “slow motion”) show the smallest gain at 1.7%. Identifying camera motion from independently encoded 2-FPS frames is harder than identifying scene content: subtle motion blur and camera trajectory information is largely lost at that sampling rate.

5.3 Comparison Against Learned Aggregation Baselines

In addition to other baselines, we compare against two strong learned aggregators built on the identical frozen PROVE frame features, trained with the same data, bidirectional-InfoNCE loss, and hard-negative pipeline, so that only the aggregation mechanism differs: a CLIP4Clip-seqTransf

Table 3: Aggregation heads over identical frozen PROVE features. @ K = nDCG@ K ; * = improvement over mean pooling significant at $p_{\text{holm}} < 0.05$ (paired randomization test). Computed on the statistics re-run described in Appendix F; full CIs and p -values there.

Head (frozen PROVE)	@1	@3	@5	@10
Mean pooling (baseline)	0.786	0.762	0.748	0.726
CLIP4Clip-seqTransf	0.773	0.746	0.730	0.707
X-Pool (reranker)	0.774	0.766	0.761	0.753
ProVE Attn pool, -PE	0.814	0.792	0.783	0.762*
ProVE Attn pool, $L=4$	0.838	0.808*	0.796*	0.780*
ProVE Attn pool, $L=2$	0.845	0.820*	0.807*	0.795*
ProVE Attn pool, $L=1$ (ours)	0.851	0.824*	0.812*	0.797*

sequence Transformer (Luo et al., 2022) and query-conditioned X-Pool (Gorti et al., 2022). We also vary the pooling depth L in our architecture.

As shown in (Table 3), CLIP4Clip sits at or below mean pooling across cutoffs; X-Pool trails mean pooling at nDCG@1 but edges above it at nDCG@3,5,10. Both stay well below our attention head. X-Pool pools each video conditioned on the query, so it has no static per-video embedding and cannot be indexed once for ANN retrieval; we evaluate it as a two-stage reranker, its only deployable form for this use case (Appendix G). Increasing pooling depth gives no benefit ($L=1 \geq L=2 \geq L=4$).

5.4 Isolating Temporal Order from Frame Selection

Attention pooling could outperform mean pooling for two distinct reasons: it may learn to *select* the most informative frames (an order-invariant effect), or it may exploit *temporal order* through the positional encodings. We disentangle a positional-encoding (PE) ablation and a judge-free order-perturbation probe. Every model here is trained on the identical frozen PROVE frame embeddings and frozen PROTE, with the same training data, bidirectional-InfoNCE loss, and hard-negative pipeline, so any difference is attributable to the aggregation mechanism alone.

Positional-encoding ablation. Removing the positional encodings leaves only learned frame selection and makes the head permutation-invariant also evident from drop in nDCG shown in (Table 3). This retains just +0.029 nDCG@1 over mean pooling (vs. +0.065 for the full head) and is not significant after Holm correction at nDCG@{1,3}, whereas the full model is significant at nDCG@{3,5,10} ($p_{\text{holm}}=0.040$). Posi-

Table 4: Retrieval on 100K test videos set with user behavioral queries. @ K denotes nDCG@ K .

Model	@1	@3	@5	@10
ProVE Thumbnail	0.7928	0.7839	0.7816	0.7645
ProVE Mean Pool	0.7824	0.7611	0.7430	0.7237
ProVE Attn Pool	0.8463	0.8360	0.8218	0.8119

tional encoding therefore accounts for $\approx 56\%$ of the nDCG@1 gain, localizing the statistically robust portion of the improvement to temporal ordering rather than frame selection. Per-cutoff CIs, Δ intervals, and p -values for all models are reported in Appendix F.

Frame-shuffle/reversal probe (judge-free). For 200 test queries and their top-10 retrieved videos ($\sim 2,000$ query-video pairs), we recompute query-video cosine similarity under (i) random frame shuffling and (ii) full frame reversal. Mean pooling is permutation-invariant and unchanged by construction; our head’s similarity drops under shuffling in 62.0% of cases and under reversal in 63.5%, with the largest drops on strongly order-dependent queries. The non-degrading cases are queries for which order is genuinely irrelevant (static scenes, or content recognizable regardless of playback direction, e.g. “time-lapse of ice melting in a glass”).

5.5 Evaluation on User Behavioral Queries

We also evaluated all three ProVE variants on 100K unseen videos from the same commercial creative video platform, using 1K real user queries from that platform’s search logs, having a mix of high, medium, and low frequency queries. Table 4 includes the thumbnail baseline alongside the mean pooling and attention pooling. Temporal attention leads by a clear margin. Notably, thumbnail embeddings (0.7928) slightly outperform mean pooling (0.7824) in overall nDCG@1, an inversion of what one might expect, given that thumbnails encode a single frame. This suggests that for certain query types, averaging across many frames introduces more redundancy than signal, making a single well-chosen frame more effective.

5.6 Matryoshka Embedding Analysis

Table 5 shows the dimension-performance trade-off after MRL fine-tuning. The 512-dimensional model retains 98.6% of the full 768 nDCG@1. The 256-dimensional model retains 95.6%. More striking is the cross-model comparison: the 256-

Table 5: MRL dimension-performance on the Golden Benchmark. @ K denotes nDCG@ K .

Model	@1	@3	@5	@10
ProVE Mean Pool [768]	0.7729	0.7503	0.7372	0.7149
ProVE Attn Pool [768]	0.8509	0.8241	0.8122	0.7970
ProVE Attn Pool [512]	0.8386	0.8122	0.8018	0.7854
ProVE Attn Pool [256]	0.8135	0.7772	0.7619	0.7487

Table 6: int8 vs. full-precision (float32) *embeddings* on Express 100K. @ K denotes nDCG@ K .

Model	Fmt	@1	@3	@5	@10
Thumbnail	fp32	0.7928	0.7839	0.7816	0.7645
Thumbnail	int8	0.7885	0.7819	0.7772	0.7610
Mean Pool	fp32	0.7824	0.7611	0.7430	0.7237
Mean Pool	int8	0.7813	0.7536	0.7416	0.7188
Attn Pool	fp32	0.8463	0.8360	0.8218	0.8119
Attn Pool	int8	0.8389	0.8297	0.8205	0.8099

dim attention-pooled model (0.8135) still beats the full 768-dim mean-pooling baseline (0.7729) by a wide margin. Temporal aggregation quality matters more than embedding capacity for this task. With MRL, a deployment that needs smaller indexes or faster ANN search can use 256-dimensional embeddings without falling below the quality of the production baseline.

5.7 Deployment Analysis

Embedding quantization. Storing tens of millions of float32 embeddings is expensive. Table 6 shows that quantising attention-pooled embeddings from float32 to int8 costs only 0.74% drop in nDCG@1. All three model variants show similarly negligible degradation.

Sparse-dense hybrid retrieval. We evaluate sparse-dense hybrid retrieval using a Residual Autoencoder (RAE) that projects 768-dim dense embeddings into a 16,384-dim sparse space suitable for inverted-index retrieval. Table 7 compares three modes: pure dense (FAISS ANN), pure sparse RAE, and a hybrid that uses RAE to recall the top-100 candidates and reranks by dense cosine similarity. Pure RAE alone falls 6.6% below pure dense, but the hybrid closes most of that gap and marginally outperforms pure dense (0.8497 vs. 0.8463 nDCG@1). This is relevant for production deployments at scale where pure ANN search over dense vectors incurs significant latency and infrastructure cost, inverted indexes offer well-understood operational advantages in such settings,

Table 7: Dense vs. sparse RAE vs. hybrid retrieval on Express 100K. @ K denotes nDCG@ K .

Mode	@1	@3	@5	@10
Pure Dense	0.8463	0.8360	0.8218	0.8119
Pure RAE	0.7749	0.7401	0.7311	0.7106
Hybrid ($N=100$)	0.8497	0.8334	0.8193	0.8094

and the hybrid recovers nearly all the retrieval quality of pure dense search. In this work, we reuse a sparse RAE trained over PROVE embeddings without any additional fine-tuning, suggesting the approach generalises across retrieval heads built on the same backbone.

5.8 Production Deployment

The model is deployed as a chain of microservices coordinated by a central orchestrator. On video ingestion, the pipeline downloads the video, extracts frames at 2 FPS, encodes each frame independently via PROVE across a parallel fanout, and passes the resulting embeddings through the temporal attention aggregator to produce a single dense vector and optional sparse projection for inverted-index retrieval. Both embeddings are written to the search index, making the video immediately retrievable. Appendix D provides a detailed capacity analysis showing that this pipeline can index a corpus of 100M videos within 7 days.

6 Conclusion

We showed that a lightweight temporal attention pooling head trained on top of frozen production encoders (PROVE and PROTE) delivers substantial gains for large-scale video-text retrieval without modifying any pretrained weights. Our model improves nDCG@1 by 10.1% overall and by 25% on complex multi-clause action queries. Matryoshka training retains 95.6% of full performance at 256 dimensions, and int8 quantization costs under 1% in retrieval quality, making the system deployable across a range of infrastructure budgets.

Several directions remain open. A promising one is replacing independent frame encoding with a full spatiotemporal attention layer—attending jointly across spatial patches and the temporal axis—to capture fine-grained motion and object interactions that frame-level pooling misses. Further directions, including event-ordering-aware hard negatives, higher-frame-rate sensing, and multilingual fine-tuning, are discussed in Appendix J.

Limitations

Proprietary training data. Our model is trained and evaluated on an internal dataset from a commercial creative platform, which limits direct reproducibility and comparison with publicly available video retrieval benchmarks such as MSR-VTT or ActivityNet.

Evaluation benchmark scope. Our evaluation benchmark comprises 400+ queries drawn from a single platform’s search logs, annotated by a single VLM judge (Qwen2.5-VL-72B). While graded relevance judgments are increasingly reliable, the benchmark is not publicly released, and the annotation process is subject to the temporal reasoning limitations and potential self-enhancement bias of the judge model and candidate retrievers.

Fixed frame sampling rate. Frames are sampled at a fixed 2 FPS, which discards fine-grained motion information. This is the primary reason cinematographic queries show the smallest gains, and it limits the model’s ability to distinguish subtle temporal patterns such as camera motion or fast action sequences. However for a semantic search platform, which is also powered by personalized and deep neural rankers, these limitations are acceptable.

English-centric training. The temporal attention head is trained exclusively on English query-video pairs. Our multilingual evaluation (Appendix C) therefore reflects the generalisation of a separately trained multilingual text encoder rather than multilingual temporal understanding per se, limiting conclusions about how well the pooling head itself transfers across languages.

Single-encoder dependency. The entire approach is contingent on the quality of the underlying production encoders (PROVE and PROTE). Organisations without a strong pretrained text-image encoder trained on domain-specific behaviorally-mined data may not observe the same gains, limiting the generalisability of the frozen-encoder design choice.

References

Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lucic, and Cordelia Schmid. 2021. [ViViT: A video vision transformer](#). In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6816–6826. IEEE.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Ming-Hsuan Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025. [Qwen2.5-vl technical report](#). *CoRR*, abs/2502.13923.

Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. 2021. [Frozen in time: A joint video and image encoder for end-to-end retrieval](#). In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1708–1718.

Gedas Bertasius, Heng Wang, and Lorenzo Torresani. 2021. [Is space-time attention all you need for video understanding?](#) In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pages 813–824. PMLR.

Meng Cao, Haoran Tang, Jinfa Huang, Peng Jin, Can Zhang, Ruyang Liu, Long Chen, Xiaodan Liang, Li Yuan, and Ge Li. 2024. [RAP: efficient text-video retrieval with sparse-and-correlated adapter](#). In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, volume ACL 2024 of *Findings of ACL*, pages 7160–7174. Association for Computational Linguistics.

João Carreira and Andrew Zisserman. 2017. [Quo vadis, action recognition? a new model and the kinetics dataset](#). In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733.

Ching-Yao Chuang, Joshua Robinson, Yen-Chen Lin, Antonio Torralba, and Stefanie Jegelka. 2020. [Debiased contrastive learning](#). In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 8765–8775. Curran Associates, Inc.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Jimmy Lin. 2021. [Overview of the TREC 2021 deep learning track](#). In *Proceedings of the Thirtieth Text REtrieval Conference (TREC 2021)*, online, November 15-19, 2021, volume 500-335. National Institute of Standards and Technology (NIST).

Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. 2019. [SlowFast networks for video recognition](#). In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 6201–6210.

Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2021. [SPLADE: Sparse lexical and expansion model for first stage ranking](#). In *SIGIR ’21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, pages 2288–2292. ACM.

- Gemma Team. 2025a. [Gemma 3 technical report](#). *arXiv preprint arXiv:2503.19786*.
- Gemma Team. 2025b. Gemma 3n. Google DeepMind. Model card, Hugging Face. <https://huggingface.co/google/gemma-3n-E4B-it>.
- Satya Krishna Gorti, Noel Vouitsis, Junwei Ma, Keyvan Golestan, Maksims Volkovs, Animesh Garg, and Guangwei Yu. 2022. [X-pool: Cross-modal language-video attention for text-video retrieval](#). In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4996–5005.
- Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2021. [Billion-scale similarity search with GPUs](#). *IEEE Transactions on Big Data*, 7(3):535–547.
- Vladimir Karpukhin, Barlas Öğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. [Dense passage retrieval for open-domain question answering](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6769–6781. Association for Computational Linguistics.
- Omar Khattab and Matei Zaharia. 2020. [ColBERT: Efficient and effective passage search via contextualized late interaction over BERT](#). In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 39–48, New York, NY, USA. Association for Computing Machinery.
- Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham Kakade, Prateek Jain, and Ali Farhadi. 2022. [Ma-tryoshka representation learning](#). In *Advances in Neural Information Processing Systems*, volume 35, pages 30233–30249. Curran Associates, Inc.
- Juho Lee, Yoonho Lee, Jungtaek Kim, Adam R. Kosiosek, Seungjin Choi, and Yee Whye Teh. 2019. [Set transformer: A framework for attention-based permutation-invariant neural networks](#). In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pages 3744–3753. PMLR.
- Jie Lei, Tamara Berg, and Mohit Bansal. 2023. [Revealing single frame bias for video-and-language learning](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 487–507.
- Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. [BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation](#). In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, volume 162, pages 12888–12900. PMLR.
- Yi Li, Kyle Min, Subarna Tripathi, and Nuno Vasconcelos. 2023a. [SViTT: Temporal learning of sparse video-text transformers](#). In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18919–18929.
- Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023b. [Towards general text embeddings with multi-stage contrastive learning](#). *CoRR*, abs/2308.03281.
- Ji Lin, Chuang Gan, and Song Han. 2019. [TSM: Temporal shift module for efficient video understanding](#). In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7082–7092.
- Ziyi Lin, Shijie Geng, Renrui Zhang, Peng Gao, Gerard de Melo, Xiaogang Wang, Jifeng Dai, Yu Qiao, and Hongsheng Li. 2022. [Frozen CLIP models are efficient video learners](#). In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 388–404.
- Ruyang Liu, Jingjia Huang, Ge Li, Jiashi Feng, Xinglong Wu, and Thomas H. Li. 2023a. [Revisiting temporal modeling for CLIP-based image-to-video knowledge transferring](#). In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6555–6564.
- Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023b. [G-Eval: NLG evaluation using GPT-4 with better human alignment](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2511–2522. Association for Computational Linguistics.
- Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. 2022. [Video swin transformer](#). In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3192–3201. IEEE Computer Society.
- Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. 2022. [CLIP4Clip: An empirical study of CLIP for end to end video clip retrieval and captioning](#). *Neurocomputing*, 508:293–304.
- Andrés Marafioti, Orr Zohar, Miquel Farré, Merve Noyan, Elie Bakouch, Pedro Cuenca, Cyril Zakka, Loubna Ben Allal, Anton Lozhkov, Nouamane Tazi, Vaibhav Srivastav, Joshua Lochner, Hugo Larcher, Mathieu Morlon, Lewis Tunstall, Leandro von Werra, and Thomas Wolf. 2025. [Smolvlm: Redefining small and efficient multimodal models](#). *CoRR*, abs/2504.05299.
- Filip Radenovic, Abhimanyu Dubey, Abhishek Kadian, Todor Mihaylov, Simon Vandenhende, Yash Patel, Yi Wen, Vignesh Ramanathan, and Dhruv Mahajan. 2023. [Filtering, distillation, and hard negatives for vision-language pre-training](#). In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6967–6977.

- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. [Learning transferable visual models from natural language supervision](#). In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.
- Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. 2021. [Contrastive learning with hard negative samples](#). In *Proceedings of the 9th International Conference on Learning Representations (ICLR)*.
- Andrew Rouditchenko, Yung-Sung Chuang, Nina Shvetsova, Samuel Thomas, Rogerio Feris, Brian Kingsbury, Leonid Karlinsky, David Harwath, Hilde Kuehne, and James Glass. 2023. [C2kd: Cross-lingual cross-modal knowledge distillation for multilingual text-video retrieval](#). In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.
- Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. [FaceNet: A unified embedding for face recognition and clustering](#). In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823.
- Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. 2022. [VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training](#). In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35.
- Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. [Representation learning with contrastive predictive coding](#). *arXiv preprint arXiv:1807.03748*.
- Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Guo Chen, Baoqi Pei, Rongkun Zheng, Zun Wang, Yansong Shi, Tianxiang Jiang, Songze Li, Jilan Xu, Hongjie Zhang, Yifei Huang, Yu Qiao, Yali Wang, and Limin Wang. 2024. [Internvideo2: Scaling foundation models for multimodal video understanding](#). In *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part LXXXV*, volume 15143 of *Lecture Notes in Computer Science*, pages 396–416. Springer.
- Wenhao Wu, Haipeng Luo, Bo Fang, Jingdong Wang, and Wanli Ouyang. 2023. [Cap4video: What can auxiliary captions do for text-video retrieval?](#) In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10704–10713.
- Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. [Approximate nearest neighbor negative contrastive learning for dense text retrieval](#). In *International Conference on Learning Representations*.
- Qinghao Ye, Guohai Xu, Ming Yan, Haiyang Xu, Qi Qian, Ji Zhang, and Fei Huang. 2023. [HiTeA: Hierarchical temporal-aware video-language pre-training](#). In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15359–15370.
- Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. 2023. [Sigmoid loss for language image pre-training](#). In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11941–11952.
- Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph Gonzalez, and Ion Stoica. 2023. [Judging llm-as-a-judge with mt-bench and chatbot arena](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 46595–46623. Curran Associates, Inc.

A Training Dataset Samples

The training corpus contains roughly 225K videos: around 213K from a large commercial video library and around 12K community-generated clips from a creative AI platform. Every video is at most one minute long and is sampled at 2 FPS, giving a maximum of 120 frames per clip.

Annotation pipeline. Raw video footage along with paired text queries or descriptions are required for contrastive training. For each video, we run Gemma 3n (Gemma Team, 2025b) in a two-stage prompting pipeline. Stage 1 generates a `complex_summary`: a single dense paragraph that covers action, objects, setting, lighting, composition, camera angles, color palette, depth of field, and overall mood. Stage 2 takes the complex summary as context and extracts the remaining structured fields. Table 8 lists all generated fields and their roles in training.

Field	Role
<code>complex_summary</code>	Dense stylistic+content paragraph (Stage 1)
<code>summary</code>	Concise content description
<code>prompts</code>	Short search queries (4–8 words)
<code>complex_prompts</code>	Multi-clause queries with camera/temporal detail
<code>objects</code>	Primary visual objects and subjects
<code>colors</code>	Prominent colors
<code>structure</code>	Scene organisation and pacing description
<code>style</code>	Lighting, composition, camera movement
<code>intents</code>	Likely search intent keywords
<code>purposes</code>	Creator intent (inform, evoke emotion, etc.)

Table 8: Annotation fields generated per video by the Gemma 3n pipeline. The prompts and `complex_prompts` fields serve as the positive text examples for each video during contrastive training.

The prompts field produces short keyword-style queries, complex_prompts produces longer multi-clause descriptions capturing temporal structure, camera behavior, and compositional detail. Both are used as positive text paired with the corresponding video during contrastive training. After annotation the dataset splits into roughly 211K training and 14K test videos.

Sample annotations. Figures 3–5 show three representative videos with uniformly-sampled frames and generated annotations. The complex_summary is truncated to two sentences, all prompts and three representative complex_prompts are shown in full.

B Qualitative Results Analysis

We present five queries chosen to span the main query types evaluated in Section 5: a cinematographic description requiring low-angle framing, two sequential-action queries, a temporal state-change query, and a simple object-action query. For each query, Figures 6–10 show six uniformly-sampled keyframes from the top-1 video retrieved by each of the four models, ordered left to right in temporal sequence. The row labelled **Attn Pool** is our model, the remaining three rows are the base-lines.

C Multilingual Queries

Table 9 reports dense-retrieval nDCG@1 for the multilingual ProTE text encoder across five languages, evaluated against the temporal attention video embeddings. European languages (French, German) fall within 2.5% of English. Spanish lags by 4.6%. Chinese shows the largest drop at 7.7% below English. This pattern is consistent with findings from cross-lingual video retrieval work (Rouditchenko et al., 2023) and points to data coverage rather than architecture as the bottleneck. The multilingual model scores lower than the English-specialised temporal attention model even on English (0.7900 vs. 0.8463), confirming the expected trade-off between multilingual coverage and single-language quality.

D Production Deployment at Scale

D.1 Pipeline Architecture

The video embedding pipeline is deployed as a chain of microservices coordinated by a central

Table 9: Multilingual retrieval nDCG@1 using the ProTE multilingual text encoder over temporal attention video embeddings.

Language	Dense	RAE	Hybrid
English	0.7900	0.7308	0.7916
French	0.7644	0.6948	0.7632
German	0.7652	0.7016	0.7722
Spanish	0.7438	0.6703	0.7599
Chinese	0.7128	0.6611	0.7178

orchestrator. On video ingestion, the pipeline executes the following stages: **(1) Video download** — the raw video is fetched and staged for processing. **(2) Frame extraction** — frames are sampled at 2 FPS, producing 60 frames for an average 30-second video. This stage runs on CPU. **(3) Frame encoding** — each frame is encoded independently by PROVE, with the GPU-accelerated encoding service processing 1,024 frames per second per pod across a parallel fanout. This is the only GPU stage in the pipeline. **(4) Temporal aggregation** — the aggregator collects all frame embeddings and passes them through the temporal attention pooling head, producing a single dense 768-dim vector. Runs on CPU. **(5) Sparse projection** — the dense embedding is projected through the RAE to produce a 16,384-dim sparse vector for inverted-index ingestion. Runs on CPU. **(6) Index update** — both dense and sparse embeddings are written to the search indexing service, making the video immediately retrievable.

D.2 Per-Video Latency

Table 10 reports the wall-clock latency per stage for an average 30-second video (60 frames at 2 FPS). The end-to-end latency per video is approximately 5.13 seconds, dominated by frame extraction at 5,000 ms. PROVE frame encoding at 1,024 frames/sec processes 60 frames in approximately 59 ms, making it negligible relative to extraction.

Table 10: Per-video latency breakdown for an average 30-second video (60 frames at 2 FPS).

Stage	Hardware	Latency	% Total
Frame Extraction	CPU	5,000 ms	97.4%
Frame Encoding (PROVE)	GPU	59 ms	1.1%
Temporal Aggregation	CPU	50 ms	1.0%
Sparse Projection	CPU	20 ms	0.4%
Total		5,129 ms	100%



Duration: 7s

complex_summary:

“The video shows a series of shots focusing on a blue road sign indicating the direction to Lyon International Airport. The sign is consistently positioned in the lower left quadrant against a clear, bright blue sky with scattered wispy clouds. [...]”

prompts:

“Blue road sign Lyon Intl. Airport right arrow”
 “Airport sign Lyon Intl. Airport airplane”
 “Blue sky airplane airport sign”
 “Directional sign Lyon Airport”
 “Travel sign Lyon Intl. Airport”

complex_prompts:

“Low-angle shot of a blue airport sign indicating Lyon Intl. Airport with a right-pointing arrow, with an airplane flying in the background.”
 “High-angle shot of a blue airport sign with a right-pointing arrow, with an airplane descending towards the airport.”
 “A clear blue sky with scattered wispy clouds, featuring a blue airport sign with the text Lyon Intl. Airport and a right-pointing arrow.”

Figure 3: Dataset Sample 1



Duration: 8s

complex_summary:

“The video features a middle-aged man, likely a doctor, in a medical office setting. He is consistently dressed in a white lab coat over a light-colored shirt and wears glasses and a stethoscope with red tubing around his neck. [...]”

prompts:

“Doctor smiling in medical office”
 “Man forming heart with hands”
 “White lab coat”
 “Heart-shaped object”
 “Medical care compassion”

complex_prompts:

“Middle-aged man in a white lab coat smiling warmly at the camera in a bright medical office, forming a heart shape with his hands.”
 “Doctor with a stethoscope and glasses creating a heart gesture, conveying care and compassion in a yellow-walled clinic.”
 “Optimistic and reassuring scene of a doctor in a medical setting, emphasising heart health and patient well-being.”

Figure 4: Dataset Sample 2

D.3 Capacity Planning for 100M Videos

To index a corpus of 100M videos (average duration 30 seconds) within a 7-day window, the required sustained throughput is:

$$T_{\text{required}} = \frac{100,000,000}{7 \times 24 \times 3600} \approx 165 \text{ videos/sec} \quad (8)$$

Table 11 reports per-service throughput and pod requirements. CPU services are hosted on c5.24xlarge instances (\$4.08/hr, 96 vCPUs), with 26 pods per instance. The PROVE encoding service is hosted on g5.xlarge instances (\$1.006/hr, 1 × NVIDIA A10G), with one GPU pod per instance.

The total 7-day on-demand cost is $22 \times \$56.95 \times$

Table 11: Pod and instance sizing to sustain 165 videos/sec for 7 days over a 100M-video corpus. CPU pods run on c5.24xlarge (26 pods/instance); GPU pods run on g5.xlarge (1 pod/instance).

Service	HW	vid/s/pod	Pods	Instances	Cost/hr
Frame Extraction	CPU	0.80	210	9	\$36.72
Frame Encoding (PROVE)	GPU	17.07	11	11	\$11.07
Temporal Aggregation	CPU	80.00	3	1	\$4.08
Sparse Projection	CPU	200.0	2	1	\$4.08
Total			226	22	\$56.95

$168 \approx \$8,028$, broken down as \$6,169 for CPU instances and \$1,859 for GPU instances.



Duration: 14.5s

complex_summary:

“The video showcases a breathtaking panoramic view of a vast mountain valley with high altitudes and lush vegetation. The scene is dominated by towering, rugged mountains on either side, their slopes covered in a mix of rocky terrain and vibrant green vegetation, including trees and grassy areas. [...]”

prompts:

“Mountain valley with wildflowers”
 “Panoramic mountain landscape”
 “Wildflower meadow in mountains”
 “Mountain valley with clouds”
 “Tranquil mountain scene”

complex_prompts:

“Panoramic view of a vast mountain valley with towering rugged mountains, slow pan-and-tilt camera movement across dense purple, yellow, and white wildflowers.”
 “Wide-angle shot of a valley floor filled with wildflowers and tall green grasses, dynamic sky with blue patches and fluffy white clouds, natural soft diffused lighting.”
 “Eye-level perspective of a mountain landscape emphasising immersion, with composition utilising the Rule of Thirds to frame wildflowers and distant mountain ranges.”

Figure 5: Dataset Sample 3

Query: “A low-angle shot of a person catching a basketball and then shooting a basket”

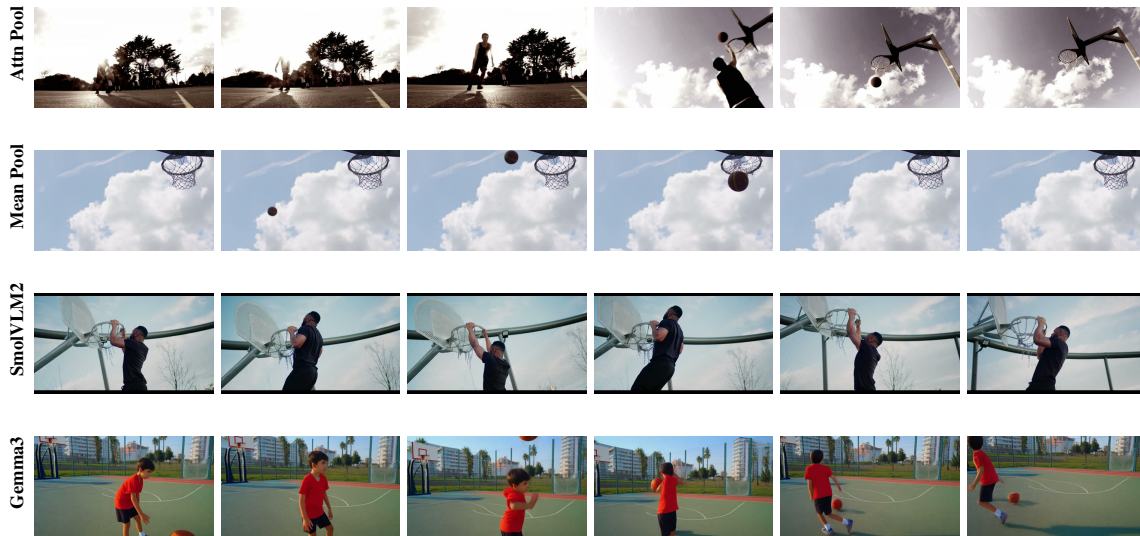


Figure 6: Attn Pool retrieves a genuine low-angle outdoor court clip showing a player catching and shooting across frames, matching both the camera angle and the sequential action. Mean Pool finds a basketball hoop from below against a bright sky but the person is absent, the frame-averaging weighted the static hoop-and-sky appearance over the motion. SmolVLM2 gets the low-angle perspective and a player near a hoop but shows the person hanging on the backboard rather than executing a catch-and-shoot. Gemma3 retrieves a child playing on an outdoor court, missing the adult, the camera angle, and the two-stage action entirely.

D.4 Analysis

Frame extraction is the sole bottleneck. Of the 226 pods required, 210 (92.9%) are CPU-bound frame extractors. All ML inference services combined — PROVE encoding (11 GPU pods), tempo-

ral aggregation (3 CPU pods), and sparse projection (2 CPU pods) — require only 16 pods across 13 instances. This confirms that scaling the corpus does not require scaling ML infrastructure proportionally. GPU-accelerated video decoding, for example via hardware decode units on modern cloud

Query: “A child runs towards an adult and gives them a hug”



Figure 7: Attn Pool shows the full temporal sequence, an adult crouching and waiting, the child closing the distance, and the hug completing across the six frames. Mean Pool retrieves a child running outdoors near a tree with an adult visible, which matches the scene type, but the hug never appears across the frames. SmolVLM2 retrieves two adults hugging indoors, capturing the endpoint gesture but not the child or the running approach. Gemma3 finds a toddler with outstretched arms being lifted, which gets the child and the physical contact but not the directed running motion.

Query: “A time-lapse of a flower wilting”

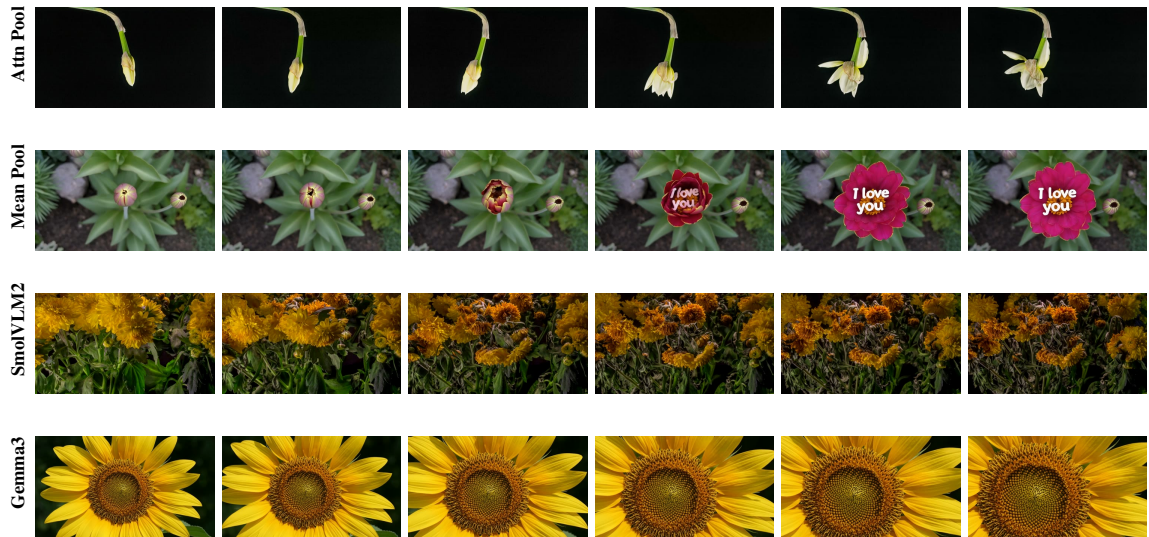


Figure 8: Attn Pool retrieves a macro time-lapse of a white flower on a black background with the drooping arc visible across the frames, directly matching both the temporal modality and the wilting direction. Mean Pool retrieves a flower clip but in the wrong temporal direction: the clip shows a bud opening into bloom, and the later frames transition to a stylised graphic overlay unrelated to the query. SmolVLM2 and Gemma3 both find flower footage but with no discernible change across frames, static shots of marigolds and a sunflower respectively rather than a time-lapse.

Query: “A person steps into an elevator, and the doors close”



Figure 9: Attn Pool retrieves a hotel lobby sequence where a formally dressed person walks toward and enters an elevator, with the approach progression and the doors closing visible across frames. Mean Pool retrieves what appears to be a plain door or wall panel sliding open against a white background, getting a door-related element but without the person, the lobby setting, or the entering action. SmolVLM2 retrieves a dark close-up of a person’s face with no elevator context. Gemma3 retrieves a scene from what appears to be a Chinese television broadcast with an indoor lobby setting but no elevator action and a persistent watermark across every frame.

Query: “A person sitting on a table closing the laptop”

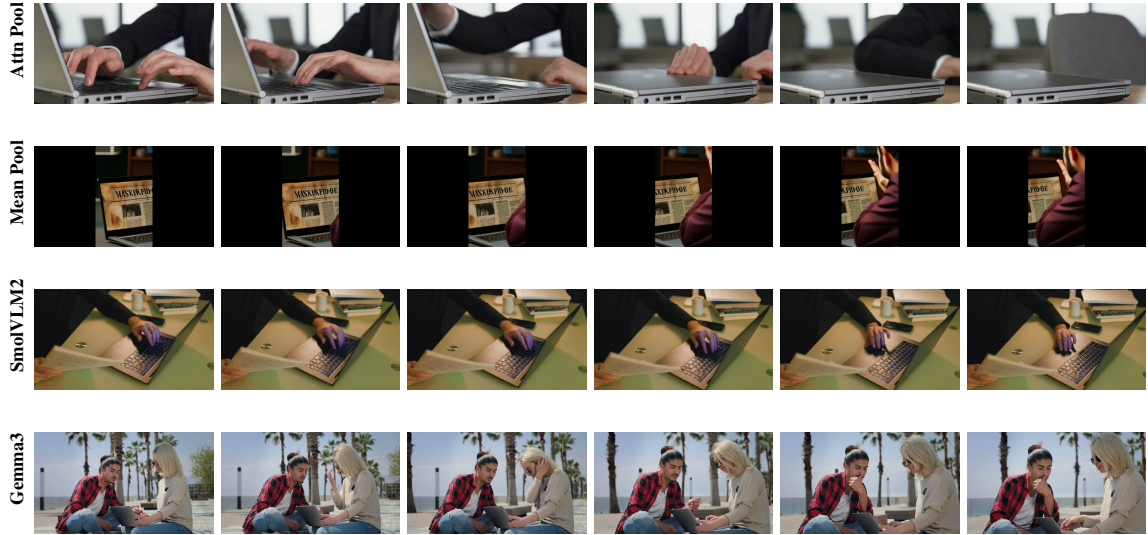


Figure 10: Attn Pool retrieves a close-up sequence of hands lowering the laptop lid, with the closing motion clearly progressing across frames as the query describes. Mean Pool retrieves a darkly-lit desk scene with a laptop and a newspaper visible, matching the object but showing no closing action and little temporal change across frames. SmolVLM2 retrieves a shot of hands on a purple laptop that has the correct object-interaction framing but shows typing rather than closing. Gemma3 retrieves two people sitting and talking outdoors with palm trees in the background, no laptop appears in any frame.

instances, would yield far greater throughput gains than any optimisation of the model inference path.

GPU footprint is minimal. Despite PROVE being a 427M-parameter ViT-L/14 model, its inference is served by just 11 g5.xlarge instances, accounting for only \$1,859 of the \$8,028 total — 23% of infrastructure cost for the only ML-heavy stage. The frozen-encoder design is directly responsible for this efficiency: because PROVE weights are shared with the existing image retrieval pipeline, no additional model development or fine-tuning cost is incurred.

CPU dominates total cost. The 9 c5.24xlarge instances running frame extraction account for \$6,169 (77%) of the 7-day cost despite being the cheapest instance type per hour. This is a direct consequence of the 5,000 ms per-video extraction latency: reducing extraction time by $2\times$ would halve the CPU instance count and reduce total cost by approximately \$3,000.

Extraction optimisation. Frame extraction is the dominant cost driver at 65.6% of total infrastructure spend, despite being the simplest pipeline stage. The primary mitigation is replacing CPU-based software decoding with GPU-accelerated hardware decoding via NVDEC, which is expected to reduce per-video extraction latency substantially and lower the required pod count proportionally. Complementary directions include adaptive frame sampling — extracting fewer frames from static or low-motion segments — and batch-level I/O scheduling to reduce storage contention across concurrent extraction workers.

Failure budget. At a 5% failure rate with one retry, the effective corpus grows to 105M videos, requiring ~ 174 videos/sec. The current frame extraction ceiling is $210 \times 0.8 = 168$ videos/sec, marginally below this. We recommend provisioning **220 frame extraction pods** (9 instances unchanged, with slack capacity) and accepting minor queue buildup during retry bursts, or provisioning a tenth c5.24xlarge instance (\$4.08/hr) as a dedicated retry worker.

E Training and Optimization Details

Trainable parameters (“lightweight” quantified). Only the pooling head is trained. It comprises a learnable query vector (768), a learned positional-encoding table (120×768 , one entry per

Table 12: Four-stage training curriculum for the pooling head. “HN” = hard negatives used; “Positives” indicates the positive-pair source at each stage.

Stage	Ep.	LR	Batch	WD	Warmup	τ	HN
1	30	1e-3	32	0.010	0.05	0.07	No
2	45	5e-4	20	0.010	0.03	0.07	Yes
3	39	3e-4	28	0.015	0.02	0.07	No
4	60	1e-4	16	0.020	0.01	0.05	Yes

frame slot at the maximum 120-frame length), and a single cross-attention block (1 attention head, hidden dimension 768). This totals **2.46M trainable parameters**, or **0.57%** of the 430M-parameter model, on top of the frozen 427.6M-parameter PROVE ViT-L/14 image encoder; the PROTE text encoder is likewise frozen. The deployed configuration uses $L=1$ pooling layer; $L \in \{2, 4\}$ give no improvement (Table 3).

Optimization. We train with AdamW, a cosine learning-rate schedule, and bf16 precision on a single 8-GPU node, with gradient accumulation of 8 and a learnable InfoNCE temperature (initialized $\tau=0.07$). The head is trained with a four-stage curriculum of increasing difficulty; each stage resumes the model weights from the previous stage’s checkpoint with a fresh optimizer/scheduler. Table 12 lists the per-stage settings.

F Statistical Methodology and Full Results

Protocol. nDCG is computed per query, and because all systems are evaluated on the same queries, every system-vs-baseline comparison is paired at the query level. We report 95% confidence intervals from a query-level bootstrap (10,000 resamples of the per-query scores) and test significance with a paired randomization (permutation) test over per-query differences (10,000 permutations, two-sided). Across the family of comparisons in Table 13 we apply a Holm–Bonferroni correction; we report the corrected p_{holm} . Significance is conditional on the graded ground truth: these tests quantify query-sampling variability, not judge error, which is addressed separately in Appendix H.

Note on baseline values. The statistics in this section are computed on a dedicated re-run of the aggregation heads with per-query score logging (required for the bootstrap and randomization tests). Mean pooling scores nDCG@1=0.786 in this run versus 0.7729 in Table 1; the difference is within

Table 13: Full nDCG@ k with 95% bootstrap CIs and Δ vs. mean pooling (paired randomization test, Holm-corrected). *: $p_{\text{holm}} < 0.05$.

Model	k	nDCG@ k [95% CI]	Δ vs. mean [95% CI]	p_{holm}
Mean pooling	1	0.786 [0.752, 0.818]	—	—
Mean pooling	3	0.762 [0.734, 0.789]	—	—
Mean pooling	5	0.748 [0.721, 0.775]	—	—
Mean pooling	10	0.726 [0.699, 0.751]	—	—
Attn +PE (ours)	1	0.851 [0.822, 0.878]	+0.065 [+0.034, +0.097]	0.069
Attn +PE (ours)	3	0.824 [0.799, 0.848]	+0.062 [+0.040, +0.084]	0.040*
Attn +PE (ours)	5	0.812 [0.788, 0.835]	+0.064 [+0.045, +0.082]	0.040*
Attn +PE (ours)	10	0.797 [0.773, 0.819]	+0.071 [+0.056, +0.087]	0.040*
Attn -PE	1	0.814 [0.782, 0.845]	+0.029 [-0.005, +0.061]	1.00
Attn -PE	3	0.792 [0.765, 0.818]	+0.030 [+0.007, +0.053]	1.00
Attn -PE	5	0.783 [0.758, 0.807]	+0.035 [+0.016, +0.053]	0.069
Attn -PE	10	0.762 [0.738, 0.786]	+0.036 [+0.021, +0.052]	0.040*

run-to-run variation and does not affect any comparison, since every Δ , confidence interval, and p -value here is computed against the mean-pooling baseline *from the same run*.

Positional-encoding ablation with significance.

Table 13 gives the full breakdown behind Section 5.4. Removing positional encodings drops nDCG@{1,3,5,10} by 0.037/0.032/0.029/0.035 and removes the statistical significance of the improvement over mean pooling at the top ranks: the -PE head is not significant at @1/@3 ($p_{\text{holm}}=1.0$), whereas the full head is significant at @3/@5/@10 ($p_{\text{holm}}=0.040$). At nDCG@1 the full head’s improvement is +0.065 with a 95% CI of [+0.034, +0.097] (excluding zero) and an uncorrected $p=2 \times 10^{-4}$; the family-wise-corrected value ($p_{\text{holm}}=0.069$) is marginally above threshold, as expected given nDCG@1’s higher per-query variance under a conservative correction over many tests. The effect is consistent in sign and magnitude across all four cutoffs and all pooling depths.

Small categories. Consistent with the caption of Table 2, we treat the *visual attributes* category ($n=5$) as indicative only: no significance test has meaningful power at that sample size, and its bootstrap CI spans most of the metric range.

Per-category confidence intervals and significance. Tables 14 and 16 report the per-category confidence intervals and significance. Our head has the highest nDCG@1 in six of the seven categories—all except Visual Attributes, where $n=4$ makes the estimate unreliable and which we therefore report descriptively and exclude from significance testing. Over the confirmatory family of our head vs. mean pooling across the remaining six categories and four

cutoffs (24 tests), the improvement is significant at the deepest cutoff in three categories—Subject/Object ($\Delta=+0.077$, $p_{\text{holm}}=0.029$), Action/Activity ($\Delta=+0.086$, $p_{\text{holm}}=0.002$), and Emotion/Expression ($\Delta=+0.119$, $p_{\text{holm}}=0.023$), all at nDCG@10, where the gain is largest and most consistent. The remaining categories do not reach family-wise significance individually—per-category samples are small ($n=24-48$), limiting power—but the improvement is positive in sign in every category at every cutoff, and significance strengthens with the aggregate sample (Table 15): the improvement is significant at nDCG@{3,5,10} overall and, most relevantly, on the complex multi-clause query split at nDCG@10 ($\Delta=+0.112$, $p_{\text{holm}}=0.040$), the query type this work primarily targets.

G Aggregation Baselines under the Frozen-Encoder Constraint

To compare aggregation mechanisms fairly, we port only the pooling *architecture* of each baseline and retrain that head on our identical frozen features, data, loss, and hard negatives; we do not use published checkpoints. Published CLIP4Clip (Luo et al., 2022) and X-Pool (Gorti et al., 2022) weights are trained end-to-end on a CLIP backbone over public benchmarks and are unusable here: they expect 512-dimensional CLIP features rather than our 768-dimensional PROVE/PROTE embeddings, were fit to a different query distribution, and using them would require unfreezing or replacing the production encoders, violating the frozen-encoder premise. We therefore resize each head to 768 dimensions and retrain only that head: CLIP4Clip-seqTransf as a sequence Transformer over frames followed by mean pooling, and X-Pool as query-conditioned cross-attention.

Why a query-conditioned method is unsuitable for first-stage retrieval. X-Pool pools each video conditioned on the query text, so a video has no single static embedding and cannot be indexed once for approximate nearest-neighbor search. Production retrieval indexes each of 100K–100M videos offline; a query-conditioned model would instead require re-pooling the entire corpus per query ($O(\text{corpus})$ attention at query time), which is infeasible at scale. Its only deployable form is a two-stage reranker over a shortlist, which is how we evaluate it in Table 3. Even in that favorable setting it does not beat the static attention

Table 14: Per-category nDCG@1 with 95% bootstrap CIs on the seven semantic categories of the Golden Benchmark, for mean pooling, our head, and the two learned baselines (all over identical frozen PROVE features; statistics re-run of Appendix F). Δ is our head vs. mean pooling. Bold = best per category. Point estimates are from the statistics re-run (this appendix) and differ slightly from the primary-pool run in Table 2. † Visual Attributes has $n=4$ and is underpowered; reported as indicative only and excluded from the significance family in Table 16.

Category	n	Mean pooling	ProVE Attn Pooling	CLIP4Clip	X-Pool	Δ (ours–mean)
Subject/Object	38	0.789 [0.668, 0.892]	0.876 [0.784, 0.951]	0.798 [0.677, 0.899]	0.824 [0.716, 0.918]	+0.087
Action/Activity	48	0.818 [0.733, 0.892]	0.846 [0.757, 0.923]	0.823 [0.731, 0.904]	0.734 [0.629, 0.827]	+0.028
Scene/Setting	39	0.892 [0.817, 0.952]	0.959 [0.937, 0.979]	0.856 [0.756, 0.938]	0.892 [0.815, 0.951]	+0.067
Emotion/Expression	25	0.913 [0.824, 0.970]	0.959 [0.935, 0.981]	0.868 [0.753, 0.954]	0.809 [0.678, 0.915]	+0.046
Theme/Topic	24	0.890 [0.800, 0.951]	0.921 [0.829, 0.977]	0.916 [0.829, 0.970]	0.775 [0.626, 0.897]	+0.031
Visual Attributes †	4	0.899 [0.866, 0.932]	0.852 [0.803, 0.951]	0.734 [0.361, 0.973]	0.934 [0.855, 1.000]	−0.047
Cinematography	40	0.855 [0.784, 0.909]	0.870 [0.798, 0.924]	0.694 [0.571, 0.808]	0.793 [0.683, 0.888]	+0.015

Table 15: Our head (+PE) vs. mean pooling on the aggregate query splits, kept as a correction family separate from the per-category tests (Table 16). p_{holm} from the global analysis. The complex-query split—the paper’s primary target—is significant at nDCG@10. *: $p_{\text{holm}} < 0.05$.

Split	k	Δmean	p_{holm}	sig
Simple	10	+0.040	0.202	—
Complex	10	+0.112	0.040	*
Overall	3	+0.062	0.040	*
Overall	5	+0.064	0.040	*
Overall	10	+0.071	0.040	*

head (0.774 vs. 0.851 nDCG@1) and does not beat mean pooling at @1, while adding query-time cost and breaking the drop-in, index-once deployment that is the paper’s premise.

H Judge Validation and Pooling Protocol

Threshold sensitivity. The graded ground truth thresholds VLM relevance scores at 0.85. We recomputed the benchmark at $\tau \in \{0.75, 0.80, 0.85, 0.90\}$ (Table 17); the system ranking is invariant and 0.80–0.85 is a flat band, so 0.85 is not a sensitive choice. At 0.90 some queries retain no video above threshold and drop out, so the lower absolute nDCG there is a coverage effect on a smaller query set; even so, our model’s lead widens.

Pool composition and pooling bias. Each query’s candidate pool is the union of the top-100 retrievals from every system in the comparison—the production system (metadata), mean pooling,

Table 16: ProVE Attn Pooling vs. mean pooling per category: ΔnDCG and Holm-corrected p at the two deepest cutoffs. Correction is over the confirmatory family of this single comparison across 6 categories $\times$ 4 cutoffs (24 tests); raw p -values are identical to the global analysis, only the family differs. Visual Attributes ($n=4$) is excluded from the family as underpowered and reported descriptively in Table 14. *: $p_{\text{holm}} < 0.05$.

Category	n	$\Delta@5$ (p_{holm})	$\Delta@10$ (p_{holm})	sig
Subject/Object	38	+0.084 (0.204)	+0.077 (0.029)	@10*
Action/Activity	48	+0.067 (0.204)	+0.086 (0.002)	@10*
Scene/Setting	39	+0.047 (0.364)	+0.041 (0.234)	—
Emotion/Expression	25	+0.109 (0.143)	+0.119 (0.023)	@10*
Theme/Topic	24	+0.029 (1.000)	+0.039 (1.000)	—
Cinematography	40	+0.046 (0.959)	+0.051 (0.805)	—

Table 17: Judge-threshold sensitivity on the Golden Benchmark (nDCG@1). The system ranking is invariant across thresholds.

τ	Mean pool @1	Ours @1	Δ
0.75	0.807	0.863	+0.055
0.80	0.786	0.851	+0.065
0.85	0.786	0.851	+0.065
0.90	0.572	0.669	+0.097

thumbnail, our attention head, and all ablation baselines (X-Pool, CLIP4Clip, attention with $L=2$ and $L=4$)—with all resulting (query, video) pairs judged by the VLM; scores ≥ 0.85 form the graded ground truth. This follows the standard TREC pooling protocol (Craswell et al., 2021) and is unbiased for or against any single system: every compared retriever contributes its top-100 symmetrically, so no system’s uniquely surfaced relevant videos go unjudged and no method is over-sampled (pooling only decides which pairs are judged; the judge then

Table 18: Zero-shot MSR-VTT test-1k (text→video). Recall@K.

MSR-VTT test-1k	R@1	R@5	R@10
Trained attention head	0.402	0.634	0.728
Mean pooling (frozen frames)	0.411	0.645	0.757

scores relevance per pair). Pool incompleteness—relevant videos that no system retrieved in its top-100—affects all systems equally and is the accepted graded-pooling standard.

Judge reliability on temporal queries. The temporal-reasoning limitation we note in the Limitations section is a general property of current VLM/LLM judges rather than specific to Qwen2.5-VL-72B (Bai et al., 2025); in a manual spot-check the judge showed good accuracy on temporally-oriented (query, video) pairs. Crucially, our headline temporal evidence—the frame-shuffle/reversal probe (Section 5.4)—is judge-free, measured directly in embedding space with no judge in the loop, and is therefore unaffected by any residual judge limitation. A full human–judge agreement study is left to future work (Appendix J).

I Public Benchmark: Zero-Shot MSR-VTT

Although this is an applications paper centered on a production setting, we additionally report zero-shot results on MSR-VTT test-1k (text→ video, 1000 queries); neither the frozen PROVE/PROTE encoders nor the pooling head were trained on MSR-VTT (Table 18).

The head shows a marginal drop relative to mean pooling. A per-query look at the R@1 failures unique to our method shows they are near-misses rather than errors: the head retrieves other videos that genuinely match the query text but are not the single clip marked as ground truth. MSR-VTT test-1k contains ~50 videos per category, so many queries have several equally valid near-duplicate answers and R@1 penalizes retrieving the “wrong” one. A secondary factor is domain shift: the head is a lightweight adapter trained on professional stock footage with attribute-dense, LLM-authored captions, whereas MSR-VTT comprises short web clips with terse crowd-sourced captions. This is consistent with the paper’s central finding that one-to-one protocols underestimate quality on corpora with many valid answers, and we report it in the

interest of completeness.

J Future Work

Human–judge agreement on temporal queries.

Our benchmark’s ground truth is produced by a single VLM judge, and while we also present a judge-free temporal evidence (Section 5.4), quantifying judge reliability remains valuable. As future work we plan a human–judge agreement study on a stratified sample of (query, video) pairs spanning relevance-score bands, semantic categories, and a dedicated temporally-oriented subset, reporting agreement via Cohen’s κ and Spearman correlation, with particular attention to the temporal subset where LLM-based judges are known to be weakest. We retain single-judge annotation as an acknowledged limitation in the meantime.

Higher-frame-rate and motion-aware sensing.

The fixed 2 FPS sampling limits camera-motion and fast-action cues (the main reason cinematography queries gain least); adaptive or higher-rate sampling and lightweight motion features are a natural extension.

Event-ordering-aware hard negatives. Queries describing causally ordered events require sensitivity to action sequence, not just co-occurrence. Constructing hard negatives by shuffling or reversing frame sequences would explicitly train the head to distinguish correct from incorrect orderings, complementing the order-sensitivity we measure in Section 5.4.

Multilingual fine-tuning. The 7.7% nDCG@1 gap for Chinese queries (Appendix C) reflects a data-coverage bottleneck; fine-tuning the pooling head on multilingual query–video pairs would directly address it.