CoVisco: Codec-Native Vision Encoder with Native Token Compression for Efficient Unified Visual Understanding

Anonymous authors
Paper under double-blind review

Abstract

Vision-language models face a fundamental scaling bottleneck: the number of visual tokens grows with frames times patches per frame, and long-video understanding quickly becomes computationally prohibitive. Existing remedies compress visual tokens *after* the vision encoder and *before* the language model—an inherently lossy post-hoc pruning of a representation that was never trained to be compressed, with particularly severe information loss on video. Recent work has moved compression into encoder pretraining, but does so by discarding most fine-grained tokens outright, conflating *compression* with *deletion*. We present CoVisco, a unified image-video codec-native vision encoder that performs **native token compression**: during pretraining, each video segment is equipped with a small set of learnable *abstract tokens* that learn to absorb the segment's semantics, while all fine-grained patch tokens retain their full information flow—compression and fine-grained preservation coexist rather than compete. Cross-segment interaction flows exclusively through abstract tokens, providing a long-video modeling scheme that avoids dense cross-frame full attention. For MLLM deployment, a lightweight token selector further enables a family of token strategies that combine abstract tokens with top-K fine-grained tokens selected on demand, dynamically trading accuracy for the visual-token budget. In codec mode, the selector operates on the tokens already retained by codec-based input selection, yielding a two-stage compression pipeline before the LLM: codec-based selection first reduces the visual input, while the learned abstract-token interface and selector further reduce the visual sequence exposed to the LLM. For uniformly sampled and frame-collage inputs, the same learned second stage operates directly on the encoder's fine-grained token stream. Together, these components provide multi-level visual modeling: patch tokens preserve fine-grained evidence, abstract tokens encode segment-level semantics, and abstract-mediated interaction builds video-level context. Pretrained with three contrastive objectives on 565M image–text pairs, and 6.4M videos, CoVisco is competitive with selected larger embedding models on video-oriented evaluation while using a compact vision tower. It retains useful image-side retrieval and classification performance, and compresses 64-frame videos into as few as 400 visual tokens (2.4% retention) while achieving video understanding performance comparable to, and in some cases better than, OneVision-Encoder under the evaluated settings. These results highlight its ability to support efficient unified visual understanding across both images and videos.

1 Introduction

Unified vision-language models must process images and videos through a common visual interface, but video token counts grow with both duration and spatial resolution. A 256-frame video at 448×448 already corresponds to roughly 262K patch tokens, creating a bottleneck for both visual encoding and LLM prefill. The central challenge is therefore not only how to sample frames, but how to learn a visual representation whose bandwidth can adapt to the downstream task (Li et al., 2023a; Bai et al., 2025).

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

Most existing methods address this bottleneck after dense visual encoding, by pruning, merging, or summarizing patch tokens before they reach the LLM (Chen et al., 2024b; Bolya et al., 2023; Li et al., 2023a; 2024). Such post-hoc compression can create a train–deploy mismatch: the encoder is trained with dense patch flow, but the deployed interface must preserve information after many tokens have been removed. This is especially problematic for video, where brief actions, small objects, or text may occupy only a few patches in a few frames. Encoder-internal compression improves this alignment, but methods that retain only a compressed trace can permanently remove evidence needed for OCR, spatial reasoning, or fine-grained temporal understanding.

Recent codec-native encoders use motion and residual signals to reduce the visual input before or during encoding, including OneVision-Encoder (Tang et al., 2026) and Mage-VL (Yang et al., 2026). CoVisco is complementary: in codec mode, codec selection first reduces the input and a learned abstract-token interface and selector compress the encoded stream again before the LLM. This "compression after compression" perspective motivates the name CoVisco. The same learned interface also supports uniformly sampled and frame-collage inputs, where it operates directly on the fine-grained token stream. Unlike methods that equate compression with deletion, CoVisco preserves fine-grained patch tokens inside the encoder and exposes them on demand.

We present CoVisco (**CO**dec-native **VIS**ion encoder with native token **CO**mpression), a unified image-video encoder built around three ideas. First, each segment contains learnable abstract tokens supervised through complementary image–text, image–image, and video–text objectives; these tokens provide a compact segment representation. Second, alternating intra-segment and abstract-communication layers avoid dense cross-segment patch attention while preserving the fine-grained patch stream. Third, a lightweight selector exposes abstract-only or abstract+top-K outputs to the LLM, allowing the visual budget to be adjusted at deployment. In codec mode this is "compression after compression": codec selection reduces the input first, followed by learned LLM-side compression. The same learned interface also supports uniform and frame-collage inputs.

Across the reported evaluations, CoVisco provides a compact unified interface: it is competitive with selected larger models on video-oriented representation tasks, retains useful image-side retrieval and classification performance, and supports video understanding with as few as 400 visual tokens in the evaluated 64-frame setting. Our contributions are threefold:

1. **Native compact representation with preserved fine-grained flow.** Abstract tokens learn a compact representation inside the ViT, while patch tokens remain available for full-token or top-K downstream use.
2. **Abstract-mediated segmented video encoding.** Alternating attention layers restrict cross-segment interaction to abstract tokens, avoiding dense patch-to-patch attention while retaining a global communication path.
3. **Dynamic unified image-video interface.** A selector supports abstract-only and abstract+top-K deployment across codec, uniform, and collage inputs, and a unified multi-resolution pretraining recipe transfers across retrieval and language-conditioned understanding tasks.

2 Related Work

2.1 Vision encoders and codec-native long-video processing

Contrastive vision encoders such as CLIP, SigLIP, SigLIP2, and MetaCLIP (Radford et al., 2021; Zhai et al., 2023; Tschannen et al., 2025; Xu et al., 2024) form the standard visual front-end for VLMs. Recent systems such as MoonViT (Kimi Team et al., 2025) support native-resolution inputs, while OneVision-Encoder (Tang et al., 2026) shows that codec motion and residual signals can provide useful input sparsity. Long-video systems further combine frame sampling, temporal aggregation, positional encoding, codec-aware selection, or streaming memory. Sampling-based methods reduce frames or merge patches before encoding; codec-structured methods such as CoViAR, EMA, OneVision-Encoder, and Mage-VL exploit compressed-video structure; and streaming methods maintain a compact temporal memory (Wu et al., 2018; Zhao et al., 2025; Tang et al., 2026; Yang et al., 2026; Chen et al., 2024a; Zhang et al., 2025a). These approaches expose a trade-off between temporal coverage, computational cost, and fine-grained evidence. CoVisco is complementary: codec

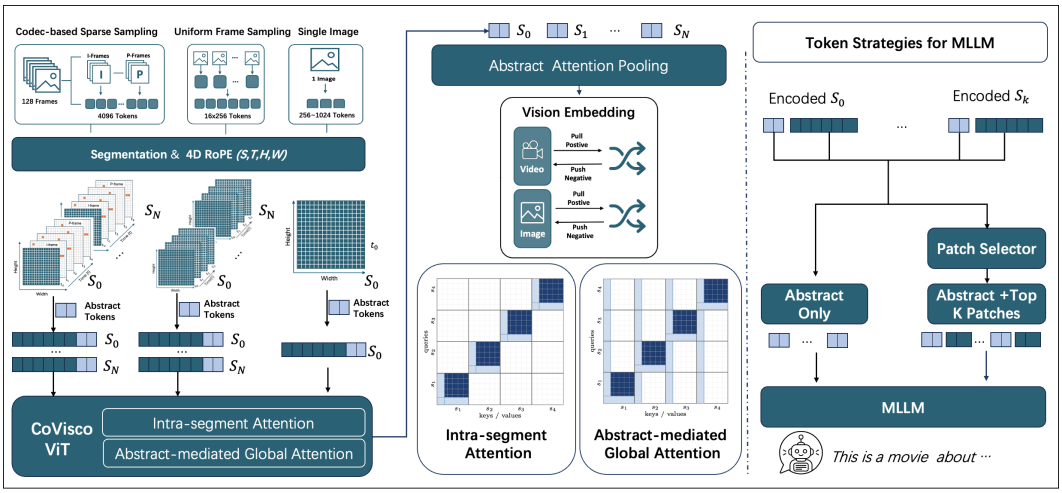

Figure 1: Framework of CoVisco. (1) The input video is divided into temporal segments and converted into codec-native visual tokens using uniform, codec-based, or collage sampling. (2) Inside the ViT, alternating attention blocks perform full attention within each segment and abstract-mediated communication across segments. The central attention maps make this distinction explicit. Each segment contains learnable abstract tokens (orange) and fine-grained patch tokens (blue). (3) The ViT outputs both token types: abstract tokens are pooled and supervised by image–text, image–image, and video–text contrastive objectives, whereas fine-grained tokens are preserved throughout the encoder and remain available to a lightweight selector. At inference, the two outputs support abstract-only, abstract-plus-top-K, and full-token modes.

sampling is an optional input policy, while segmented attention routes cross-segment communication through abstract tokens for both codec and uniform inputs.

2.2 Visual token compression and query-based interfaces

Post-hoc methods prune, merge, or summarize tokens after visual encoding, including importance-based dropping, token merging, and query-based summarization (Rao et al., 2021; Liu et al., 2023; Chen et al., 2024b; Bolya et al., 2023; Marin et al., 2023; Li et al., 2023a; 2024). They are easy to attach to existing VLMs, but the dense encoder is not necessarily trained for the deployed compressed interface. Pretraining-time methods such as Video-LaVIT and OneVision-Encoder use codec-derived signals during visual encoding, while LLaVA-UHD v4 places early compression inside shallow ViT layers (Jin et al., 2024b; Tang et al., 2026; Fang et al., 2026). Learned query interfaces such as Perceiver, BLIP-2, and Chat-UniVi provide compact summaries, but typically replace or aggregate away fine-grained patch tokens (Jaegle et al., 2021; Li et al., 2023a; Jin et al., 2024a). Recent studies also question whether attention importance reliably identifies redundancy and whether common benchmarks isolate compression sensitivity (Wen et al., 2025b;a; Liao et al., 2026). CoVisco instead supervises abstract tokens as a compact segment representation and the exclusive cross-segment communication channel, while preserving the fine-grained patch stream for full-token or top-K use. This yields an adjustable interface that separates compression from permanent deletion.

3 Method

We present CoVisco, a codec-native vision encoder with a compact token interface. Abstract tokens learn segment-level representations, patch tokens remain available for fine-grained evidence, and a selector controls the visual budget exposed to the language model. We describe the interface, segmented attention, pretraining, selector, and training curriculum below; Figure 1 summarizes the design.

3.1 PRELIMINARIES AND PROBLEM SETTING

Given a video of T frames at resolution $H \times W$, patchify with patch size $p = 14$ into $n_{\mathrm{f}} = \lceil H/p \rceil \lceil W/p \rceil$ tokens per frame. The video is partitioned into S segments of T_{seg} frames each ($T_{\mathrm{seg}} = 32$ by default; $S = 1$ for images). Each segment carries Q learnable **abstract tokens** ($Q = 100$), so a segment holds $n = n_{\mathrm{f}} T_{\mathrm{seg}} + Q$ tokens in total. The encoder is a ViT-L backbone (24 layers, hidden size $d = 1024$, 16 heads, FlashAttention-2, $\sim$300M parameters) shared by images and videos; an image is simply the degenerate case $S = 1$.

The goal is twofold: (i) learn a compact representation—of size $O(S \cdot Q)$—that is useful for retrieval and coarse understanding, while preserving fine-grained tokens for optional downstream access; and (ii) keep long-video computation tractable while avoiding dense cross-segment patch-to-patch attention.

The encoder has three functional levels: patch tokens preserve local evidence, abstract tokens summarize each segment, and abstract-mediated attention provides video-level context. The selector exposes these levels through abstract-only, abstract-plus-top-K, or full-token strategies.

The visual pipeline is *input → patch embedding and segmentation → abstract-token augmented ViT* → two output paths. The abstract path pools $\mathbf{A}$ for contrastive pretraining and compact language-model input, while the patch path preserves $\mathbf{P}$ for optional selector-based access. Given S segments with m visible patch tokens per segment, patchification produces $\mathbf{X} \in \mathbb{R}^{S \times m \times d}$. A shared learnable abstract-token matrix $\mathbf{Q} \in \mathbb{R}^{Q \times d}$ is expanded over the segments and prepended to form

$$\mathbf{H}_s^{(0)} = [\, \mathbf{Q} \,\|\, \mathbf{X}_s \,] \in \mathbb{R}^{(Q+m) \times d}, \qquad s = 1, \ldots, S. \tag{1}$$

The resulting segment sequences pass through the alternating segmented ViT layers described in Section 3.3; 4D RoPE is applied to the visible tokens using their segment, temporal, and spatial coordinates. At the output, the sequence is split into abstract tokens $\mathbf{A} \in \mathbb{R}^{S \times Q \times d}$ and fine-grained patch tokens $\mathbf{P} \in \mathbb{R}^{S \times m \times d}$. The abstract path pools $\mathbf{A}$ and applies the three contrastive projection heads described in Section 3.5, while the patch path preserves $\mathbf{P}$ for the selector. For multimodal generation, the selected visual sequence is mapped by a trainable MLP projector into the input space of the downstream language model. Therefore, the language model can receive $\mathbf{A}$ alone, $\mathbf{A}$ together with the selector's top-K tokens from $\mathbf{P}$, or all visual tokens, without changing the pretrained ViT.

3.2 ABSTRACT TOKENS: A NATIVE COMPRESSION CHANNEL

Each segment is prefixed with a shared set of Q learnable parameters $\mathbf{Q} \in \mathbb{R}^{Q \times d}$ (initialized $\mathcal{N}(0, 0.05^2)$), so the input to the encoder is $[\, \mathbf{Q} \,\|\, \mathbf{x}_1 \ldots \mathbf{x}_{n_{\mathrm{f}} T_{\mathrm{seg}}} \,]$ per segment. Crucially, **only abstract tokens are pooled and supervised** (Section 3.5): the contrastive signal must therefore flow through $S \cdot Q$ tokens, which encourages them—as a consequence of the training objective—to become sufficiently informative for retrieval and coarse understanding. Compression is thus *native*: it is learned by the encoder during pretraining as a by-product of the objective, not imposed afterwards by cropping the input or the representation. Patch tokens, meanwhile, receive no direct supervision yet participate fully in every layer; they retain a fine-grained information path and remain available for downstream use (Section 3.6).

The design separates two roles: abstract tokens provide the compact supervised representation, while patch tokens preserve fine-grained information for optional downstream use.

3.3 SEGMENTED ATTENTION WITH ABSTRACT-MEDIATED CROSS-SEGMENT COMMUNICATION

The 24 encoder layers alternate between two attention patterns.

Let N denote the original number of patch tokens before adding abstract tokens, let the N patch tokens be divided evenly into S segments, and let $m = N/S$ be the number of patch tokens per segment. Each segment therefore contains $n = m + Q$ tokens after adding its Q abstract tokens.

Intra-segment layers. Tokens attend fully within their own segment; the per-layer cost is $O(S(m + Q)^2)$. **Abstract-communication layers.** Each token attends to local patch tokens and the SQ abstract tokens from all segments, with cost $O(S(m+Q)(m+SQ))$. Thus, cross-segment information flows exclusively through abstract tokens. Compared with dense attention over N patch tokens, the leading

patch-to-patch term changes from $O(N^2)$ to $O(N^2/S)$, with additional abstract communication terms. This alternating structure also defines the model's inductive bias: segment summaries retain persistent semantics while fine-grained tokens preserve local evidence. Full derivations are given in Appendix B.

3.4 Positional encoding: 4D RoPE over (segment, time, height, width)

We extend the 3D RoPE of prior codec-native encoders with a segment axis. The rotary embedding factorizes head-dim frequencies into four groups with a **2:4:5:5** split over (S, T, H, W): segments, being coarse, need the least positional bandwidth, while space receives the most. Patch positions are computed on the *full unpruned* spatio-temporal grid (a virtual grid of $S \times (T_{\text{seg}}+1) \times h \times w$) and then gathered for whichever subset of patches is actually visible—so sparsity (from codec sampling or token selection) never perturbs the geometry, mirroring the shared positional encoding principle of OneVision-Encoder (Tang et al., 2026). Abstract tokens are placed at temporal index $t = 0$ of their segment on a 10×10 spatial grid ($Q = 100$), anchoring each abstract token to a spatial region of the segment while leaving it "before" all real frames—encoding that it *summarizes*, rather than *observes*.

3.5 Unified multi-resolution contrastive pretraining

Objectives. Pooled abstract tokens are passed to an attention-pooling head and three projection heads trained with image–text, image–image, and video–text SigLIP losses. Because supervision reaches the encoder only through abstract tokens, all three objectives shape the compact representation.

Data and targets. We use 480M LAION-2B image–caption pairs, 85M pairs from `mvp-lab/LLaVA-OneVision-1.5-Mid-Training-85M`, and 6.4M 30/60-second videos from `mvp-lab/LLaVA-OneVision-2-Data`. Target embeddings are pre-extracted: SigLIP2 image/text targets are used for the LAION stream, and Qwen3-VL-Embedding-8B text targets for the 85M image and video-caption streams. The three heads map to the corresponding target spaces. The image and video streams are mixed during training with per-stream batch sizes and gradient accumulation.

Training recipe. Images use dynamic resolutions $\{224, 336, 448\}$; videos use 224×224 inputs and four segments. Global batch sizes are 32K for images and 3,200 for videos. Codec and uniform sampling are each used with probability $1/2$; the uniform branch samples 16 frames and uses frame collage with probability $1/2$.

3.6 Token selector and deployment modes

While abstract tokens suffice for retrieval and coarse understanding, tasks such as OCR, spatial grounding, and fine-grained temporal reasoning need fine-grained evidence. CoVisco therefore trains a lightweight **token selector** that ranks the encoder's patch tokens for a given input, and supports a continuum of deployment modes. In codec mode, these patch tokens are already the subset retained by codec-based input selection; the selector therefore performs a second, LLM-oriented compression over the codec-selected stream. The selector follows the DynamicViT-style differentiable score-gating idea for hard token selection (Rao et al., 2021), while using a segment-aware scoring transformer that is tailored to our abstract-token interface. In our default configuration, the scoring transformer contains two layers. Each layer first contextualizes patch tokens within their own segment, then lets each patch token attend to the abstract tokens of that segment, and finally applies a feed-forward network. The resulting patch representations are mapped to scalar keep scores, and the selector chooses top-K patches independently for each segment. Appendix A gives the selector's tensor shapes and layer-level implementation details.

- **Abstract-only**: feed the LLM the $S \cdot Q$ abstract tokens (highest compression; approximately 400 tokens for the evaluated 64-frame, four-segment setting);
- **Abstract + top-K**: augment abstract tokens with the K highest-ranked patch tokens selected on demand—K is a runtime budget parameter that exposes a controllable accuracy–token trade-off;
- **Full**: all patch tokens, when compute is unconstrained.

The selector is trained with Qwen3-1.7B on 1M images and 1M videos at native resolution while the ViT remains frozen; the selector, projector, and LLM are trainable. This preserves the pretrained visual backbone and lets the selector retain fine-grained evidence when abstract tokens alone are insufficient.

3.7 Training data and four-stage curriculum

The four stages are: (1) ViT contrastive pretraining on 480M LAION-2B image–caption pairs, 85M pairs from `mvp-lab/LLaVA-OneVision-1.5-Mid-Training-85M`, and 6.4M 30/60-second videos from `mvp-lab/LLaVA-OneVision-2-Data`, followed by freezing the ViT; (2) selector, projector, and Qwen3-1.7B training on 1M images and 1M videos with the ViT frozen; (3) captioning-based alignment with Qwen3-4B on 4M images and 2M videos; and (4) instruction tuning of Qwen3-4B-Instruct-2507 and the projector on 740K LLaVA-Next image instructions and 800K LLaVA-Next-Videos samples, with the ViT and selector frozen. The first stage uses image resolutions $\{224, 336, 448\}$, 224×224 video inputs, four segments, and global batch sizes of 32K and 3,200 for image and video inputs; the final video stage uses equal-probability codec or uniform sampling with 16 frames in the uniform branch.

Compute. ViT contrastive pretraining was conducted on 4 nodes equipped with H800 GPUs. MLLM supervised fine-tuning (SFT) was conducted on 2 nodes equipped with H800 GPUs.

4 Experiments

4.1 Evaluation setup

All multimodal understanding experiments use Qwen3-4B-Instruct-2507 as the fixed language backbone. We report codec-guided and frame-based inputs. For CoVisco, the reported visual-token budget counts tokens exposed to the LLM: an abstract-only setting uses $S \cdot Q$, while abstract+top-K additionally uses K selected patch tokens. This LLM-side budget is distinct from the encoder's internal token count and attention cost.

Reference models use their reported protocols: OneVision-Encoder passes 10,368 visual tokens to the LLM, SigLIP2 uses 512-resolution frames, and OneVision- Encoder uses 504-resolution codec or frame inputs. CoVisco uses 224-resolution frame inputs or 224/504-resolution codec inputs, then retains only the abstract tokens and selected fine-grained tokens for the LLM. Detailed comparison paths are given in Appendix D.

4.2 Image-side transfer reference

Table 1 provides a compact reference for CLIP-L-scale zero-shot image classification and image-text retrieval results, retaining only the L/14 and L/16 blocks from the SigLIP 2 comparison table.

Table 1 shows that CoVisco retains useful image-side performance while using a unified encoder for both images and videos. Its zero-shot classification scores remain below the strongest recent L/14 and L/16 references, whereas several image–text retrieval metrics are competitive. The 448-resolution setting further improves several image and retrieval metrics, indicating that the multi-resolution training recipe learns a useful visual representation rather than a representation tied to a single fixed grid. This image-side behavior is important for the paper's central claim: CoVisco is a unified image-video encoder whose main advantage is the combination of compact video interfaces, flexible resolution support, and shared image/video processing.

4.3 Unified image-video-document representation

Table 2 reports the MMEB-V2 benchmark used to evaluate unified image, video, and visual-document embedding. This benchmark is useful for separating two properties that are otherwise entangled in downstream VLM accuracy: whether the visual representation is broadly retrievable, and whether it transfers beyond caption-style image–text matching to video and visual-document tasks.

Despite using only a 0.3B vision tower, CoVisco is competitive with selected larger open-source multimodal embedding models on video-oriented portions of MMEB-V2. Its strongest relative

Table 1: **Zero-shot image classification and image-text retrieval reference.** Results are adapted from the SigLIP 2 zero-shot comparison (Tschannen et al., 2025); only the ViT-L/14 and ViT-L/16 blocks are retained. Best values within each block are in bold formatting.

				ImageNet-1k				COCO		Flickr		XM3600	
ViT	Res.	Seq.	Model	val	v2	ReaL	ObjNet	T→I	I→T	T→I	I→T	T→I	I→T
L/14	224	256	OpenCLIP (Ilharco et al., 2021)	74.0	61.1	–	66.4	46.1	62.1	75.0	88.7	–	–
			CLIP (Radford et al., 2021)	75.5	69.0	–	69.9	36.5	56.3	65.2	85.2	–	–
			MetaCLIP (Xu et al., 2024)	79.2	72.6	–	74.6	55.7	–	83.3	–	–	–
			CLIPA-v2 (Li et al., 2023b)	79.7	72.8	–	71.1	46.3	64.1	73.0	89.1	–	–
			EVA-CLIP (Sun et al., 2023)	79.8	72.9	–	**75.3**	47.5	63.7	77.3	89.7	–	–
			DFN (Fang et al., 2024)	**82.2**	**75.7**	–	74.8	**59.6**	–	**84.7**	–	–	–
	224	256	CoVisco(Ours)	71.5	64.0	78.7	61.1	47.5	66.4	75.3	91.1	**47.7**	**60.5**
	448	1024	CoVisco(Ours)	73.3	66.1	80.3	65.1	48.5	**67.0**	77.3	**92.1**	47.4	60.2
L/16	256	256	SigLIP (Zhai et al., 2023)	80.5	74.2	85.9	77.9	51.2	69.6	81.3	92.0	30.9	40.1
			SigLIP2Tschannen et al. (2025)	82.5	76.8	87.3	83.0	54.7	71.5	84.1	94.5	46.5	56.5
	384	576	SigLIP (Zhai et al., 2023)	82.1	75.9	87.1	80.9	52.8	70.5	82.6	92.9	31.4	39.7
			SigLIP2Tschannen et al. (2025)	83.1	77.4	87.6	84.4	**55.3**	71.4	85.0	95.2	47.1	56.3
	512	1024	SigLIP2Tschannen et al. (2025)	**83.5**	**77.8**	**87.7**	**84.6**	55.2	**72.1**	**85.3**	**95.8**	**47.4**	**56.7**

Table 2: Results on the MMEB-V2 benchmark (Meng et al., 2025). CLS: classification, RET: retrieval, VDR: ViDoRe, VR: VisRAG. †: link to the model's homepage.

Model	**Vision Tower**	**Image**	**Video**		**VisDoc**		
		CLS	**CLS**	**RET**	**VDRv1**	**VDRv2**	**VR**
# of Datasets →		10	5	5	10	4	6
VLM2Vec (Jiang et al., 2025)	2B	58.7	33.4	20.6	49.8	13.5	51.8
VLM2Vec-V2 (Meng et al., 2025)	2B	62.9	39.3	28.8	75.5	44.9	79.4
GME (Zhang et al., 2025b)	2B	54.4	34.9	25.6	86.1	54.0	82.5
Ops-MM-embedding-v1†	2B	68.1	53.6	41.8	76.4	53.2	77.6
RzenEmbed (Jian et al., 2025)	2B	68.5	50.4	46.6	87.1	55.1	87.2
VLM2Vec (Jiang et al., 2025)	8B	62.7	39.1	29.0	56.9	9.4	59.1
GME (Zhang et al., 2025b)	8B	57.7	37.4	28.4	89.4	55.6	85.0
Ops-MM-embedding-v1†	8B	69.7	59.7	45.7	80.1	59.6	79.3
RzenEmbed (Jian et al., 2025)	8B	70.6	58.8	51.0	89.7	60.7	88.7
IFM-TTE†	8B	76.7	60.5	51.7	85.2	71.5	92.7
Seed-1.6-embedding-0615†	-	76.1	55.0	51.3	85.3	56.6	84.7
Seed-1.6-embedding-1215†	-	75.0	85.2	59.1	90.0	60.3	90.0
Qwen3-VL-Embedding-2B	2B	70.3	71.9	53.9	84.4	65.3	86.4
Qwen3-VL-Embedding-8B	8B	74.2	78.4	58.7	87.2	69.9	88.7
CoVisco (ours)	0.3B	55.1	65.1	46.2	61.8	42.6	63.8

behavior is on video classification, where it exceeds several open-source 2B/8B embedding models and is behind the Qwen3-VL embedding family among the open-source models listed in Table 2; its video retrieval score is also within the range of several substantially larger open-source models. Performance on image and visual-document metrics remains lower than that of the strongest listed baselines. These results support the intended role of the abstract-token channel: it provides a compact representation that remains useful for video retrieval and discrimination, rather than serving only as an intermediate compression device for downstream VLMs.

4.4 Core video understanding under a fixed language backbone

Table 3 compares vision encoders under a unified multimodal setting with Qwen3-4B-Instruct-2507 as the language backbone. This controlled setting keeps the decoder fixed and varies the visual

Table 3: **Comparison of different vision encoders on multimodal benchmarks.** All models are evaluated on a unified multimodal setting using Qwen3-4B-Instruct-2507 as the language backbone. SigLIP2 uses `siglip2-so400m-patch16-naflex`. **(Codec)** indicates codec-based token sampling, while **(Frame)** indicates uniform frame sampling. Each tuple in the column headers has the form (sampling mode, input resolution, fine-grained token retention ratio): the integer denotes the input resolution, and the fraction denotes the proportion of encoded fine-grained tokens retained when feeding the LLM (with 0.0 indicating that no fine-grained tokens are retained). Bold values indicate the best performance under the same evaluation setting.

Task	Benchmark	Qwen3-4B-Instruct-2507							
		OV-Encoder (Codec,504,1.0)	OV-Encoder-Frame (Frame,504,1.0)	SigLIP2 (Frame,512,1.0)	CoVisco (Codec,224,0.4)	CoVisco (Codec,504,0.4)	CoVisco (Frame,224,0.0)	CoVisco (Frame,224,0.2)	CoVisco (Frame,224,1.0)
Video	MVBench	52.4	49.8	47.2	56.7	53.2	56.0	**56.9**	–
	MLVU-dev	46.3	49.4	48.4	58.5	56.7	**61.2**	59.8	–
	NExT-QA (MC)	**75.6**	71.9	70.6	74.0	71.3	73.5	73.9	–
	VideoMME	53.4	49.3	46.8	55.7	53.0	53.5	**57.0**	–
	Perception Test	**60.3**	56.7	56.0	–	–	59.0	59.5	–
	TOMATO	22.2	21.8	22.3	26.0	**26.4**	25.5	25.7	–
	LongVideoBench-Val-Video	50.4	45.5	45.2	50.5	48.2	47.3	**52.7**	–
Image	AI2D	75.7	76.5	78.6	–	–	66.8	67.9	70.5
	ChartQA	76.5	77.8	76.4	–	–	17.4	51.8	70.7
	DocVQA	78.4	79.5	75.0	–	–	20.5	64.0	75.5
	InfoVQA	43.1	45.5	42.0	–	–	20.7	32.1	44.1
	MMBench-EN	77.2	78.5	79.6	–	–	68.2	70.9	71.6
	OCRBench	605	630	621	–	–	261	474	576
	OCRBench v2	26.3	26.1	26.1	–	–	–	–	24.4
	MMStar	52.1	54.3	55.0	–	–	43.8	45.7	48.3
	RealWorldQA	60.8	61.2	62.1	–	–	45.5	52.8	53.3

encoder, sampling mode, input resolution, and retained token budget. As a result, the comparison directly probes whether the compact abstract-token interface and the optional fine-grained token budget are sufficient for language-conditioned visual reasoning.

The multimodal results reveal a clear asymmetry. CoVisco is most effective on video benchmarks: abstract-only inference exposes about 400 visual tokens in the four-segment setting and remains competitive with references that pass over 10K tokens to the LLM; adding selected patches improves localized reasoning. Image-centric tasks remain weaker, consistent with the absence of OCR-specific pretraining. Codec results further separate encoder-side input sampling from LLM-side token compression, but do not by themselves measure wall-clock speed.

4.5 Token-budget and segment-count behavior

The reported budget sweep shows that abstract tokens and a small selected patch budget are complementary: abstract-only inference provides a compact baseline, while the top 20% setting gives the best observed video performance. The same pretrained encoder also generalizes from four to 8 and 16 segments at inference time; detailed results are reported in Appendix C.

4.6 Deployment interpretation and token accounting

The two budgets in CoVisco should be distinguished. The encoder-side budget counts sampled patch tokens processed by the ViT, whereas the LLM-side budget counts visual tokens forwarded after encoding. Abstract-only inference uses SQ tokens; with $Q = 100$, the four-, eight-, and sixteen-segment settings expose 400, 800, and 1,600 tokens, respectively.

Abstract-only inference is suited to segment-level semantics, global event structure, and retrieval-oriented representations. Abstract+top-K retains the same summaries and adds selected patch tokens when local evidence is needed; changing K changes the LLM sequence length without changing the pretrained ViT. Thus, K is a deployment-time accuracy–context knob.

Segmented attention and token selection address different costs: the former restricts encoder-side cross-segment attention, while the latter controls the visual context seen by the LLM. Our experiments measure the latter through token budgets and analyze the former through operation counts; reducing the LLM-side budget alone does not eliminate all encoder computation or establish a wall-clock speedup.

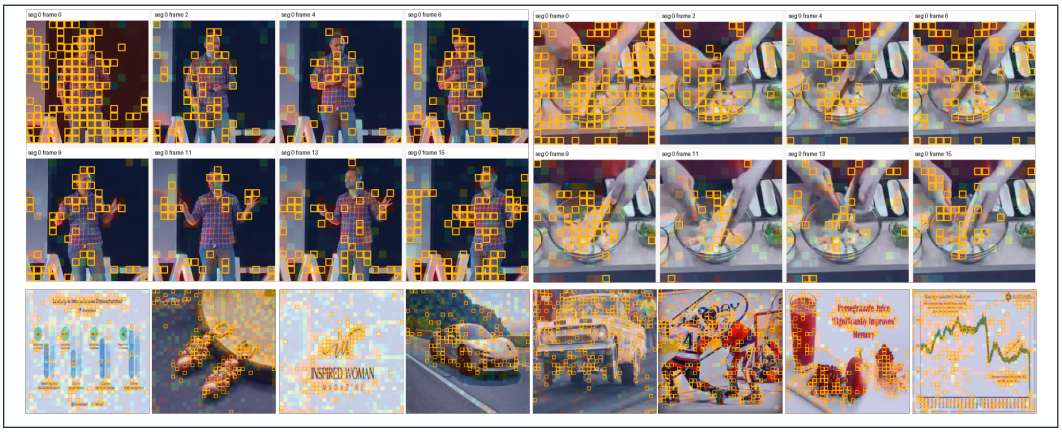

Figure 2: Qualitative visualization of the token selector on images and videos. Yellow boxes mark fine-grained patch tokens retained by the selector. The selected regions typically focus on objects, text, and localized changes, illustrating how selected patch tokens complement the abstract-token summaries.

4.7 Qualitative diagnostics

Qualitative selector visualizations are included in Figure 2. They are intended as diagnostics rather than quantitative evidence of selector optimality: selected patches typically focus on objects, text, and localized temporal changes, complementing the segment-level information carried by abstract tokens.

4.8 Discussion and limitations

The results support the central claim: abstract tokens provide a compact segment-level representation, while selected patch tokens restore local evidence when required. The interface is strongest for video understanding, where abstract-only inference is compact and additional patches improve localized reasoning.

The main limitations are the lack of OCR-specific pretraining, modest instruction-tuning scale, and the fixed $Q = 100$ capacity. Consequently, OCR and document results should be viewed as transfer measurements, and the best token budget remains task-dependent. A systematic capacity and end-to-end efficiency study is left for future work.

5 Conclusion

We introduced CoVisco, a unified image-video encoder with a native compact token interface. Abstract tokens learn segment-level representations inside the ViT, while fine-grained patch tokens remain available for downstream selection. Segmented attention routes cross-segment communication through abstract tokens, and deployment can vary from abstract-only to abstract+top-K inputs. Across the reported evaluations, this design provides a compact alternative to exposing all visual patches to the language model, with the clearest benefits on video-oriented tasks. Future work will improve OCR-oriented transfer and characterize end-to-end budget policies.

Impact statement

This work aims to reduce the computational cost of visual understanding, especially for video inputs whose token counts grow rapidly with duration and resolution. More efficient visual encoders can lower the cost of research and deployment, make long-video analysis more accessible, and reduce energy use for applications that currently rely on dense frame-level processing. These benefits also create the possibility of broader deployment, including in settings where video understanding may affect privacy, surveillance, moderation, or automated decision support. The model is evaluated only on research benchmarks and should not be treated as validated for high-stakes use. As with

other systems trained on large-scale web data, the learned representations may reflect dataset biases, uneven coverage across languages and cultures, and benchmark artifacts. We therefore view efficiency improvements as complementary to, not a replacement for, careful data governance, task-specific validation, and human oversight in sensitive applications.

AI USE STATEMENT

In this work, we used generative AI tools to assist with language editing, organization of manuscript prose, and LaTeX formatting and consistency checks. We have not used generative AI tools to generate experimental data, fabricate or alter experimental results, run experiments, or make final scientific claims; the remaining required disclosure tasks are not applicable to this work. Additionally, we used generative AI tools for recommended writing assistance, including drafting and revising explanatory text. We have reviewed all AI-assisted work, checked mathematical derivations and numerical values against our experimental records, and verified the final LaTeX source. We take responsibility for the final content of this work, including all text, claims, results, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work does not involve human participants, animal subjects, direct collection of personal or sensitive information, or interaction with individual users. The experiments use publicly available image–text and video data and standard evaluation benchmarks. We acknowledge that large-scale web data and multimodal benchmarks may contain societal biases and content with varying licensing and privacy properties; the data are used for research in accordance with their applicable terms. The proposed encoder is not evaluated for high-stakes decision making, and the reported benchmark results should not be interpreted as evidence of suitability for such applications.

REPRODUCIBILITY STATEMENT

Sections 3–3.6 describe the model architecture and token-routing mechanism, Section 3.5 specifies the contrastive objectives, target embeddings, and sampling modes, and Section 3.7 summarizes the datasets and four-stage training curriculum. The evaluation protocol and benchmark results are given in Section 4. Appendix A specifies the tensor organization and visual-token interfaces, Appendix C reports the segment-count generalization study, and Appendix B gives the corresponding attention-cost derivation. We will provide the implementation, training and evaluation configurations, and scripts needed to reproduce the reported experiments as anonymous supplementary material, subject to the licensing constraints of the underlying datasets and pretrained components.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025. URL https://arxiv.org/abs/2511.21631.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your ViT but faster. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://arxiv.org/abs/2210.09461.

Joya Chen, Zhaoyang Lv, Shiwei Wu, Kevin Qinghong Lin, Chenan Song, Difei Gao, Jia-Wei Liu, Ziteng Gao, Dongxing Mao, and Mike Zheng Shou. Videollm-online: Online video large language model for streaming video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18407–18418, 2024a.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *Computer Vision – ECCV 2024*, volume 15139 of *Lecture Notes in Computer Science*, pp. 19–35. Springer, 2024b. doi: 10.1007/978-3-031-73004-7_2.

Alex Fang, Albin Madappally Jose, Amit Jain, Ludwig Schmidt, Alexander Toshev, and Vaishaal Shankar. Data filtering networks. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=KAk6ngZ09F.

Kechen Fang, Yihua Qin, Chongyi Wang, Wenshuo Ma, Tianyu Yu, and Yuan Yao. LLaVA-UHD v4: What makes efficient visual encoding in MLLMs? *arXiv preprint arXiv:2605.08985*, 2026. URL https://arxiv.org/abs/2605.08985.

Gabriel Ilharco, Mitchell Wortsman, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. OpenCLIP. https://github.com/mlfoundations/open_clip, 2021.

Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 4651–4664. PMLR, 2021.

Weijian Jian, Yajun Zhang, Dawei Liang, Chunyu Xie, Yixiao He, Dawei Leng, and Yuhui Yin. RzenEmbed: Towards comprehensive multimodal retrieval. *arXiv preprint arXiv:2510.27350*, 2025. URL https://arxiv.org/abs/2510.27350.

Ziyan Jiang, Rui Meng, Xinyi Yang, Semih Yavuz, Yingbo Zhou, and Wenhu Chen. VLM2Vec: Training vision-language models for massive multimodal embedding tasks. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2410.05160.

Peng Jin, Ryuichi Takanobu, Wancai Zhang, Xiaochun Cao, and Li Yuan. Chat-univi: Unified visual representation empowers large language models with image and video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13700–13710, 2024a. doi: 10.1109/CVPR52733.2024.01300.

Yang Jin, Zhicheng Sun, Kun Xu, Kun Xu, Liwei Chen, Hao Jiang, Quzhe Huang, Chengru Song, Yuliang Liu, Di Zhang, Yang Song, Kun Gai, and Yadong Mu. Video-lavit: Unified video-language pre-training with decoupled visual-motional tokenization. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 22185–22209. PMLR, 2024b.

Kimi Team, Angang Du, Bohong Yin, Bowei Xing, Bowen Qu, Bowen Wang, Cheng Chen, Chenlin Zhang, Chenzhuang Du, Chu Wei, Congcong Wang, Dehao Zhang, Dikang Du, Dongliang Wang, Enming Yuan, Enzhe Lu, Fang Li, Flood Sung, Guangda Wei, Guokun Lai, Han Zhu, Hao Ding, Hao Hu, Hao Yang, Hao Zhang, Haoning Wu, Haotian Yao, Haoyu Lu, Heng Wang, Hongcheng Gao, Huabin Zheng, Jiaming Li, Jianlin Su, Jianzhou Wang, Jiaqi Deng, Jiezhong Qiu, Jin Xie, Jinhong Wang, Jingyuan Liu, Junjie Yan, Kun Ouyang, Liang Chen, Lin Sui, Longhui Yu, Mengfan Dong, Mengnan Dong, Nuo Xu, Pengyu Cheng, Qizheng Gu, Runjie Zhou, Shaowei Liu, Sihan Cao, Tao Yu, Tianhui Song, Tongtong Bai, Wei Song, Weiran He, Weixiao Huang, Weixin Xu, Xiaokun Yuan, Xingcheng Yao, Xingzhe Wu, Xinhao Li, Xinxing Zu, Xinyu Zhou, Xinyuan Wang, Y. Charles, Yan Zhong, Yang Li, Yangyang Hu, Yanru Chen, Yejie Wang, Yibo Liu, Yibo Miao, Yidao Qin, Yimin Chen, Yiping Bao, Yiqin Wang, Yongsheng Kang, Yuanxin Liu, Yuhao Dong, Yulun Du, Yuxin Wu, Yuzhi Wang, Yuzi Yan, Zaida Zhou, Zhaowei Li, Zhejun Jiang, Zheng Zhang, Zhilin Yang, Zhiqi Huang, Zihao Huang, Zijia Zhao, Ziwei Chen, and Zongyu Lin. Kimi-vl technical report. *arXiv preprint arXiv:2504.07491*, 2025. URL https://arxiv.org/abs/2504.07491.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 19730–19742. PMLR, 2023a.

Xianhang Li, Zeyu Wang, and Cihang Xie. CLIPA-v2: Scaling CLIP training with 81.1% zero-shot ImageNet accuracy within a $10,000 budget; an extra $4,000 unlocks 81.8% accuracy. *arXiv preprint arXiv:2306.15658*, 2023b. URL https://arxiv.org/abs/2306.15658.

Yanwei Li, Chengyao Wang, and Jiaya Jia. LLaMA-VID: An image is worth 2 tokens in large language models. In *Computer Vision – ECCV 2024*, volume 15104 of *Lecture Notes in Computer Science*, pp. 323–340. Springer, 2024. doi: 10.1007/978-3-031-72952-2_19.

Chenfei Liao, Wensong Wang, Zichen Wen, Xu Zheng, Yiyu Wang, Haocong He, Yuanhuiyi Lyu, Lutao Jiang, Xin Zou, Yuqian Fu, Bin Ren, Linfeng Zhang, and Xuming Hu. Are we using the right benchmark: An evaluation framework for visual token compression methods. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4236–4253. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.195.

Yue Liu, Christos Matsoukas, Fredrik Strand, Hossein Azizpour, and Kevin Smith. PatchDropout: Economizing vision transformers using patch dropout. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 3953–3962, 2023. doi: 10.1109/WACV56688.2023.00394.

Dmitrii Marin, Jen-Hao Rick Chang, Anurag Ranjan, Anish Prabhu, Mohammad Rastegari, and Oncel Tuzel. Token pooling in vision transformers for image classification. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 12–21, 2023.

Rui Meng, Ziyan Jiang, Ye Liu, Mingyi Su, Xinyi Yang, Yuepeng Fu, Can Qin, Zeyuan Chen, Ran Xu, Caiming Xiong, Yingbo Zhou, Wenhu Chen, and Semih Yavuz. VLM2Vec-V2: Advancing multimodal embedding for videos, images, and visual documents. *arXiv preprint arXiv:2507.04590*, 2025. URL https://arxiv.org/abs/2507.04590.

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. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 2021.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. DynamicViT: Efficient vision transformers with dynamic token sparsification. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pp. 13937–13949, 2021. URL https://arxiv.org/abs/2106.02034.

Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. EVA-CLIP: Improved training techniques for CLIP at scale. *arXiv preprint arXiv:2303.15389*, 2023. URL https://arxiv.org/abs/2303.15389.

Feilong Tang, Xiang An, Yunyao Yan, Yin Xie, Bin Qin, Kaicheng Yang, Yifei Shen, Yuanhan Zhang, Chunyuan Li, Shikun Feng, Changrui Chen, Huajie Tan, Ming Hu, Manyuan Zhang, Bo Li, Ziyong Feng, Ziwei Liu, Zongyuan Ge, and Jiankang Deng. Onevision-encoder: Codec-aligned sparsity as a foundational principle for multimodal intelligence. *arXiv preprint arXiv:2602.08683*, 2026. URL https://arxiv.org/abs/2602.08683.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier Hénaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025. URL https://arxiv.org/abs/2502.14786.

Zichen Wen, Yifeng Gao, Weijia Li, Conghui He, and Linfeng Zhang. Token pruning in multimodal large language models: Are we solving the right problem? In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 15537–15549. Association for Computational Linguistics, 2025a. doi: 10.18653/v1/2025.findings-acl.802.

Zichen Wen, Yifeng Gao, Shaobo Wang, Junyuan Zhang, Qintong Zhang, Weijia Li, Conghui He, and Linfeng Zhang. Stop looking for "important tokens" in multimodal language models: Duplication

matters more. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 9961–9980, Suzhou, China, 2025b. Association for Computational Linguistics. doi: 10.18653/v1/2025.emnlp-main.505.

Chao-Yuan Wu, Manzil Zaheer, Hexiang Hu, R. Manmatha, Alexander J. Smola, and Philipp Krähenbühl. Compressed video action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6026–6035, 2018. doi: 10.1109/CVPR.2018.00631.

Hu Xu, Saining Xie, Xiaoqing Ellen Tan, Po-Yao Huang, Russell Howes, Vasu Sharma, Shang-Wen Li, Gargi Ghosh, Luke Zettlemoyer, and Christoph Feichtenhofer. Demystifying CLIP data. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2309.16671.

Senqiao Yang, Kaichen Zhang, Zhaoyang Jia, Jinghao Guo, Yifei Shen, Xinjie Zhang, Xiaoyi Zhang, Haoqing Wang, Xiao Li, Peng Zhang, Xiang An, Yin Xie, Zhening Liu, Xun Guo, Jiahao Li, Shicheng Zheng, Jinglu Wang, Zongyu Guo, Wenxuan Xie, Zihan Zheng, Yuxuan Luo, Bin Li, and Yan Lu. Mage-vl: An efficient codec-native streaming multimodal foundation model. *arXiv preprint arXiv:2607.24904*, 2026. URL https://arxiv.org/abs/2607.24904.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11975–11986, 2023. doi: 10.1109/ICCV51070.2023.01100.

Haoji Zhang, Yiqin Wang, Yansong Tang, Yong Liu, Jiashi Feng, and Xiaojie Jin. Flash-vstream: Efficient real-time understanding for long video streams. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 21059–21069, 2025a.

Xin Zhang, Yanzhao Zhang, Wen Xie, Mingxin Li, Ziqi Dai, Dingkun Long, Pengjun Xie, Meishan Zhang, Wenjie Li, and Min Zhang. Bridging modalities: Improving universal multimodal retrieval by multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9274–9285, 2025b. doi: 10.1109/CVPR52734.2025.00866.

Zijia Zhao, Yuqi Huo, Tongtian Yue, Longteng Guo, Haoyu Lu, Bingning Wang, Weipeng Chen, and Jing Liu. Efficient motion-aware video MLLM. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24159–24168, 2025.

A Detailed model architecture

This appendix makes the tensor organization and output interfaces explicit. The same encoder is used for images and videos. An image is represented by one segment ($S = 1$), whereas a video is divided into S temporal segments. Let m be the number of visible patch tokens in one segment, Q the number of abstract tokens, and $d = 1024$ the hidden dimension. After patch embedding and sampling, the visual input is represented as

$$\mathbf{X} = [\mathbf{X}_1, \ldots, \mathbf{X}_S], \qquad \mathbf{X}_s \in \mathbb{R}^{m \times d}. \tag{2}$$

The abstract tokens are a single shared parameter matrix, rather than a separate parameter set for every segment:

$$\mathbf{Q} \in \mathbb{R}^{Q \times d}, \qquad \mathbf{H}_s^{(0)} = [\,\mathbf{Q} \,\|\, \mathbf{X}_s\,] \in \mathbb{R}^{(Q+m) \times d}. \tag{3}$$

The matrix $\mathbf{Q}$ is expanded along the segment and batch dimensions at runtime. Thus, the segments have independent token positions at the input but share the same abstract-token parameterization. With the default $Q = 100$, the compact representation exposed by abstract-only inference contains SQ tokens.

A.1 Attention routing and positional inputs

The ViT contains 24 alternating transformer layers. In an intra-segment layer, the attention mask is block diagonal over segments: a token in segment s can read the $m + Q$ tokens in that segment only. In an abstract-communication layer, a query in segment s can read its m local patch tokens and the SQ abstract tokens from all segments. Equivalently, writing $\mathcal{P}_s$ for the patch-token indices of segment s and $\mathcal{A}$ for all abstract-token indices, the visible key sets are

$$\mathcal{K}_s^{\text{intra}} = \mathcal{P}_s \cup \mathcal{A}_s, \qquad \mathcal{K}_s^{\text{comm}} = \mathcal{P}_s \cup \mathcal{A}. \tag{4}$$

This routing allows every segment to write local evidence into its abstract tokens and allows all segments to exchange information through the abstract channel, while disallowing direct cross-segment patch-to-patch attention. The 4D rotary embedding receives the segment, temporal, height, and width coordinates of each visible token. Patch coordinates are gathered from the full virtual grid after sampling, while abstract tokens use the reserved summary positions specified in Section 3.4.

A.2 Output paths and downstream interfaces

After the final transformer layer, the hidden states are split into

$$\mathbf{A} \in \mathbb{R}^{S\times Q\times d}, \qquad \mathbf{P} \in \mathbb{R}^{S\times m\times d}, \tag{5}$$

where $\mathbf{A}$ contains the abstract tokens and $\mathbf{P}$ contains the fine-grained patch tokens. Only $\mathbf{A}$ is sent to the attention-pooling head during contrastive pretraining. The pooled representation is then mapped by three separate fully connected heads to the image–image (1536-dimensional), image–text (1536-dimensional), and video–text (4096-dimensional) target spaces. Each head has its own learnable logit scale and bias and is trained with its corresponding SigLIP sigmoid loss. The image–video-caption stream uses the video–text projection space, so it does not introduce a fourth projection head.

The patch output $\mathbf{P}$ is not pooled away. During the selector stage, a lightweight selector ranks these tokens and exposes a runtime-selected subset of them. The selected visual tokens, either $\mathbf{A}$ alone or $\mathbf{A}$ concatenated with selected tokens from $\mathbf{P}$, are passed through the MLP projector used by the multimodal language model. The ViT is frozen during this alignment stage; the selector, projector, and language model are trained according to the curriculum in Section 3.7. At inference, changing K changes only the number of fine-grained tokens exposed to the language model and does not alter the encoder computation or its pretrained parameters.

A.3 Token selector details

The selector receives the final ViT representations with the segment dimension preserved. For a batch of inputs, its inputs are

$$\mathbf{A} \in \mathbb{R}^{B\times S\times Q\times d}, \qquad \mathbf{P} \in \mathbb{R}^{B\times S\times P\times d}, \tag{6}$$

where $\mathbf{A}$ denotes the per-segment abstract/query tokens and $\mathbf{P}$ denotes the corresponding fine-grained patch tokens. Selection is performed independently within each segment: the (B, S) dimensions are flattened temporarily, so a segment never competes directly with patches from another segment.

The default selector is a two-layer scoring transformer. For each layer l, let $\mathbf{Z}_s^{(l)}$ denote the patch-token features for segment s, and let $\mathbf{A}_s$ denote its abstract tokens. The layer applies

$$\widetilde{\mathbf{Z}}_s^{(l)} = \mathrm{LN}\Big(\mathbf{Z}_s^{(l)} + \mathrm{SelfAttn}(\mathbf{Z}_s^{(l)})\Big), \tag{7}$$

$$\widehat{\mathbf{Z}}_s^{(l)} = \mathrm{LN}\Big(\widetilde{\mathbf{Z}}_s^{(l)} + \mathrm{CrossAttn}(\widetilde{\mathbf{Z}}_s^{(l)}, \mathbf{A}_s, \mathbf{A}_s)\Big), \tag{8}$$

$$\mathbf{Z}_s^{(l+1)} = \mathrm{LN}\Big(\widehat{\mathbf{Z}}_s^{(l)} + \mathrm{FFN}(\widehat{\mathbf{Z}}_s^{(l)})\Big). \tag{9}$$

Here patch tokens are the queries in the cross-attention, while the abstract/query tokens provide keys and values. The default implementation uses 8 attention heads, a $4d$ feed-forward hidden dimension when not otherwise specified, GELU activation, and zero dropout. The layer count remains a configuration parameter, with two layers used as the default selector configuration.

After the second layer, a final LayerNorm and a linear score head produce one logit per patch token:

$$\ell_{s,i} = \mathrm{Linear}\Big(\mathrm{LN}(\mathbf{Z}_{s,i}^{(2)})\Big), \qquad p_{s,i} = \sigma(\ell_{s,i}). \tag{10}$$

The default sigmoid score treats each patch as having an independent keep probability. Invalid padded positions are assigned $-\infty$ logits and cannot be selected. For each segment, the selector then chooses the K patches with largest scores, returning their indices, the full score map, and the selected features. In the implementation, the scores are computed from the contextualized selector features, while the selected feature values are gathered from the corresponding original ViT patch tokens.

Hard top-K selection is non-differentiable. Following the score-gating idea of DynamicViT (Rao et al., 2021), the selected features are multiplied by their continuous scores during training, which provides a straight-through gradient path to the scoring network while retaining hard top-K indices in the forward pass. An optional Gumbel mode perturbs the logits during training for stochastic exploration; evaluation uses deterministic scores. The selector therefore returns $\mathbf{P}_{\text{sel}} \in \mathbb{R}^{B\times S\times K\times d}$ and indices with shape (B, S, K), which are concatenated with $\mathbf{A}$ in the abstract-plus-top-K deployment mode and then passed to the MLP projector.

B Attention-cost analysis details

Complexity scope. The attention-cost expressions below characterize encoder-side attention operations. They should be read separately from the LLM-side token budgets reported in the main experiments and do not constitute a wall-clock latency measurement. Let N be the number of patch tokens before adding abstract tokens, S the number of temporal segments, $m = N/S$ the number of patch tokens per segment, and Q the number of abstract tokens prepended to each segment. Each segment therefore has $n = m + Q$ tokens.

A dense ViT layer over the original patch sequence costs $O(N^2)$ attention operations per layer. If the same dense attention were applied after adding all SQ abstract tokens, the cost would become

$$O\big((N + SQ)^2\big) = O\big(N^2 + 2NSQ + S^2Q^2\big)\,. \tag{11}$$

This gives every patch token direct access to every other patch token, but it is precisely the pattern that becomes prohibitive for long videos.

CoVisco alternates two cheaper patterns. In intra-segment layers, each segment attends only within its own length-n sequence, giving

$$O(Sn^2) = O\big(S(m + Q)^2\big) = O\left(\frac{N^2}{S} + 2NQ + SQ^2\right). \tag{12}$$

In abstract-communication layers, each token attends to its local segment patch tokens and to the abstract tokens from all segments. Each of the $S(m + Q)$ query tokens therefore attends to $m + SQ$ keys, giving

$$O(S(m + Q)(m + SQ)) = O\left(\frac{N^2}{S} + NSQ + NQ + S^2Q^2\right). \tag{13}$$

The leading dense patch-to-patch term is reduced from N^2 to N^2/S, while all global communication is restricted to the abstract-token channel. When When segment length is fixed and longer videos are handled by increasing S, the per-segment patch count m and abstract count Q remain constant. The segmented design removes dense all-to-all patch interaction across segments, but the abstract-communication layers still incur additional costs because each segment attends to the abstract tokens of all segments. Thus, the method trades dense cross-segment patch attention for local segment computation plus a compact global communication channel. This design avoids forming an all-to-all patch sequence as video duration increases, while its actual scaling depends on the number of segments, the per-segment token count, and the abstract-token budget.

The analysis isolates attention operations and does not include implementation-dependent constants such as FlashAttention kernel efficiency, memory layout, or data-loading overhead. Those constants affect wall-clock latency and should therefore be evaluated separately from the operation-count comparison above.

C Generalization across segment counts

The same pretrained visual encoder is evaluated with different segment counts at inference time. Table 4 reports results on two long-video understanding benchmarks using the abstract-only deployment mode. The language model is SFT-tuned with four segments, whereas evaluation changes the

Table 4: Long-video understanding under different segment configurations. The language model is SFT-tuned with four segments; evaluation changes the number of segments at inference time without retraining. All configurations use the abstract-only mode. The abstract-token budget is $S \times Q$ with $Q = 100$.

Benchmark	Resolution	Frames	Segments	sampling	Mode	Accuracy (%)
MLVU	224	64	4	Frame	Abstract-only	61.2
MLVU	224	128	8	Frame	Abstract-only	62.6
MLVU	224	128	16	Frame	Abstract-only	62.3
MLVU	224	256	16	Frame	Abstract-only	61.3
MLVU	224	256	16	codec	Abstract-only	59.9
LongVideoBench-Val	224	64	4	Frame	Abstract-only	47.3
LongVideoBench-Val	224	128	8	Frame	Abstract-only	49.9
LongVideoBench-Val	224	128	16	Frame	Abstract-only	50.2
LongVideoBench-Val	224	256	16	Frame	Abstract-only	50.3
LongVideoBench-Val	224	256	16	codec	Abstract-only	52.4

number of segments without retraining the visual encoder or language model. The model maintains comparable performance when the same video duration is partitioned into different numbers of segments and remains effective when the input length is increased to 256 frames. This study is intended as a generalization analysis of the token interface, not as an architectural ablation of the attention patterns.

Each configuration exposes $S \times Q$ abstract tokens to the language model in abstract-only mode: 400, 800, and 1,600 tokens for the reported 4-, 8-, and 16-segment settings, respectively. The results therefore test whether the same compact interface remains useful as temporal coverage and its corresponding LLM-side budget increase.

D Controlled evaluation protocol

The controlled comparison separates the contribution of the visual encoder from that of the language backbone and instruction data. Each path uses the same Qwen3-4B-Instruct-2507 language model, the same MLP projector family, and the same instruction-tuning data described in Section 3.7. The paths differ only in the visual encoder and the visual-token interface exposed to the language model. This design is intended to answer a narrow question: for a fixed decoder and fixed multimodal training recipe, how much performance is retained when dense frame tokens are replaced by a compact abstract-token interface plus an optional selected patch budget?

Path A uses CoVisco with segmented abstract tokens and, where specified, abstract+top-K fine-grained tokens. Path B replaces the visual encoder with a codec-native OneVision-Encoder-style baseline under matched sampling and token budgets. Path C uses a frame-based SigLIP2-style encoder under uniform sampling. For fair comparison, reported token budgets count only visual tokens presented to the language model, not internal tokens used transiently inside the frozen vision encoder. Codec and frame variants are reported separately because they make different assumptions about access to compressed-video signals.

D.1 Qualitative diagnostics of selected fine-grained tokens

Qualitative selector visualizations are provided above in Figure 2. They are intended as diagnostics rather than quantitative evidence of selector optimality: selected patches concentrate on objects, text, and localized temporal changes, complementing the segment-level information carried by abstract tokens.

The long-video walkthrough follows the same principle at the largest temporal scale evaluated in this work. A 256-frame input is partitioned into 16 segments, each represented by its abstract tokens and, optionally, a small selected patch budget. Abstract-only inference should retain the overall event structure but may miss small text or brief local actions, whereas abstract+top-K inference should recover more localized evidence without exposing the full patch sequence to the language model.