OneVision-Encoder: Codec-Aligned Sparsity as a Foundational Principle for Multimodal Intelligence

Feilong Tang[1,4*] **Xiang An**[1,*] **Yunyao Yan**[2,*] **Yin Xie**[2,*] **Kaicheng Yang**[2] **Yifei Shen**[5] **Yuanhan Zhang**[1] **Shikun Feng**[5] **Chunyuan Li**[1] **Changrui Chen**[3] **Huajie Tan**[6] **Ming Hu**[4] **Manyuan Zhang**[1] **Bo Li**[1†] **Ziyong Feng**[2,†] **Ziwei Liu**[7,†] **Zongyuan Ge**[4,†] **Jiankang Deng**[3,†]

[1]LMMs-Lab [2]Glint Lab [3]MVP Lab [4]Monash University
[5]Zhongguancun Academy [6]Peking University [7]Nanyang Technological University

Abstract

Visual signals, especially videos, are highly redundant: most regions are temporally predictable, while informative changes are sparse. We hypothesize that vision encoders should allocate computation according to this non-uniform information density rather than process dense pixel grids uniformly. From this perspective, codec-derived motion and residual signals provide a natural basis for identifying informative regions. OneVision-Encoder instantiates this idea through Codec Patchification, which replaces uniform dense computation with selective processing of only 3.1%-25% of regions that carry high signal entropy. To support irregular spatiotemporal token layouts, OneVision-Encoder employs a shared 3D RoPE and is pretrained with a large-scale cluster discrimination objective over more than one million semantic concepts, enabling unified representation learning for appearance and motion. Empirically, OV-Encoder improves both efficiency and accuracy under fixed token budgets. When integrated into large multimodal models, it achieves a 4.1% average improvement over Qwen3-ViT on video understanding benchmarks while remaining competitive on image and document understanding. Under attentive probing, it further yields substantial gains on motion-sensitive benchmarks, improving Top-1 accuracy on Diving48 by 17.1 and 8.1 percentage points over SigLIP2 and DINOv3, respectively, at matched patch budgets. These results suggest that codec-guided patch sparsity is an effective design principle for scalable visual representation learning.

Code: `https://github.com/EvolvingLMMs-Lab/OneVision-Encoder`
Data: `https://github.com/EvolvingLMMs-Lab/OneVision-Encoder/blob/main/docs/data_card.md`
Model: `https://huggingface.co/collections/lmms-lab-encoder/onevision-encoder`

1 Introduction

Transformer-based models have advanced video understanding [17, 89, 79, 6, 76, 99] by representing videos as token sequences capable of modeling long-range spatial and temporal dependencies [90, 80]. In self-supervised representation learning, reconstruction-based objectives such as MAE [41] and V-JEPA [10] capture pixel- or feature-level regularities, while contrastive paradigms such as CLIP [70] and SigLIP [103] induce semantic separation through instance-level, often language-supervised alignment. Recent cluster-discrimination methods [2, 4, 94, 85] further encourage semantically related samples to form coherent groups, improving structured object-centric representation learning. Nevertheless, most video transformers construct representations from sparsely sampled full frames, producing dense token sequences that assign similar computational status to informative motion, static context, and redundant regions. This frame-centric assumption motivates a more explicit account of which visual evidence should be encoded in naturally redundant video signals.

*Equal Contribution.
†Project leads.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

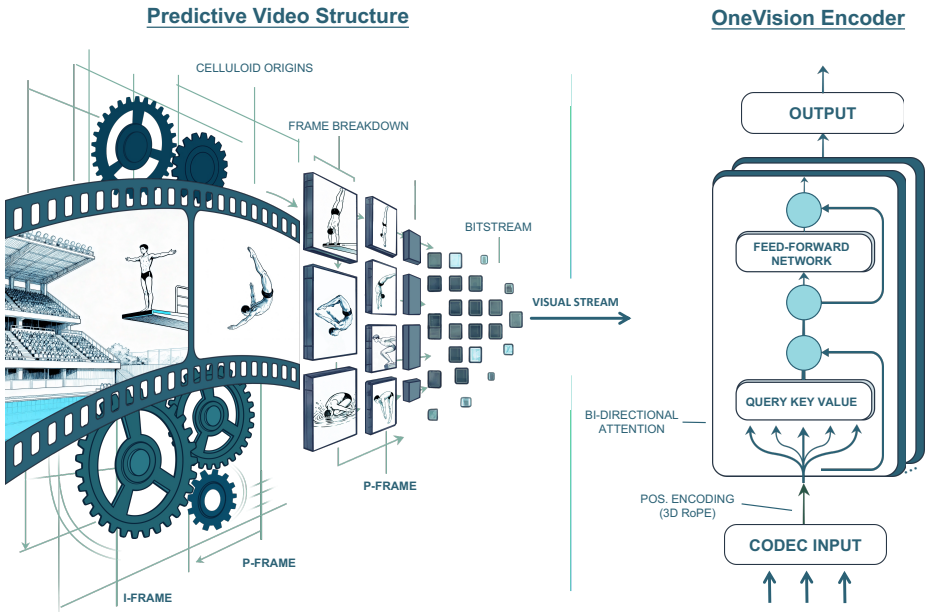


Figure 1: **Visual intelligence as codec-aligned predictive compression.** Video exemplifies this principle: most visual content is redundant and predictable, while meaningful information arises sparsely as motion and residual change. Video codecs make this structure explicit by decomposing visual signals into stable spatial context and sparse temporal updates. Grounded in this codec principle, OV-Encoder reframes visual modeling as predictive compression, serving as a scalable engine for universal multimodal intelligence that sees, updates, and reasons over time.

Visual intelligence can be viewed through compression: natural videos exhibit strong spatial and temporal regularities, so most content is predictable context rather than new discriminative evidence. Yet standard video pretraining applies uniform computation to dense pixel grids, spending capacity on static or easily inferred regions while the informative *surprise* remains sparse. Modern codecs make this signal structure explicit: H.264 and H.265/HEVC *(High Efficiency Video Coding)* encode global context with intra-coded frames (I-frames) and inter-frame changes with predicted frames (P-frames), motion compensation, and residuals [82]. This decomposition suggests that meaningful visual evidence is localized and motion-dependent, not a dense field of equally informative pixels.

These observations lead to a unified conclusion: to scale visual intelligence, architectures must align with the information-theoretic structure of the data. In this work, we present **OneVision-Encoder (OV-Encoder)**, a HEVC-style Vision Transformer that aligns spatiotemporal representation learning with the intrinsic predictive structure of video signals, as illustrated in Figure 1. Rather than uniformly processing dense pixel grids, OV-Encoder explicitly determines which visual signals constitute independent evidence and selectively encodes only the regions rich in signal entropy. To enable this, we introduce **Codec Patchification**, a codec-inspired input formulation that leverages temporal signals exposed by video codecs to organize visual tokens at the patch level, together with a 3D Rotary Position Embedding (RoPE) [81] that jointly encodes spatial and temporal positions to support coherent attention over irregular spatiotemporal layouts. Specifically, **(a) Dense Video-Codec Patchification**: a codec-inspired video encoding formulation that leverages motion-centric temporal signals exposed by P-frames to patchify selected visual regions (3.1%-25%) in dense video inputs, while preserving dense temporal coverage. **(b) Chunk-wise Patchification**: a codec-inspired temporal patchification scheme that partitions video streams into fixed-length chunks and constructs patch-level representations with chunk-level positional encoding. **(c) Single-Image Spatial Patchification**: a spatial instantiation of Codec Patchification that constructs patch-level representations for single-image input, enabling structured modeling of static visual content. Furthermore, explicitly modeling visual evidence at the patch level requires a training objective that enforces semantic structure. We adopt a self-supervised cluster discrimination objective based on large-scale semantic clustering over a *concept bank with more than one million clusters*, jointly capturing object-level permanence and

motion dynamics. In particular, OV-Encoder provides a bi-directional attention-based vision encoder that effectively supports image and video understanding.

Extensive experiments validate OV-Encoder under both multimodal transfer and representation-level evaluation. For SigLIP2 [86], we compare all models under a unified multimodal adaptation protocol using a 1.5M-scale LLaVA-Next [60] and LLaVA-Next-Video [106] instruction corpus with native-resolution processing. Under this setting, Qwen3-4B-based multimodal models built on OV-Encoder achieve stronger performance than SigLIP2 on a 16-benchmark suite spanning video, image, and document understanding.

For Qwen3-ViT [8], we adopt a controlled protocol. Specifically, OV-Encoder is first adapted to native-resolution inputs with Qwen3-1.7B under the LLaVA-OV-1.5 framework [3], including both Stage 1 and Stage 1.5, and is then decoupled for comparison under the same LLaVA-Next-Video instruction-tuning setting with a Qwen3-4B-based LMM. Despite being pretrained on substantially fewer vision-language tokens (approximately 100B versus more than 2.1T caption and instruction-aligned tokens), OV-Encoder achieves stronger overall performance than Qwen3-ViT on the same 16-benchmark suite. Under attentive probing on seven benchmarks, OV-Encoder further demonstrates strong representation quality, including gains of 17.1 and 8.1 Top-1 points over SigLIP2 and DINOv3 [78], respectively, on Diving48 at matched patch budgets. It also remains competitive with dense-patch baselines such as DINOv3, MetaCLIP2 [23], and AIMv2 [33]. We release the data, training protocols, and model checkpoints to support transparent and reproducible vision-language research.

Our contributions are as follows. *(i)* We present OneVision-Encoder (OV-Encoder), a codec-guided vision encoder that aligns spatiotemporal representation learning with the predictive structure of video through sparse codec-aware patch encoding. *(ii)* We introduce Codec Patchification, a codec-inspired input formulation that uses codec-derived temporal cues to select only 3.1%-25% informative patches from dense video, while unifying codec-sparse video, chunk-wise video, and single-image inputs through a shared 3D RoPE. *(iii)* We adopt a self-supervised cluster discrimination objective over a large-scale concept bank to jointly model appearance and motion semantics. *(iv)* Extensive experiments across multimodal transfer and attentive probing demonstrate the effectiveness of OV-Encoder for both multimodal understanding and visual representation learning.

2 Approach

Although most previous video encoders focus on short clips of 16 frames (roughly seconds) [11, 89], we explore training with longer clips of up to 64 frames at higher spatial resolutions. Let $\mathcal{V}_i = \{\mathcal{I}_{i,t}\}_{t=1}^{T_i}$ denote the i-th input video of spatial size $H \times W$, where T_i is the total number of frames in $\mathcal{V}_i$. Our objective is to process raw video inputs within the proposed different input configurations and jointly encode them using a shared ViT for unified spatiotemporal representation learning.

2.1 HEVC Guided Patch Selection

Codec Based Video Factorization. Following the standard High Efficiency Video Coding formulation [82], each video $\mathcal{V}_i$ is divided into N_i Groups of Pictures (GOP), $\{\mathcal{S}_{i,n}\}_{n=1}^{N_i}$. Applying the HEVC codec to each Groups of Pictures yields one intra-coded frame and $(K_{i,n}-1)$ predicted frames:

$$\left(F_{i,n}^{\mathrm{I}},\ \{F_{i,n,\tau}^{\mathrm{P}}\}_{\tau=1}^{K_{i,n}-1}\right)\ =\ \mathcal{C}_{\mathrm{HEVC}}(\mathcal{S}_{i,n}), \tag{1}$$

where $F_{i,n}^{\mathrm{I}} \in \mathbb{R}^{H\times W\times C_{\text{img}}}$ is the I-frame (RGB, $C_{\text{img}} = 3$), and each $F_{i,n,\tau}^{\mathrm{P}}$ denotes a P-frame, which is represented in the bitstream by motion vectors and a residual signal. $K_{i,n}$ denotes the GOP length.

Motion and Residual Signals. In HEVC, motion is represented by motion vectors $\boldsymbol{d}_{i,n,\tau}$ that encode block level displacements between the current frame and its motion compensated prediction from the reference frame. Concretely, P-frames are partitioned into coding units (CUs) with variable sizes ranging from 4×4 to 64×64, and all pixels within a CU share the same motion vector. To align codec signals with ViT patchification, we first broadcast each CU motion vector to its covered pixels, obtaining a dense pixel level motion field. The magnitude $\|\boldsymbol{d}_{i,n,\tau}\|_2$ reflects the intensity of local motion, with larger values indicating stronger or more complex dynamics. In addition to motion, each P-frame is associated with a residual signal that captures appearance changes not explained by motion compensation; we decode the luma residual into the pixel domain and measure its energy as a complementary cue for unpredictable visual variation. At the patch level, we aggregate motion

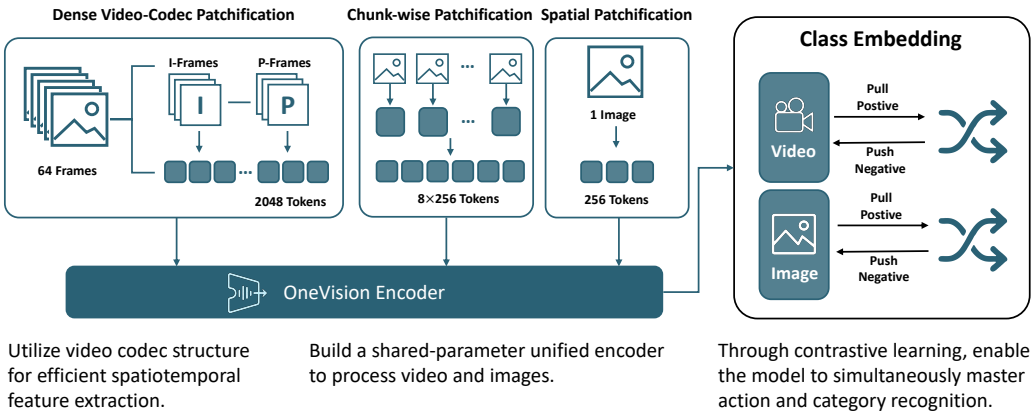


Figure 2: **Overview of the OneVision-Encoder framework. Left:** Input formulation. The framework integrates three Codec Patchification strategies: Dense Video-Codec Patchification, Chunk-wise Patchification, and (Single-Image/Frame) Spatial Patchification. All inputs are processed by a shared-parameter OneVision-Encoder. **Right:** Unified cluster discrimination objective. Image and video embeddings are aligned through contrastive learning against a global set of class centers, jointly optimizing object-level and action-level representations within a single encoder.

magnitude and residual energy over all pixels inside each ViT patch, so the two signals jointly characterize the amount of new visual evidence introduced by a region. We use these codec exposed signals as a principled proxy for unpredictability, enabling the identification of salient regions that contribute new spatiotemporal information.

Sparse Patch Selection. Let p denote the patch size (*e.g.*, p=14). We define the patch grid $\mathcal{G} = \{(y, x) \mid 0 \le y < H/p,\ 0 \le x < W/p\}$, with cardinality $P_0 = (H/p)(W/p)$. For each P-frame, we compute a patch level saliency score by aggregating the codec exposed motion magnitude and residual energy defined above. Based on the aggregated saliency score, we construct a binary mask $\Omega_{i,n,\tau} \subseteq \mathcal{G}$ by selecting a fixed proportion of the most salient patches. The sparsity ratio is fixed throughout training and inference by selecting a fixed proportion r of the most salient patches, i.e., $|\Omega_{i,n,\tau}| = \lfloor rP_0 \rfloor$.

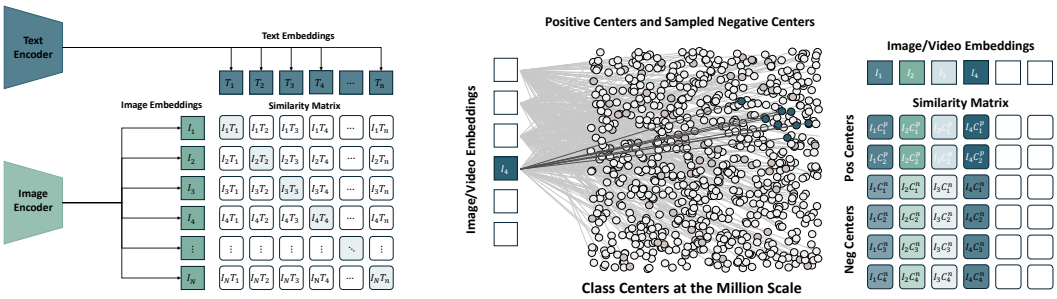
2.2 Codec Patchification

Dense Video-Codec Patchification. Following the codec formulation, each video $\mathcal{V}_i$ is partitioned into N_i GOP. For the n-th GOP, the HEVC encoder produces one intra-coded frame $F^{\mathrm{I}}_{i,n}$ and $(K_{i,n} - 1)$ predicted frames $\{F^{\mathrm{P}}_{i,n,\tau}\}_{\tau=1}^{K_{i,n}-1}$ that are defined by motion vectors and a residual signal after motion compensation. Let $\Omega_{i,n,\tau}$ denote the codec-derived binary mask that selects dynamically informative patches based on motion magnitude and residual energy, and let $\Pi_p(\cdot)$ denote patchification with patch size p. The HEVC-compressed input sequence is defined as

$$\mathcal{F}^{(\mathrm{hevc})}_{i,n} \triangleq \Big[\Pi_p(F^{\mathrm{I}}_{i,n}) \oplus \big\{\Pi_p(\bar{F}^{\mathrm{P}}_{i,n,\tau})[\Omega_{i,n,\tau}]\big\}_{\tau=1}^{K_{i,n}-1}\Big], \tag{2}$$

where $\bar{F}^{\mathrm{P}}_{i,n,\tau} \in \mathbb{R}^{H\times W\times C_{\mathrm{img}}}$ denotes the decoded RGB P-frame in the pixel domain obtained by decoding the HEVC bitstream. The binary mask $\Omega_{i,n,\tau}$ is computed from the codec-exposed motion vectors $\boldsymbol{d}_{i,n,\tau}$ and the associated motion-compensated residual signal of the same P-frame, and is used only for salient patch selection. Therefore, the tokens fed to the encoder for P-frames are RGB patches from $\bar{F}^{\mathrm{P}}_{i,n,\tau}$ indexed by $\Omega_{i,n,\tau}$. $K_{i,n}$ is the GOP length. $[\Omega]$ denotes masked patch selection along the patch dimension. $\oplus$ denotes concatenation along the token dimension. Let $M_{i,n}$ be the number of tokens in $\mathcal{F}^{(\mathrm{hevc})}_{i,n}$; the total tokens over the video satisfy:

$$M_i = \sum_{n=1}^{N_i} M_{i,n} = \sum_{n=1}^{N_i}\Big(P_0 + \sum_{\tau=1}^{K_{i,n}-1} |\Omega_{i,n,\tau}|\Big), \tag{3}$$



(a) Standard contrastive learning methods

(b) Cluster discrimination learning methods

Figure 3: **Contrastive learning vs. cluster discrimination.** (a) Standard contrastive learning contrasts samples against batch-local negatives, constraining the view of the embedding space. (b) Cluster discrimination contrasts samples against a global concept bank of clustered centers at scale, yielding discriminative and structurally separated representations.

which yields a pixel/token compression ratio per GOP, $\gamma_{i,n} = 1 - M_{i,n}/(K_{i,n}P_0)$. Under our default setting (64 frames, GOP size 32, token budget 2048, $P_0 = 256$), the overall clip-level token reduction is $1 - 2048/(64P_0) = 87.5\%$.

Chunk-wise Patchification. To unify Codec patchification with sparse temporal sampling, each video $\mathcal{V}_i$ is uniformly partitioned into C temporal chunks of $\lfloor T_i/C \rfloor$ consecutive frames. From every chunk, one frame is randomly sampled, resulting in a temporally stratified subsequence $\pi_i = \{t_c\}_{c=1}^{C}$, where $t_c \in [(c-1)\lfloor T_i/C \rfloor,\ c\lfloor T_i/C \rfloor)$. The corresponding chunk-wise sampling sequence is defined as

$$\mathcal{F}_i^{(\text{chunk})} \triangleq \big\{\Pi_p(\mathcal{I}_{i,t_c}) \mid t_c \in \pi_i\big\} = \Pi_p(\mathcal{V}_i[\pi_i]). \tag{4}$$

Single-Image Spatial Patchification. To achieve spatial scalability, each image $\mathcal{I}_{i,t} \in \mathbb{R}^{H\times W\times C_{\text{img}}}$ is processed independently as a static image input. Each image is directly patchified in a row-wise manner from top to bottom to ensure a deterministic spatial ordering of visual tokens, preserving the original spatial layout. The resulting patch sequence is defined as

$$\mathcal{F}_i^{(\text{image})} \triangleq \big\{\Pi_p(\mathcal{I}_{i,t}) \mid t = 1, \ldots, T_i\big\}, \tag{5}$$

where $\Pi_p(\cdot)$ denotes the patchification operator applied to a single frame, and $T_i = 1$ for single-image inputs. A detailed spatial bias analysis is provided in the supplementary material.

Tokenization and Transformer Encoding. After generating the three types of input sequences, namely the Codec Patchification formulation, we uniformly tokenize and encode them using a shared Transformer backbone $\phi(\cdot)$. For each video input $\text{vid} \in \{\text{hevc}, \text{chunk}, \text{image}\}$, the token sequence $\mathcal{F}_i^{\text{vid}} = \{x_{i,k}^{\text{vid}}\}_{k=1}^{M_i^{\text{vid}}}$ is processed by the encoder $\phi(\cdot)$ followed by a linear projection head $W \in \mathbb{R}^{d\times D}$ where d and D denote hidden and latent dimensions of the encoder, respectively, to obtain latent embeddings:

$$E_i^{\text{vid}} = \phi\big(\mathcal{F}_i^{\text{vid}}\big)W = Z_i^{\text{vid}}W. \tag{6}$$

which defines $Z_i^{\text{vid}} = \phi(\mathcal{F}_i^{\text{vid}}) \in \mathbb{R}^{M_i^{\text{vid}}\times d}$ as the encoder output features before projection. Finally, a function $f(\cdot)$, such as attentive pooling, integrates all token embeddings into a compact video-level representation $e_i^{\text{vid}} = f(E_i^{\text{vid}}) \in \mathbb{R}^D$.

2.3 Training Objective

Visual samples commonly exhibit multiple semantic components, including object-level semantics from images and motion-level semantics from videos, rendering single-label assignments inadequate for unified representation learning. We therefore supervise visual embeddings with an offline concept bank whose construction is detailed in Appendix B. Let

$$\mathcal{C}_{uni} = \mathcal{C}_{obj} \cup \mathcal{C}_{vid}, \quad \mathcal{C}_m = \{c_k^m\}_{k=1}^{K_m}, \quad m \in \{\text{obj}, \text{vid}\}, \tag{7}$$

where $\mathcal{C}_{obj}$ and $\mathcal{C}_{vid}$ denote object-level and motion-level semantic centroids, respectively. For each visual embedding $e_u \in \mathbb{R}^D$, we assign positive pseudo-labels using the nearest centroids in the corresponding semantic granularity, and sample negatives from the remaining centroids in the same set $\mathcal{C}_m$. The binary variable $y^m_{u,k} \in \{+1, -1\}$ indicates whether centroid c^m_k is a positive or negative semantic association for sample u. To capture both semantic structures, we optimize the following multi-label cluster discrimination objective, illustrated in Figure 3:

$$\mathcal{L} = \sum_{m \in \{\text{obj,vid}\}} \mathbb{E}_{(u,k)\sim\mathcal{C}_m} \log\big(1 + \exp(-y^m_{u,k}\, \sigma^m_{u,k})\big), \tag{8}$$

where $\sigma^m_{u,k} = e_u^\top c^m_k$ denotes the similarity score between the embedding e_u and its semantic centroid c^m_k, and u indexes visual embeddings while k indexes centroids in $\mathcal{C}_m$. For each granularity m, $(u, k) \sim \mathcal{C}_m$ samples visual embeddings e_u and semantic centroids c^m_k from the corresponding cluster set, with pseudo-labels $y^m_{u,k} \in \{+1, -1\}$ indicating positive or negative semantic associations. This unified formulation integrates both object- and motion-level clustering signals, enforcing spatiotemporal consistency and promoting discriminative representation learning.

2.4 Architecture

For the OneVision-Encoder, we adopt the Vision Transformer (ViT) architecture [29].

3D-RoPE for Codec Patchification. Unlike absolute encoding $p = (t, x, y)$, 3D-RoPE adopts a relative positional scheme [81] represented as $\Delta p = (t_1 - t_2, x_1 - x_2, y_1 - y_2)$, as illustrated in Figure 4. The relative offsets Δp for the three inputs are defined as:

- *Dense Video-Codec Patchification*: $\Delta p_{\text{codec}} = (t_i{-}t_j,\ x_i{-}x_j,\ y_i{-}y_j)$, which emphasizes inter-frame (I/P) residual alignment via the temporal offset $t_i{-}t_j$.
- *Chunk-wise Patchification*: for frames t_i, t_j belonging to chunks $c_i, c_j \in \{1, \ldots, C\}$, the relative offset is defined as $\Delta p_{\text{chunk}} = (\Delta c,\ \Delta x,\ \Delta y)$, capturing inter-chunk temporal disparity under non-uniform sampling, where $\Delta c = c_i - c_j$.
- *Single-Image Spatial Patchification*: for two patches within the same frame t with spatial coordinates (x_i, y_i) and (x_j, y_j), the relative positional offset is defined as $\Delta p_{\text{spatial}} = (0,\ \Delta x,\ \Delta y)$, where $\Delta x = x_i - x_j$ and $\Delta y = y_i - y_j$, encoding spatial positional relationships without temporal shifts.

Attentive Pooling Head. We employ a multi-head attention pooling module, adapted from SigLIP [103], to aggregate spatiotemporal tokens into compact class embeddings through learnable token-to-class attention weights, emphasizing salient regions and enabling unified global contextual representation across image and video modalities.

3 Experiments

We organize the empirical evaluation around two questions that directly test the central claim of the paper. First, does codec-guided sparse patchification transfer to multimodal reasoning when the vision encoder is embedded in an LMM? Second, does the same encoder learn stronger spatiotemporal representations when the language model is removed and the backbone is evaluated with a frozen attentive probe?

Pretraining summary. OV-Encoder is pretrained with a two-stage pipeline on image, OCR, and video data using a shared ViT backbone and the cluster-discrimination objective illustrated in Figure 3. Stage 1 uses image-only data at 224^2 resolution, while Stage 2 introduces OCR and video data, increases image resolution to 448^2, and trains video inputs with a mixture of codec-guided and frame-based patchification. For the default codec setting, a 64-frame clip is represented by two full I-frames and sparse P-frame patches selected from codec-derived motion and residual cues, yielding 2,048 visual tokens instead of the dense $64 \times 256 = 16{,}384$ tokens. Full data composition, annotation, optimizer settings, batch construction, and architecture details are reported in Appendix A and C.

3.1 LMM Probing Evaluation - Language Alignment

In this section, we evaluate how effectively OV-Encoder integrates into LMMs, focusing on whether its visual representations transfer to multimodal reasoning while isolating the impact of the vision encoder from language model capacity and training protocol. We lift these features through alignment

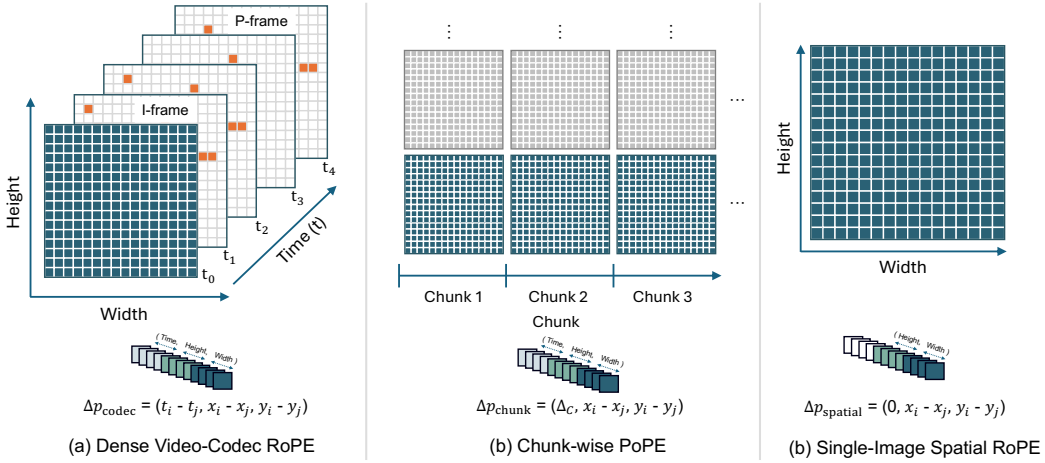


Figure 4: **3D-RoPE for Codec Patchification.** A unified relative positional encoding scheme is adopted for Codec Patchification. (a) encodes full spatiotemporal offsets $(\Delta t, \Delta x, \Delta y)$ over I/P-frame sequences to preserve motion-driven inter-frame structure. (b) defines temporal offsets at the chunk level, enabling structured reasoning under non-uniform temporal sampling. (c) degenerates the formulation to purely spatial offsets $(0, \Delta x, \Delta y)$ for static inputs. 3D-RoPE preserves structural consistency, enabling coherent attention over sparse and irregular token layouts.

tuning to construct a new codec encoder, **OV-Encoder-Lang**, specialized for MLLMs. The evaluation pipeline is shown in Appendix Figure 5.

All vision encoders are integrated into a Qwen3-4B-based LMM and evaluated with LMMs-Eval [105] on 16 image-, document-, and video-centric benchmarks. The image/document suite includes ChartQA [63], DocVQA [65], InfoVQA [64], AI2D [45], MMBench-EN [34], OCRBench [62], OCRBench v2 [62], MMStar [20], and RealWorldQA [24]; the video suite includes MVBench [54], MLVU-dev [109], NExT-QA [92], VideoMME [35], PerceptionTest [69], TOMATO [75], and LongVideoBench-Val-Video [91].

Protocol. We keep the language model architecture fixed and vary only the vision encoder. All models are trained with the same two-stage multimodal adaptation procedure: a Stage-1 alignment phase followed by Stage-2 instruction tuning on approximately 1.5M samples, consisting of 740K LLaVA-Next image-instruction samples and 800K LLaVA-Next-Video samples. We adopt native-resolution evaluation: input frames matching the encoder's native resolution are processed directly without additional spatial tiling or cropping. For codec-based video inference, each test video is sampled over 64 frames, codec-derived motion and residual signals select sparse patches across the sampled temporal extent, and the total visual-token budget is matched to the frame-based baseline. The decoupled comparison protocol against Qwen3-ViT and SigLIP2 is detailed in Appendix D.

Results. Table 1 shows that OV-Encoder provides consistent gains on video-centric benchmarks and remains competitive on image and document tasks. Compared with SigLIP2 under the same multimodal fine-tuning corpus and visual-token budget, OV-Encoder achieves stronger average transfer across the 16-benchmark suite. Compared with Qwen3-ViT, OV-Encoder obtains a 4.1-point average improvement on the video subset despite using substantially fewer vision-language pretraining tokens. These results indicate that the benefit of codec-guided sparse patchification is not limited to standalone video classification, but transfers to LMM-based reasoning under a controlled language-backbone and instruction-tuning protocol.

3.2 Attentive Probing Evaluation

We next isolate the intrinsic quality of the visual representation by freezing each backbone and training the same multi-head attentive pooling classifier on top of extracted spatiotemporal features. This protocol removes language-model effects and directly measures backbone-level video understanding. We evaluate SSV2, Diving48 [58], Perception Test [69], CharadesEgo [77], Epic-Kitchens-100 [44], Kinetics-400 [25], and HMDB51 [48].

Task	Benchmark	Qwen3-4B-Instruct2507				
		OV-Encoder-Lang (Codec)	**Qwen3-ViT (Frame)**	**OV-Encoder (Codec)**	**OV-Encoder-Frame (Frame)**	**SigLIP2 (Frame)**
Video	MVBench	**53.2**	47.4	**52.4**	49.8	47.2
	MLVU-dev	**47.4**	47.2	46.3	**49.4**	48.4
	NExT-QA (MC)	**76.1**	70.1	**75.6**	71.9	70.6
	VideoMME	**54.1**	47.2	**53.4**	49.3	46.8
	Perception Test	**60.6**	57.1	**60.3**	56.7	56.0
	TOMATO	21.8	**22.2**	22.2	21.8	**22.3**
	LongVideoBench-Val-Video	**51.6**	45.0	**50.4**	45.5	45.2
Image	AI2D	**80.2**	77.8	75.7	76.5	**78.6**
	ChartQA	**80.1**	79.6	76.5	**77.8**	76.4
	DocVQA	83.2	**85.1**	78.4	**79.5**	75.0
	InfoVQA	**51.6**	49.0	43.1	**45.5**	42.0
	MMBench-EN	**80.2**	79.4	77.2	78.5	**79.6**
	OCRBench	657	**706**	605	**630**	621
	OCRBench v2	**30.8**	30.6	**26.3**	26.1	26.1
	MMStar	**56.6**	56.6	52.1	54.3	**55.0**
	RealWorldQA	**66.1**	63.3	60.8	61.2	**62.1**

Table 1: **Comparison of different vision encoders on multimodal benchmarks.** All models are evaluated on a unified multimodal setting using Qwen3-4B-Instruct2507 as the language backbone. OV-Encoder-Lang denotes the language-aligned variant of the OV-Encoder architecture. Qwen3-ViT is extracted from Qwen3-VL-4B. SigLIP2 uses `siglip2-so400m-patch16-naflex`. **(Codec)** indicates codec-guided visual encoding using motion vectors and residual signals, while **(Frame)** indicates frame-based visual encoding with dense spatial patchification. Bold values indicate the best performance under the same evaluation setting. Results reported in the left columns correspond to encoders trained with caption supervision, whereas results in the right columns correspond to encoders trained without caption supervision.

Model Setup			Video Benchmarks (Acc. %)								
Method	**Backbone**	**Res.**	**Avg.**	SSV2	Diving48	Perception Test	CharadesEgo	Epic-Verb	Epic-Noun	Kinetics-400	HMDB51
8 Frames / 2048 Patches											
CLIP	ViT-L/14	224	50.5	48.2	46.6	52.2	10.8	52.8	36.1	79.3	78.0
SigLIP	ViT-L/16	256	50.1	50.7	43.9	48.9	10.9	52.2	39.1	78.2	77.0
MetaCLIP	ViT-L/14	224	48.5	50.6	28.9	49.8	10.4	54.1	37.1	79.6	77.1
MetaCLIP2	ViT-L/14	224	50.2	47.2	48.0	47.7	11.0	48.0	40.9	82.4	76.3
AIMv2	ViT-L/14	224	53.8	55.1	43.6	55.1	12.0	56.6	45.6	81.1	81.3
SigLIP2	ViT-L/16	256	53.1	52.6	50.1	52.7	11.6	54.2	43.8	80.9	79.1
DINOv3	ViT-L/14	224	58.0	57.4	58.6	59.3	**13.2**	**62.5**	51.7	82.9	78.6
OV-Encoder (Frame)	ViT-L/14	224	58.4	57.7	57.6	58.3	12.1	61.4	52.5	84.3	83.1
OV-Encoder (Codec)	ViT-L/14	224	**60.2**	**58.5**	**67.2**	**60.0**	12.3	62.3	**53.9**	**84.4**	**83.4**
16 Frames / 4096 Patches											
SigLIP	ViT-L/16	256	52.8	52.7	54.7	51.0	11.7	54.1	40.2	79.1	78.8
MetaCLIP2	ViT-L/14	224	51.0	49.3	42.1	51.1	11.2	49.2	43.2	84.0	78.2
AIMv2	ViT-L/14	224	56.4	57.2	55.7	56.4	12.4	58.3	46.2	82.2	82.6
SigLIP2	ViT-L/16	256	55.7	58.2	56.7	53.3	11.9	56.4	45.2	82.7	81.2
DINOv3	ViT-L/14	224	59.1	58.3	61.3	60.8	**14.0**	63.2	51.9	83.9	79.7
OV-Encoder (Frame)	ViT-L/14	224	59.9	58.7	63.2	60.3	12.6	62.9	**54.5**	85.1	81.6
OV-Encoder (Codec)	ViT-L/14	224	**61.5**	**60.1**	**69.4**	**60.9**	12.9	**63.3**	54.4	**85.4**	**85.3**

Table 2: **Comparison with state-of-the-art methods on video understanding benchmarks.** We report top-1 accuracy (%) using an attentive probe with frozen backbones, evaluated under two input configurations: **8 Frames / 2048 Patches** and **16 Frames / 4096 Patches**. For OV-Encoder (Codec), inputs are constructed using *Dense Video-codec Patchification*, which selectively encodes temporally salient patches from dense video inputs under the corresponding patch budgets. **Bold** indicates the best performance and underline indicates the second-best.

Protocol. All models use the same attentive pooling classifier, optimization schedule, and single-crop inference protocol. Frame-centric baselines use uniform temporal sampling with 8-frame and 16-frame inputs. OV-Encoder is evaluated in both frame-based and codec-guided modes under matched visual-token budgets: the codec-guided mode observes a dense 64-frame temporal span but retains

only the codec-selected patch tokens. Thus, improvements reflect better token allocation over time rather than a larger token budget.

Results. As shown in Table 2, OV-Encoder (Codec) improves the average attentive-probing accuracy over strong frame-centric baselines under both 2,048- and 4,096-token budgets. The gains are largest on motion-sensitive datasets, especially Diving48, where OV-Encoder (Codec) improves Top-1 accuracy by 17.1 points over SigLIP2 and 8.1 points over DINOv3 at the 2,048-patch budget. At the same time, performance remains strong on appearance-dominated datasets such as Kinetics-400 and HMDB51. These results support the central hypothesis that dense temporal coverage with sparse, codec-guided patch allocation provides a more effective use of visual tokens than sampling frames.

Patch-budget evidence. Table 3 reports the average over all eight attentive-probing benchmarks and highlights three motion-sensitive benchmarks; the full per-benchmark results are kept unchanged in Appendix E.2. Across patch budgets, OV-Encoder (Codec) consistently improves over the dense SigLIP2 baseline, with the strongest gains on Diving48, where dense temporal coverage is critical. The table and quantifies the 75.0%–96.9% patch reduction relative to dense 64-frame processing. Appendix E.3 performs controlled interventions on selected motion patches, and Appendix E.4 visualizes I/P-frame decomposition and token allocation patterns.

Budget	Method	*AVG	SSV2	Diving48	Perception Test
512	SigLIP2	43.4	42.8	28.1	38.7
	OV-Encoder (Codec)	**50.1**	50.0	46.5	50.5
1024	SigLIP2	50.8	49.0	48.7	50.1
	OV-Encoder (Codec)	**56.2**	56.6	54.9	58.6
2048	SigLIP2	53.1	52.6	50.1	52.7
	OV-Encoder (Codec)	**60.2**	58.5	67.2	60.0
4096	SigLIP2	55.7	58.2	56.7	53.3
	OV-Encoder (Codec)	**61.5**	60.1	69.4	60.9

Table 3: **Compact patch-budget scaling under attentive probing.** *AVG denotes the mean over all eight attentive-probing benchmarks;

4 Related Work

Advances in Visual Representation Learning. Vision Transformers [30, 53] have become a standard backbone for scalable visual representation learning. Recent self-supervised methods improve such backbones through complementary pretext structures: DINO-style self-distillation [43, 78], equivariant prediction under structured transformations [28, 68, 36, 39, 26], masked image modeling [41, 9, 31, 95], and latent predictive objectives such as JEPA and data2vec [5, 6, 7]. In parallel, CLIP-style language-image pretraining [14, 84, 57, 32, 96, 73] learns instance-level cross-modal alignment at large scale. Closest to our objective are cluster-discrimination methods, including UNICOM [2], MLCD [4], RICE [94], and UniViT [85], which use semantic grouping to move beyond purely pixel-level reconstruction or instance-wise contrast.

Efficient Video Representations. Efficient video modeling has primarily been studied through temporal sampling, token pruning, and token merging. Sampling-based methods improve temporal coverage by selecting frames uniformly or with motion-aware heuristics [98, 88], allocating or merging patches more selectively in long-video systems [18, 97, 46], or decomposing videos into keyframes and motion signals for generative multimodal pretraining [42]. However, these approaches typically still encode dense spatial patches within selected frames or impose an explicit frame-count limit. Token pruning reduces sequence length through random dropout [40, 61] or learned selection [71, 101, 19], while token merging aggregates redundant tokens via background-aware aggregation [59], semantic enhancement [102, 110], trainable merging [51], compressed auxiliary streams [47], or attention-based merging [12, 13]. In contrast, OV-Encoder does not learn an auxiliary pruning policy or merge tokens after dense patchification; instead, it directly exploits codec-derived motion and residual cues to allocate sparse patches over dense temporal coverage.

Video Codec Compression. Video compression exploits temporal predictability and structural redundancy [37, 1, 49]. Classical standards such as H.264/AVC [82, 107] and HEVC/H.265 [83, 82] operationalize this principle through motion compensation, transform coding, block partitioning, and entropy coding. More recent neural video compression methods [52, 66, 100, 56, 104, 87] learn spatiotemporal redundancy directly from data, while efficient video understanding models have begun to leverage compressed or redundancy-aware streams, e.g., Run-Length Tokenization [22] and motion-aware GOP encoding in EMA [108]. Rather than improving codec rate-distortion performance, we use codec decomposition itself as an inductive prior for representation learning, separating stable spatial context from sparse temporal variation before transformer encoding.

5 Conclusion

In this work, we presented OV-Encoder, a unified self-supervised vision encoder comprising Codec Patchification, 3D-RoPE, and cluster-discrimination supervision over image and video concept banks. We have shown that codec-guided patch selection can preserve dense temporal coverage while encoding only sparse, high-value visual evidence under fixed token budgets. We have demonstrated that OV-Encoder improves LMM probing over strong visual backbones under the same multimodal adaptation protocol and visual-token budget. We have further established through attentive probing that OV-Encoder (Codec) outperforms frame-centric encoders on motion-sensitive video benchmarks, with especially large gains on Diving48. Overall, OV-Encoder unlocks the potential to scale visual representation learning from dense frame processing to codec-aligned sparse spatiotemporal evidence, addressing the efficiency and temporal-coverage limits of conventional encoders.

References

[1] Nicola Adami, Alberto Signoroni, and Riccardo Leonardi. State-of-the-art and trends in scalable video compression with wavelet-based approaches. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(9):1238–1255, 2007.

[2] Xiang An, Jiankang Deng, Kaicheng Yang, Jaiwei Li, Ziyong Feng, Jia Guo, Jing Yang, and Tongliang Liu. Unicom: Universal and compact representation learning for image retrieval. In *ICLR*, 2023.

[3] Xiang An, Yin Xie, Kaicheng Yang, Wenkang Zhang, Xiuwei Zhao, Zheng Cheng, Yirui Wang, Songcen Xu, Changrui Chen, Didi Zhu, et al. Llava-onevision-1.5: Fully open framework for democratized multimodal training. *arXiv preprint arXiv:2509.23661*, 2025.

[4] Xiang An, Kaicheng Yang, Xiangzi Dai, Ziyong Feng, and Jiankang Deng. Multi-label cluster discrimination for visual representation learning. In *ECCV*, pages 428–444. Springer, 2024.

[5] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding. *CVPR*, 2023.

[6] Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv:2506.09985*, 2025.

[7] Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. Data2vec: A general framework for self-supervised learning in speech, vision and language. *arXiv:2202.03555*, 2022.

[8] 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.

[9] Hangbo Bao, Li Dong, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv:2106.08254*, 2021.

[10] Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video, 2024.

[11] Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mido Assran, and Nicolas Ballas. V-jepa: Latent video prediction for visual representation learning. *arXiv:2404.08471*, 2023.

[12] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token Merging: Your ViT But Faster. In *ICLR*. OpenReview.net, 2023.

[13] Daniel Bolya and Judy Hoffman. Token Merging for Fast Stable Diffusion. In *CVPR Workshops*, pages 4599–4603. IEEE, 2023.

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

[15] Minwoo Byeon, Beomhee Park, Haecheon Kim, Sungjun Lee, Woonhyuk Baek, and Saehoon Kim. Coyo-700m: Image-text pair dataset. `https://github.com/kakaobrain/coyo-dataset`, 2022.

[16] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *ECCV*, 2018.

[17] João Carreira, Dilara Gokay, Michael King, Chuhan Zhang, Ignacio Rocco, Aravindh Mahendran, Thomas Albert Keck, Joseph Heyward, Skanda Koppula, Etienne Pot, et al. Scaling 4d representations. *arXiv:2412.15212*, 2024.

[18] Wenhao Chai, Enxin Song, Yilun Du, Chenlin Meng, Vashisht Madhavan, Omer Bar-Tal, Jenq-Neng Hwang, Saining Xie, and Christopher D Manning. Auroracap: Efficient, performant video detailed captioning and a new benchmark. *arXiv:2410.03051*, 2024.

[19] Lei Chen, Zhan Tong, Yibing Song, Gangshan Wu, and Limin Wang. Efficient Video Action Detection with Token Dropout and Context Refinement. In *ICCV*, pages 10354–10365. IEEE, 2023.

[20] Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, et al. Are we on the right way for evaluating large vision-language models? *arXiv:2403.20330*, 2024.

[21] Tsai-Shien Chen et al. Panda-70m: Captioning 70m videos with multiple cross-modality teachers. *CVPR*, 2024.

[22] Rohan Choudhury, Guanglei Zhu, Sihan Liu, Koichiro Niinuma, Kris Kitani, and László Jeni. Don't look twice: Faster video transformers with run-length tokenization. *Advances in Neural Information Processing Systems*, 2024.

[23] Yung-Sung Chuang, Yang Li, Dong Wang, Ching-Feng Yeh, Kehan Lyu, Ramya Raghavendra, James Glass, Lifei Huang, Jason Weston, Luke Zettlemoyer, et al. Meta clip 2: A worldwide scaling recipe. *arXiv preprint arXiv:2507.22062*, 2025.

[24] X.AI Corp. Grok-1.5 vision preview: Connecting the digital and physical worlds with our first multimodal model. `https://x.ai/blog/grok-1.5v`, 2024.

[25] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling Egocentric Vision: Collection, Pipeline and Challenges for EPIC-KITCHENS-100. *Int. J. Comput. Vis.*, 130(1):33–55, 2022.

[26] Rumen Dangovski, Li Jing, Charlotte Loh, Seungwook Han, Akash Srivastava, Brian Cheung, Pulkit Agrawal, , and Marin Soljačić. Equivariant contrastive learning. *arXiv:2111.00899*, 2021.

[27] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[28] Alexandre Devillers and Mathieu Lefort. Equimod: An equivariance module to improve self-supervised learning. *arXiv:2211.01244*, 2023.

[29] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv:2010.11929*, 2020.

[30] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

[31] Alaaeldin El-Nouby, Michal Klein, Shuangfei Zhai, Miguel Angel Bautista, Alexander Toshev, Vaishaal Shankar, Joshua M Susskind, and Armand Joulin. Scalable pre-training of large autoregressive image models. *arXiv:2401.08541*, 2024.

[32] Alex Fang, Albin Madappally Jose, Amit Jain, Ludwig Schmidt, Alexander Toshev, and Vaishaal Shankar. Data filtering networks. In *ICLR*, 2024.

[33] Enrico Fini, Mustafa Shukor, Xiujun Li, Philipp Dufter, Michal Klein, David Haldimann, Sai Aitharaju, Victor G Turrisi da Costa, Louis Béthune, Zhe Gan, et al. Multimodal autoregressive pre-training of large vision encoders. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 9641–9654, 2025.

[34] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, et al. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv:2306.13394*, 2023.

[35] Chaoyou Fu, Yuhan Dai, Yondong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. *arXiv:2405.21075*, 2024.

[36] Quentin Garrido, Laurent Najman, and Yann Lecun. Self-supervised learning of split invariant equivariant. *Proceedings of Machine Learning Research*, pages 10975–10996, 2023.

[37] B. Girod, A.M. Aaron, S. Rane, and D. Rebollo-Monedero. Distributed video coding. *Proceedings of the IEEE*, 93(1):71–83, 2005.

[38] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzyńska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, Florian Hoppe, Christian Thurau, Ingo Bax, and Roland Memisevic. The "something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE international conference on computer vision*, pages 5842–5850, 2017.

[39] Sharut Gupta, Joshua Robinson, Derek Lim, Soledad Villar, and Stefanie Jegelka. Structuring representation geometry with rotationally equivariant contrastive learning. In *ICLR*, 2024.

[40] Tengda Han, Weidi Xie, and Andrew Zisserman. Turbo Training with Token Dropout. In *BMVC*, page 622. BMVA Press, 2022.

[41] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv:2111.06377*, 2021.

[42] Yang Jin, Zhicheng Sun, Kun Xu, Liwei Chen, Hao Jiang, Quzhe Huang, Chengru Song, Yuliang Liu, Di Zhang, Yang Song, et al. Video-lavit: Unified video-language pre-training with decoupled visual-motional tokenization. *arXiv preprint arXiv:2402.03161*, 2024.

[43] Cijo Jose, Théo Moutakanni, Dahyun Kang, Federico Baldassarre, Timothée Darcet, Hu Xu, Daniel Li, Marc Szafraniec, Michaël Ramamonjisoa, Maxime Oquab, Oriane Siméoni, Huy V. Vo, Patrick Labatut, and Piotr Bojanowski. DINOv2 meets text: A unified framework for image- and pixel-level vision-language alignment. In *CVPR*, 2025.

[44] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[45] Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *ECCV*, 2016.

[46] Wonkyun Kim, Changin Choi, Wonseok Lee, and Wonjong Rhee. An image grid can be worth a video: Zero-shot video question answering using a vlm. *IEEE Access*, 2024.

[47] Rajat Koner, Gagan Jain, Prateek Jain, Volker Tresp, and Sujoy Paul. Lookupvit: Compressing visual information to a limited number of tokens. In *ECCV (86)*, volume 15144 of *Lecture Notes in Computer Science*, pages 322–337. Springer, 2024.

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

[49] Dr T Senthil Kumar. A novel method for hdr video encoding, compression and quality evaluation. *Journal of Innovative Image Processing*, 1(2):71–80, 2019.

[50] Hugo Laurençon, Lucile Saulnier, Léo Tronchon, Stas Bekman, Amanpreet Singh, Anton Lozhkov, Thomas Wang, Siddharth Karamcheti, Alexander Rush, Douwe Kiela, et al. Obelics: An open web-scale filtered dataset of interleaved image-text documents. *NeurIPS*, 2023.

[51] Seon-Ho Lee, Jue Wang, Zhikang Zhang, David Fan, and Xinyu Li. Video token merging for long-form video understanding. *arXiv:2410.23782*, 2024.

[52] Jiahao Li, Bin Li, and Yan Lu. Deep contextual video compression. *NeurIPS*, 34:18114–18125, 2021.

[53] Kunchang Li, Yali Wang, Peng Gao, Guanglu Song, Yu Liu, Hongsheng Li, and Yu Qiao. Uniformer: Unified transformer for efficient spatiotemporal representation learning. *arXiv:2201.04676*, 2022.

[54] Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, Limin Wang, and Yu Qiao. Mvbench: A comprehensive multi-modal video understanding benchmark, 2023.

[55] Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Limin Wang, and Yu Qiao. Uniformerv2: Spatiotemporal learning by arming image vits with video uniformer. *arXiv:2211.09552*, 2022.

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

[57] 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:2306.15658*, 2023.

[58] Yingwei Li, Yi Li, and Nuno Vasconcelos. Resound: Towards action recognition without representation bias. In *Proceedings of the European conference on computer vision (ECCV)*, pages 513–528, 2018.

[59] Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. Not All Patches are What You Need: Expediting Vision Transformers via Token Reorganizations. *CoRR*, abs/2202.07800, 2022.

[60] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, January 2024.

[61] Yue Liu, Christos Matsoukas, Fredrik Strand, Hossein Azizpour, and Kevin Smith. PatchDropout: Economizing Vision Transformers Using Patch Dropout. In *WACV*, pages 3942–3951. IEEE, 2023.

[62] Yuliang Liu, Zhang Li, Mingxin Huang, Biao Yang, Wenwen Yu, Chunyuan Li, Xu-Cheng Yin, Cheng-Lin Liu, Lianwen Jin, and Xiang Bai. Ocrbench: on the hidden mystery of ocr in large multimodal models. *Science China Information Sciences*, 2024.

[63] Ahmed Masry, Xuan Long Do, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. In *ACL*, 2022.

[64] Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and CV Jawahar. Infographicvqa. In *WACV*, 2022.

[65] Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. Docvqa: A dataset for vqa on document images. In *WACV*, 2021.

[66] Fabian Mentzer, George Toderici, David Minnen, Sung-Jin Hwang, Sergi Caelles, Mario Lucic, and Eirikur Agustsson. Vct: A video compression transformer. *arXiv:2206.07307*, 2022.

[67] Antoine Miech et al. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. *ICCV*, 2019.

[68] Jung Yeon Park, Ondrej Biza, Linfeng Zhao, Jan Willem van de Meent, and Robin Walters. Learning symmetric embeddings for equivariant world models. *arXiv:2204.11371*, 2022.

[69] Viorica Patraucean, Lucas Smaira, Ankush Gupta, Adria Recasens, Larisa Markeeva, Dylan Banarse, Skanda Koppula, joseph heyward, Mateusz Malinowski, Yi Yang, Carl Doersch, Tatiana Matejovicova, Yury Sulsky, Antoine Miech, Alexandre Fréchette, Hanna Klimczak, Raphael Koster, Junlin Zhang, Stephanie Winkler, Yusuf Aytar, Simon Osindero, Dima Damen, Andrew Zisserman, and Joao Carreira. Perception test: A diagnostic benchmark for multimodal video models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *NeurIPS*, volume 36, pages 42748–42761. Curran Associates, Inc., 2023.

[70] 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 *ICML*, 2021.

[71] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *NeurIPS*, 34:13937–13949, 2021.

[72] Oshmita Sarkar, Satyam Sinha, Ajay Kumar Jena, Ajaya Kumar Parida, Nirupama Parida, and Raj Kumar Parida. Automatic number plate character recognition using paddle-ocr. In *2024 International Conference on Innovations and Challenges in Emerging Technologies (ICICET)*, pages 1–7. IEEE, 2024.

[73] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade W Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa R Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5b: An open large-scale dataset for training next generation image-text models. In *NeurIPS Datasets and Benchmarks*, 2022.

[74] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv:2111.02114*, 2021.

[75] Kazutoshi Shinoda, Nobukatsu Hojo, Kyosuke Nishida, Saki Mizuno, Keita Suzuki, Ryo Masumura, Hiroaki Sugiyama, and Kuniko Saito. Tomato: Verbalizing the mental states of role-playing llms for benchmarking theory of mind. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 1520–1528, 2025.

[76] Yan Shu, Zheng Liu, Peitian Zhang, Minghao Qin, Junjie Zhou, Zhengyang Liang, Tiejun Huang, and Bo Zhao. Video-xl: Extra-long vision language model for hour-scale video understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 26160–26169, 2025.

[77] Gunnar A Sigurdsson, Abhinav Gupta, Cordelia Schmid, Ali Farhadi, and Karteek Alahari. Charades-ego: A large-scale dataset of paired third and first person videos. *arXiv preprint arXiv:1804.09626*, 2018.

[78] Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3, 2025.

[79] Mattia Soldan, Fabian Caba Heilbron, Bernard Ghanem, Josef Sivic, and Bryan Russell. Residualvit for efficient temporally dense video encoding. In *ICCV*, pages 22305–22315, 2025.

[80] Enxin Song, Wenhao Chai, Shusheng Yang, Ethan Armand, Xiaojun Shan, Haiyang Xu, Jianwen Xie, and Zhuowen Tu. Videonsa: Native sparse attention scales video understanding. *arXiv:2510.02295*, 2025.

[81] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

[82] Gary J Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand. Overview of the high efficiency video coding (hevc) standard. *IEEE Transactions on circuits and systems for video technology*, 22(12):1649–1668, 2012.

[83] Gary J Sullivan and Thomas Wiegand. Video compression-from concepts to the h. 264/avc standard. *Proceedings of the IEEE*, 93(1):18–31, 2005.

[84] Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. EVA-CLIP: Improved training techniques for clip at scale. *arXiv:2303.15389*, 2023.

[85] Feilong Tang, Xiang An, Haolin Yang, Yin Xie, Kaicheng Yang, Ming Hu, Zheng Cheng, Xingyu Zhou, Zimin Ran, Imran Razzak, Ziyong Feng, Behzad Bozorgtabar, Jiankang Deng, and Zongyuan Ge. Univit: Unifying image and video understanding in one vision encoder. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[86] 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:2502.14786*, 2025.

[87] Chenting Wang, Kunchang Li, Tianxiang Jiang, Xiangyu Zeng, Yi Wang, and Limin Wang. Make your training flexible: Towards deployment-efficient video models. *arXiv:2503.14237*, 2025.

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

[89] 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. InternVideo2: Scaling foundation models for multimodal video understanding. In *ECCV*, 2024.

[90] Yuetian Weng, Mingfei Han, Haoyu He, Xiaojun Chang, and Bohan Zhuang. LongVLM: Efficient Long Video Understanding via Large Language Models. In *ECCV (33)*, volume 15091 of *Lecture Notes in Computer Science*, pages 453–470. Springer, 2024.

[91] Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. Longvideobench: A benchmark for long-context interleaved video-language understanding, 2024.

[92] Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. Next-qa: Next phase of question-answering to explaining temporal actions. In *CVPR*, pages 9777–9786, June 2021.

[93] Chunyu Xie, Heng Cai, Jincheng Li, Fanjing Kong, Xiaoyu Wu, Jianfei Song, Henrique Morimitsu, Lin Yao, Dexin Wang, Xiangzheng Zhang, et al. Ccmb: A large-scale chinese cross-modal benchmark. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 4219–4227, 2023.

[94] Yin Xie, Kaicheng Yang, Xiang An, Kun Wu, Yongle Zhao, Weimo Deng, Zimin Ran, Yumeng Wang, Ziyong Feng, Roy Miles, et al. Region-based cluster discrimination for visual representation learning. In *ICCV*, pages 1793–1803, 2025.

[95] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling, 2022.

[96] 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 *ICLR*, 2024.

[97] Weili Xu, Enxin Song, Wenhao Chai, Xuexiang Wen, Tian Ye, and Gaoang Wang. Auroralong: Bringing rnns back to efficient open-ended video understanding. *arXiv:2507.02591*, 2025.

[98] Hongwei Xue, Tiankai Hang, Yanhong Zeng, Yuchong Sun, Bei Liu, Huan Yang, Jianlong Fu, and Baining Guo. Advancing high-resolution video-language representation with large-scale video transcriptions. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[99] Haolin Yang, Feilong Tang, Lingxiao Zhao, Xiang An, Ming Hu, Huifa Li, Xinlin Zhuang, Yifan Lu, Xiaofeng Zhang, Abdalla Swikir, et al. Streamagent: Towards anticipatory agents for streaming video understanding. *arXiv:2508.01875*, 2025.

[100] Ren Yang, Fabian Mentzer, Luc Van Gool, and Radu Timofte. Learning for video compression with recurrent auto-encoder and recurrent probability model. *IEEE Journal of Selected Topics in Signal Processing*, 15(2):388–401, 2021.

[101] Hongxu Yin, Arash Vahdat, José M. Álvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. A-ViT: Adaptive Tokens for Efficient Vision Transformer. In *CVPR*, pages 10799–10808. IEEE, 2022.

[102] Wang Zeng, Sheng Jin, Wentao Liu, Chen Qian, Ping Luo, Wanli Ouyang, and Xiaogang Wang. Not All Tokens Are Equal: Human-centric Visual Analysis via Token Clustering Transformer. In *CVPR*, pages 11091–11101. IEEE, 2022.

[103] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *ICCV*, 2023.

[104] Guozhen Zhang, Yuhan Zhu, Haonan Wang, Youxin Chen, Gangshan Wu, and Limin Wang. Extracting motion and appearance via inter-frame attention for efficient video frame interpolation. In *CVPR*, pages 5682–5692, 2023.

[105] Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, and Ziwei Liu. Lmms-eval: Reality check on the evaluation of large multimodal models. In *ACL*, 2025.

[106] Yuanhan Zhang, Bo Li, haotian Liu, Yong jae Lee, Liangke Gui, Di Fu, Jiashi Feng, Ziwei Liu, and Chunyuan Li. Llava-next: A strong zero-shot video understanding model, April 2024.

[107] Zhuo Zhao and Ping Liang. A highly efficient parallel algorithm for h. 264 video encoder. In *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, volume 5, pages V–V. IEEE, 2006.

[108] 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 Computer Vision and Pattern Recognition Conference*, pages 24159–24168, 2025.

[109] Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Shitao Xiao, Xi Yang, Yongping Xiong, Bo Zhang, Tiejun Huang, and Zheng Liu. Mlvu: A comprehensive benchmark for multi-task long video understanding. *arXiv:2406.04264*, 2024.

[110] Xingyi Zhou, Anurag Arnab, Chen Sun, and Cordelia Schmid. How can objects help action recognition? In *CVPR*, pages 2353–2362. IEEE, 2023.

Table 4: **OneVision-Encoder Pretraining Dataset.** The pretraining corpus combines large-scale image and video datasets for unified visual representation learning. Image datasets primarily provide broad visual coverage, while video datasets support temporal modeling and video–language alignment. We use "ExoVideo" to denote large-scale third-person web videos, and "ActionVideo" to denote curated action recognition datasets.

Source	Samples	Type	Modality	Temporal	Curation
LAION-400M [74]	250M	WebImages	Image	–	Yes
COYO-700M [15]	400M	WebImages	Image	–	Yes
OBELICS [50]	15M	Documents	Image	–	Yes
Zero250M [93]	15M	CuratedImages	Image	–	Yes
ImageNet-21K [27]	14M	Images	Image	–	Yes
HowTo100M [67]	50M	ExoVideo	Video	Short	No
Panda-70M [21]	50M	ExoVideo	Video	Long	Yes
Kinetics-710 [55]	658K	ActionVideo	Video	Short	Yes
SSV2 [38]	221K	ActionVideo	Video	Short	Yes

A Pretraining Dataset

Data Annotation and Processing. This section details the annotation and processing pipeline for the OneVision-Encoder pretraining data. Our core objective is to generate high-quality supervision signals for massive-scale data through automated means.

Image Data Annotation. For image data, we primarily process LAION-400M [73] and COYO-700M [15]. First, we employ a Union-Find algorithm to strictly deduplicate the dataset. Subsequently, we utilize the metaclip-h14-fullcc2.5b [96] model to extract image features and cluster all images into two million classes. Based on this clustering, each image sample is annotated with the nearest Top-10 class centers as its multi-label supervision signal. Furthermore, we incorporate the OBELICS [50] and Zero250M [93] datasets. We utilize PaddleOCR [72] to recognize text within images and perform word segmentation on the recognized content; the resulting vocabulary is used as multi-labels to construct a supervision signal containing exactly 100 fine-grained tags per image.

Video Data Annotation. The video data construction encompasses HowTo100M [67], Panda-70M [21], Kinetics-710 [55], and Something-Something-V2 (SSV2) [38]. We uniformly adopt metaclip-h14-fullcc2.5b as the video encoder, performing uniform frame sampling to extract features from a fixed 8-frame clip. In the feature processing stage, we adopt a "L2 Normalize $\rightarrow$ Concatenate $\rightarrow$ L2 Normalize" strategy to generate video-level representations. We then cluster the video representations into 400k classes and assign each video clip the nearest Top-10 class centers as its multi-labels.

B Image and Video Clustering

Iterative clustering-discrimination approaches commonly suffer from substantial computational overhead [16]. To address this issue, we adopt a single-step offline clustering to efficiently capture both object-level semantics from images and motion-level semantics from videos. Notably, embeddings used for clustering are extracted using a separate frozen vision encoder (e.g., metaclip-h14 [96]).

For the image modality, we follow the clustering formulation in MLCD [4] to capture object-level semantics. Given an image embedding $e_i^{\mathrm{obj}} \in \mathbb{R}^D$, we learn a set of object-level semantic centroids $\mathcal{C}_{\mathrm{obj}} = \{c_k^{\mathrm{obj}}\}_{k=1}^{K_{\mathrm{obj}}} \subseteq \mathbb{R}^D$ by minimizing the within-cluster distance:

$$\mathcal{C}_{\mathrm{obj}} \;=\; \arg\min_{\{c_k^{\mathrm{obj}}\}} \sum_{i=1}^{N_{\mathrm{obj}}} \min_{1\le k\le K_{\mathrm{obj}}} \left\| e_i^{\mathrm{obj}} - c_k^{\mathrm{obj}} \right\|_2^2, \tag{9}$$

where N_{obj} denotes the number of image samples.

For the video modality, we extend this formulation to model motion-level dynamics. Video embeddings are derived from fixed-length 16-frame inputs, where frame-level features are concatenated to form a single video-level representation $e_i^{\text{vid}} \in \mathbb{R}^D$. We then define a set of video semantic centroids $\mathcal{C}_{\text{vid}} = \{c_k^{\text{vid}}\}_{k=1}^{K_{\text{vid}}} \subseteq \mathbb{R}^D$, and formulate the clustering objective for videos as

$$\mathcal{C}_{\text{vid}} \;=\; \arg\min_{\{c_k^{\text{vid}}\}} \sum_{i=1}^{N_{\text{vid}}} \min_{1\le k\le K_{\text{vid}}} \left\| e_i^{\text{vid}} - c_k^{\text{vid}} \right\|_2^2, \tag{10}$$

where N_{vid} denotes the number of video samples. We define a set of shared semantic centroids $\mathcal{C}_{uni} = \{c_k^{\text{obj}}\}_{k=1}^{K_{\text{obj}}} \cup \{c_k^{\text{vid}}\}_{k=1}^{K_{\text{vid}}}$, and $K = K_{\text{obj}} + K_{\text{vid}}$ represents the total number of clusters across both image and video modalities. These semantic centroids are later used as supervision signals in the cluster discrimination objective, where image embeddings are supervised by object-level centroids and video embeddings are supervised by motion-level centroids.

C Implementation Details

Model Architecture and Configuration. OneVision-Encoder Large is implemented as a Vision Transformer with 24 transformer layers, a hidden dimension of 1024, and 16 attention heads. The model uses a patch size of 14×14 and adopts GELU activations with Layer Normalization throughout the network. Attention computation is accelerated using Flash Attention 2, enabling efficient training and inference at scale. A summary of the core architectural hyperparameters is provided in Table 5.

Table 5: Architecture configuration of OneVision-Encoder Large.

Component	Setting
Transformer layers	24
Hidden dimension	1024
Attention heads	16
Patch size	14×14
MLP expansion ratio	$4\times$
Position encoding	3D RoPE (T:H:W = 4:6:6)

Unified Patch-based Input Representation. All inputs are converted into patch tokens and processed by a single ViT backbone. Images are treated as single-frame videos (T=1), while videos are represented in a 5D tensor format. To ensure consistent temporal reasoning, all video inputs are mapped to a virtual temporal grid of 64 frames, regardless of the actual number of frames processed. This mapping enables uniform temporal position encoding across dense, sparse, and Codec inputs.

For inputs that do not cover all 64 frames explicitly, a *visible indices* mechanism is used to associate each selected patch with its corresponding temporal position in the virtual grid. This design decouples temporal coverage from token density and allows sparse inputs to preserve long-range temporal structure.

Codec-style Patch Selection. Codec-style processing operates on dense videos of 64 frames at a spatial resolution of 224×224. Motion vectors and prediction residuals are extracted from the HEVC codec and used to estimate patch-level temporal saliency. Motion vectors capture object displacement at sub-pixel precision, while residuals encode fine-grained appearance changes not explained by motion compensation. These signals are fused into a unified saliency score for each patch across all frames.

Patches are ranked globally by their saliency scores, and only the top-K patches are retained. This selection typically preserves between 3.1% and 25% of all patches, corresponding to a compression ratio of 75%–96.9% relative to dense processing. Selected patches are then reassembled into a compact video representation and passed to the ViT using sparse visible indices. Table 6 summarizes the codec-style selection procedure.

Video Processing Modes and Batch Composition. During pretraining, OneVision-Encoder employs a mixed-modality batch that includes multiple video processing modes. This design exposes the model to diverse temporal structures and encourages robust representation learning. Video samples

Table 6: Codec-style patch selection pipeline.

Step	Description
Motion extraction	Decode HEVC motion vectors with camera motion compensation
Residual extraction	Obtain prediction residuals for fine-grained changes
Energy fusion	Combine motion and residual energies into a saliency score
Top-K selection	Retain globally most salient patches across all frames
Sparse encoding	Process selected patches with sparse visible indices

Table 7: Video processing modes used during training.

Mode	Batch Ratio	Input Form	Output Shape
Codec	50%	Dense video + saliency	$[B, 3, 8, 224, 224]$
Frame sampling	37.5%	Uniform temporal bins	$[B, 3, 8, 224, 224]$
Tiling	12.5%	Vertical frame concatenation	$[B, 3, 1792, 224]$

within a batch are split into three processing modes: Codec patchification, uniform frame sampling, and Tiling-style spatial concatenation. All modes produce inputs that are compatible with the same ViT backbone and position encoding.

Position Encoding Consistency. All input modes share the same 3D Rotary Position Embedding. The temporal dimension of the RoPE encodes the position of each patch within the 64-frame virtual grid, while spatial dimensions encode patch row and column indices. For Codec inputs, patches may originate from arbitrary frames but are positioned correctly via their temporal indices. For uniformly sampled frames, temporal gaps are explicitly encoded. Tiling inputs are treated as single-frame inputs with fixed temporal positions. This unified encoding scheme enables the model to reason coherently over heterogeneous spatiotemporal layouts.

Training and Inference Behavior. The model is trained using a unified optimization pipeline across all modalities. No modality-specific parameters or task-specific encoders are introduced. At inference time, the same preprocessing logic is applied, and the model can flexibly switch between Codec sparse processing and conventional frame sampling depending on computational constraints. This design allows efficient deployment across a wide range of image and video understanding tasks without architectural modification.

D Controlled Evaluation Pipeline

As shown in Figure 5, we adopt a controlled evaluation protocol to compare OneVision-Encoder with Qwen3-ViT and SigLIP2 under LMM probing. For comparison with Qwen3-ViT, OneVision-Encoder is first integrated with the Qwen3-1.7B language model and trained through Stage 1 and Stage 1.5 under the LLaVA-OneVision-1.5 framework to adapt the encoder to native-resolution multimodal inputs. After alignment, the trained vision encoder is decoupled and evaluated under the same LLaVA-Next-Videos instruction-tuning setting as Qwen3-ViT, ensuring a fair comparison under identical downstream supervision. For comparison with SigLIP2, all models are directly evaluated under identical multimodal fine-tuning conditions using a unified 1.5M-scale instruction-tuning corpus, while keeping the language model backbone fixed. This decoupled and unified evaluation pipeline isolates the contribution of visual representation learning and avoids confounding effects from language model capacity, instruction data leakage, or differing alignment procedures.

E Additional Experimental Results

E.1 Stage-wise Multimodal Training Analysis

We analyze how the second pretraining stage contributes to multimodal transfer. OV-Encoder-Stage1 is trained with image-only data and therefore emphasizes general visual representation learning. OV-Encoder-Stage2 continues from Stage1 and introduces OCR and video data, together with codec-style

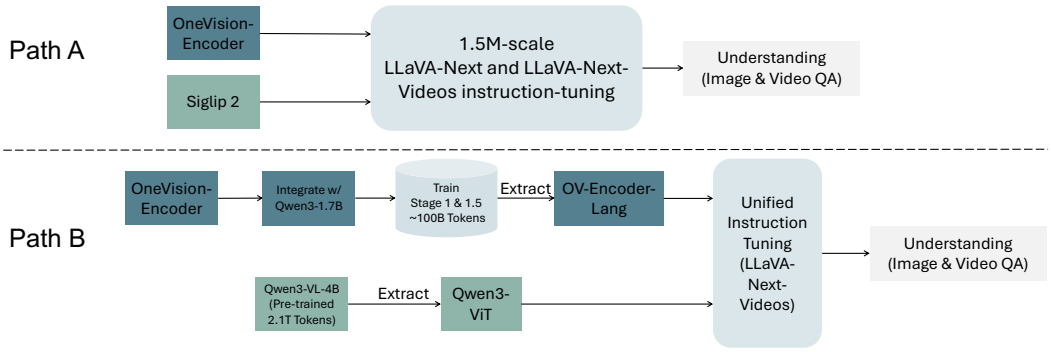


Figure 5: Controlled evaluation pipeline decoupling the encoder for fair comparison against Qwen3-ViT and SigLIP2

Benchmark	OV-Encoder-stage1	OV-Encoder-stage2	SigLIP2-sig
AI2D	73.6	74.5	**77.5**
ChartQA	73.6	76.2	**77.0**
DocVQA	74.3	**78.5**	74.4
InfoVQA	34.7	**41.4**	37.7
MMBench-EN	74.7	76.3	**78.1**
MMStar	49.3	49.7	**52.6**
RealWorldQA	61.8	61.3	**62.2**
OCRBench	551.0	**601.0**	590.0

Table 8: **Comparison of OV-Encoder training stages on image understanding benchmarks.** In this study, we evaluate two training variants of the same ViT architecture under an identical data scale: OV-Encoder-stage1 is trained with image-only data, while OV-Encoder-stage2 continues training from Stage 1 and incorporates OCR and video data with codec-style patch selection. **Bold** values indicate the best performance among the compared encoders.

patch selection and chunk-wise temporal sampling. As shown in Table E.1, Stage2 consistently improves over Stage1 on OCR- and document-sensitive benchmarks such as DocVQA, InfoVQA, and OCRBench, while preserving competitive performance on general visual reasoning tasks. This supports the use of multimodal continuation training as a bridge between generic visual representation learning and LMM-ready visual encoding.

E.2 Patch-Efficient Video Understanding

We further compare dense full-frame patch processing with OV-Encoder (Codec) under fixed token budgets. All results are computed from the same 64-frame source video, which contains $64 \times 256 = 16{,}384$ dense patches at the evaluated resolution. SigLIP2 follows the conventional frame-centric strategy, where each group of 256 patches corresponds to a full RGB frame. In contrast, OV-Encoder (Codec) preserves the 64-frame temporal span and reallocates the same patch budget across time using codec-native motion vectors and residuals. Table 9 shows that this allocation improves performance across patch budgets from 512 to 4,096 tokens, corresponding to retaining only 3.1%–25.0% of the dense patches and reducing patch processing by 75.0%–96.9%.

E.3 Ablation of Codec-guided Patch Selection

Performance gains alone do not establish that codec guidance is the functional mechanism. We therefore conduct controlled interventions that manipulate patch content while holding token count, spatiotemporal positions, positional encodings, and model parameters fixed.

Patch content necessity. We first replace 50% of codec-selected motion-heavy patches with non-motion patches sampled from the same video while preserving their original positions. The degrada-

Table 9: **Effect of patch budget scaling under attentive probing.** Patch budgets of 512/1024/2048/4096 correspond to 2/4/8/16 video frames, respectively. Dense SigLIP2 processes all spatial patches per frame, while OV-Encoder (Codec) selectively retains motion-relevant patches guided by codec-derived temporal signals.

Setting	Video Benchmarks (Acc. %)								
	AVG	SSV2	Diving48	Perception Test	CharadesEgo	Epic-Verb	Epic-Noun	Kinetics-400	HMDB51
Patch Budget = 512									
SigLIP2	43.4	42.8	28.1	38.7	10.1	42.9	37.4	74.5	72.4
OV-Encoder (Codec)	**50.1**	50.0	46.5	50.5	10.6	50.3	41.7	76.7	74.7
Patch Budget = 1024									
SigLIP2	50.8	49.0	48.7	50.1	10.8	50.6	42.0	78.8	76.6
OV-Encoder (Codec)	**56.2**	56.6	54.9	58.6	11.1	58.2	48.4	81.8	80.3
Patch Budget = 2048									
SigLIP2	53.1	52.6	50.1	52.7	11.6	54.2	43.8	80.9	79.1
OV-Encoder (Codec)	**60.2**	58.5	67.2	60.0	12.3	62.3	53.9	84.4	83.4
Patch Budget = 4096									
SigLIP2	55.7	58.2	56.7	53.3	11.9	56.4	45.2	82.7	81.2
OV-Encoder (Codec)	**61.5**	60.1	69.4	60.9	12.9	63.3	54.4	85.4	85.3

tion in Table 10 indicates that the selected patch content is necessary: the gains are not explained by sparsity or positional bias alone.

Semantic specificity of motion cues. We next replace codec-selected motion patches with motion patches drawn from unrelated videos. This preserves motion energy but breaks semantic alignment. The resulting drop is larger on fine-grained temporal benchmarks, showing that the model benefits from semantically correct motion rather than generic motion magnitude.

Negative control: patch-position shuffle. Finally, we preserve selected visual content but randomly permute the spatiotemporal positions. This causes the strongest degradation, confirming that content and position must remain jointly coherent. Overall, the ordering of interventions rules out explanations based solely on regularization, noise injection, or center-biased token sparsity.

E.4 Qualitative Analysis

We visualize the patch selection mechanism guided by residual magnitudes and motion vectors. Figure 6 compares an I-frame with subsequent P-frame residuals. I-frames retain complete spatial structure, while P-frame residual maps emphasize regions with significant motion-compensated appearance changes. The selected patches concentrate on dynamic foreground regions and are suppressed in static background areas, matching the intended role of codec-derived saliency as a proxy for new visual evidence.

Figure 7 contrasts conventional frame-centric video processing with the proposed Codec patch extraction. Dense 64-frame processing preserves full temporal context at high computational cost, while uniform frame sampling reduces computation by discarding intermediate frames. Codec patch extraction instead analyzes all frames, identifies temporally salient regions, and retains only selected patches under a fixed token budget. This decouples temporal coverage from token density and explains why OV-Encoder (Codec) is particularly effective on motion-sensitive tasks.

Table 10: **Controlled interventions on codec-selected motion patches.** All settings use identical token budgets and visual content unless otherwise specified.

Setting	Video Benchmarks (Acc. %)								
	AVG	SSV2	Diving48	Perception Test	CharadesEgo	Epic-Verb	Epic-Noun	Kinetics-400	HMDB51
OV-Encoder (Codec)	**61.5**	**60.1**	**69.4**	**60.9**	**12.9**	**63.3**	**54.4**	**85.4**	**85.3**
Non-motion Patch Replacement (50%) (Same Video, Same Position)	55.4	52.1	55.4	54.9	11.6	56.3	50.2	83.1	79.4
Counterfactual Motion Replacement (50%) (Motion Patch from Other Video)	54.9	50.6	57.2	53.8	11.3	55.1	49.6	82.7	79.0
Patch–Position Shuffle (Content Preserved, Positions Shuffled)	48.1	41.8	46.3	45.1	8.7	48.2	42.5	78.4	73.6

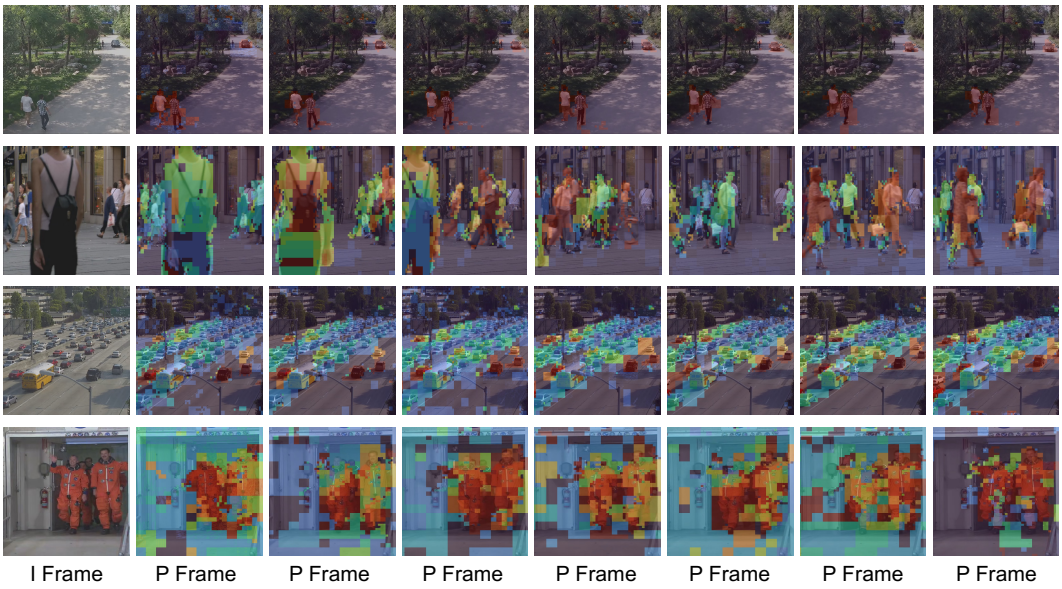


Figure 6: Visualization of I- and P-frame decomposition in HEVC. I-frames retain complete spatial structure, whereas P-frames encode motion-compensated residuals highlighting motion. Bright areas denote high residual magnitudes, while dark areas indicate static content.

F Spatial Bias Analysis

As shown in Figure 8, the reported statistics are computed from a random subset of 200,000 video samples drawn from the training data. When using codec-guided patch selection alone, the selected visual tokens exhibit a pronounced spatial center bias, with the majority of tokens concentrated in the central regions of the frame. This behavior reflects intrinsic statistical properties of video data: due to camera framing and subject placement, salient motion cues and residual signals are typically denser near the image center. While such a bias enables the model to focus on regions with strong motion signals, it also leads to insufficient coverage of peripheral areas, thereby weakening the representation of global scene structure and fine-grained action cues.

With the introduction of chunk-wise patchification, the spatial distribution of selected tokens becomes markedly more uniform, redistributing tokens toward the image periphery and boundary regions and effectively mitigating the center bias induced by codec-driven selection. This effect arises because

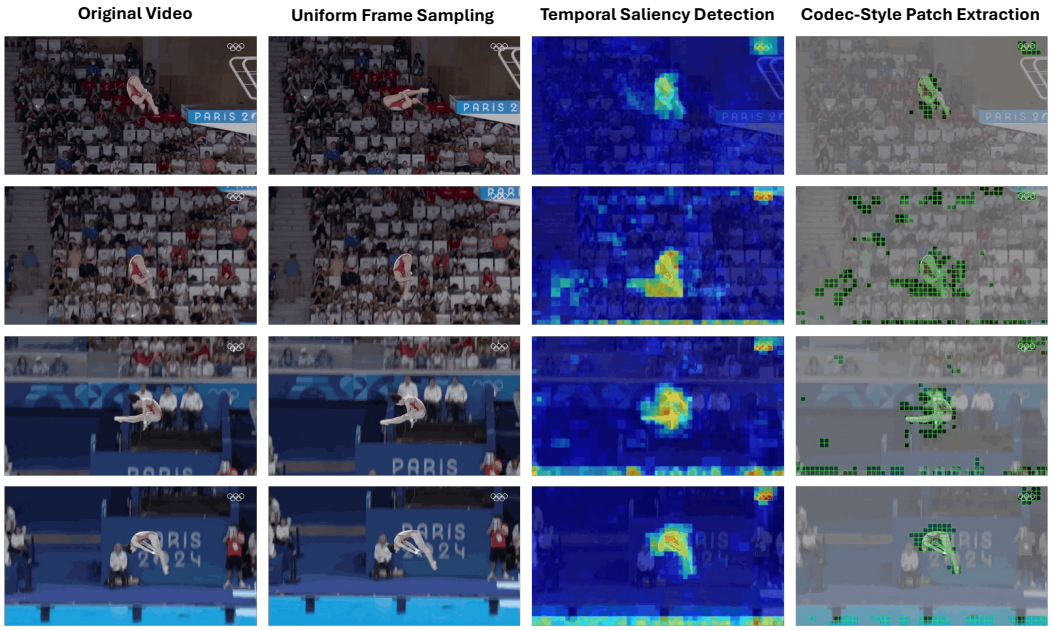


Figure 7: Comparison of video processing pipelines for spatiotemporal representation learning. (a) original dense video input with full temporal context, (b) uniform frame sampling that sparsely selects evenly spaced frames, (c) temporal saliency detection that identifies motion- and event-centric regions across all frames, and (d) Codec patch extraction that selectively retains temporally salient patches under a fixed token budget.

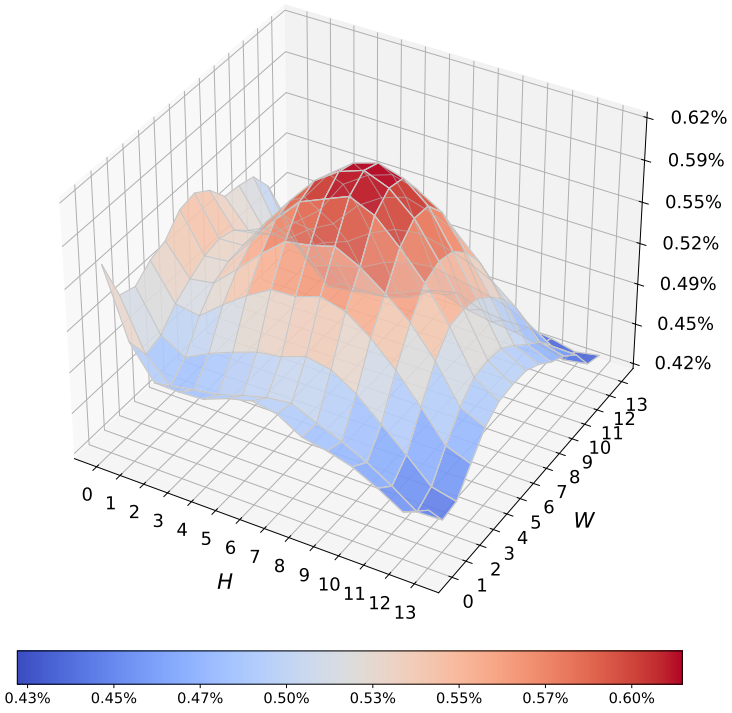


Figure 8: Spatial Bias Analysis of Selected Visual Patches under a given setting.

chunk-wise sampling partitions the video along the temporal dimension and performs global selection over visible patch indices, resulting in a more structurally balanced spatial coverage. Importantly, this rebalancing is achieved without increasing the token budget, but through a principled reallocation of patch selection that enhances spatial diversity and complements motion-centric evidence with global contextual information.

G Token Allocation Case Study

For each case, we plot time (x-axis) versus accumulated visual tokens (y-axis) under a fixed token budget. Stepwise growth indicates when a method allocates a non-trivial number of tokens to a timestamp. Vertical rugs at the bottom show the actual timestamps observed by each strategy, and triangles denote I-frame anchors in the codec-style layout. Case 1 uses a 2048-token budget (uniform 8-frame baseline), and Case 2 uses a 4096-token budget (uniform 16-frame baseline). These examples complement the quantitative results by illustrating two regimes in which codec-guided patch allocation improves temporal evidence coverage.

G.1 Case 1

We consider videos whose semantics are carried by smooth, persistent motion, where essentially every moment contributes evidence. In this regime, missing intermediate timestamps breaks trajectory continuity and harms recognition. Figure 9 visualizes this behavior on a Diving example. Under the same fixed token budget of 2048 tokens, uniform *8-frame* sampling concentrates all tokens on only eight sparse timestamps (leaving large temporal gaps), whereas codec-style allocation first samples a 64-frame timeline and then reallocates tokens through saliency-guided P-frame patch selection, preserving dense motion evidence across time. This is particularly important for high-angular-velocity actions such as diving, where discriminative cues are temporally dense and distributed across many brief pose transitions rather than confined to a single transition.

G.2 Case 2

We consider videos whose semantics concentrate in short, discrete transitions, where the true "key frames" are sparse events such as brief pours, quick tool interactions, or sudden scene changes. In this regime, the primary risk of uniform sampling is *misplacement*: even with a sufficient token budget, the 16 sampled frames can easily miss a short segment entirely, causing decisive evidence to be absent from the visual input. Figure 10 illustrates this effect on a 130-second cooking video. Notably, several pour segments are extremely short (e.g., **P4**: 38.0–39.0s), making them easy to skip under uniform sampling. In contrast, codec-style allocation first samples a 64-frame timeline and then reallocates the same token budget toward high-saliency moments via P-frame patch selection, increasing the probability of capturing the decisive evidence that later multi-turn QA may depend on.

H Limitation

Despite strong empirical results, OV-Encoder has several limitations. First, the codec-guided sparse formulation is most effective when compressed-domain side information, such as motion vectors and residuals, is available. In settings where only decoded RGB frames are accessible, the model must fall back to chunk-wise or frame-based processing, which may reduce the advantage of codec-guided token allocation. Moreover, the current formulation is motivated by standard codec decompositions and has not yet been systematically validated across a broad range of codec standards, compression levels, and real-world encoding pipelines.

Second, token sparsity does not automatically translate into proportional end-to-end system speedup. While OV-Encoder reduces the number of visual tokens processed by the transformer, practical efficiency also depends on video decoding, motion/residual extraction, data loading, and hardware-specific implementation. As a result, token efficiency and system efficiency are related but not identical, and a more complete study of latency, throughput, and energy efficiency remains necessary.

Third, the strongest gains of OV-Encoder are observed on motion-sensitive video benchmarks, whereas improvements on image and document understanding are more moderate. In addition, some document and OCR-intensive benchmarks may benefit from OCR-enriched pretraining data,

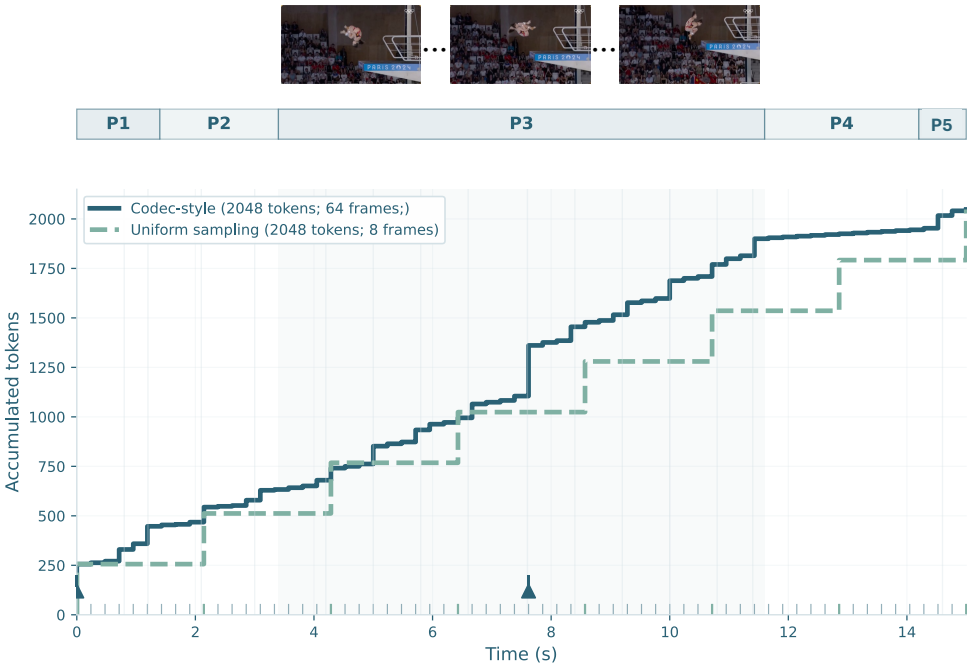


Figure 9: **Case study 1 (Diving): dense evidence for continuous motion.** Under a fixed budget of 2048 tokens, codec-style patch extraction distributes tokens over 64 sampled frames to capture dense pose transitions, while uniform sampling allocates the same budget to only 8 frames and may miss brief but critical instants. Phases: **P1** 0.0–1.4s (takeoff + angular momentum), **P2** 1.4–3.4s (tuck/pike set; body shape), **P3** 3.4–11.6s (twist+somersault progression; dense cues), **P4** 11.6–14.2s (spot water + open/align), **P5** 14.2–15.0s (entry alignment; line/splash). Diving is high-speed and continuous, so discriminative cues are temporally dense and benefit from dense coverage.

making it difficult to attribute all improvements solely to codec-guided sparsity. Finally, our results support codec-guided sparse patch allocation as an effective design principle for patch-efficient visual representation learning, but they do not establish it as a universal principle for all multimodal settings. Broader validation across domains, model scales, and downstream tasks remains important future work.

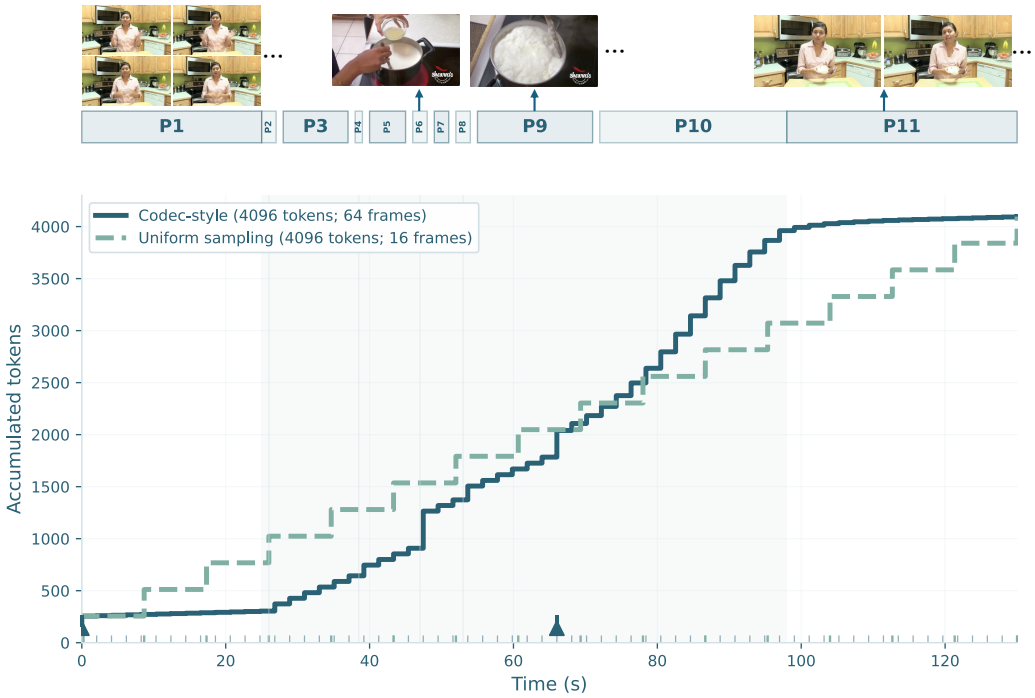


Figure 10: **Case study 2 (Cooking): sparse key frames under instantaneous motion.** We analyze a 130-second cooking video where decisive evidence concentrates in short, discrete transitions (e.g., brief pours) rather than being uniformly distributed over time. Under a fixed budget of 4096 tokens, codec-style allocation spreads tokens over a 64-frame timeline and reallocates P-frame tokens toward high-saliency moments, while uniform sampling allocates the same budget to only 16 frames and may miss brief segments entirely (misplacement). Phases: **P1** 0.0–25.0s (context), **P2** 25.0–27.0s (pour raw materials), **P3** 28.0–37.0s (mix raw materials), **P4** 38.0–39.0s (pour raw materials), **P5** 40.0–45.0s (mix raw materials), **P6** 46.0–48.0s (pour raw materials), **P7** 49.0–51.0s (mix raw materials), **P8** 52.0–54.0s (pour raw materials), **P9** 55.0–71.0s (mix raw materials), **P10** 72.0–98.0s (pour raw materials), **P11** 98.0–130.0s (context).

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We claim our contributions and scope in Introduction.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

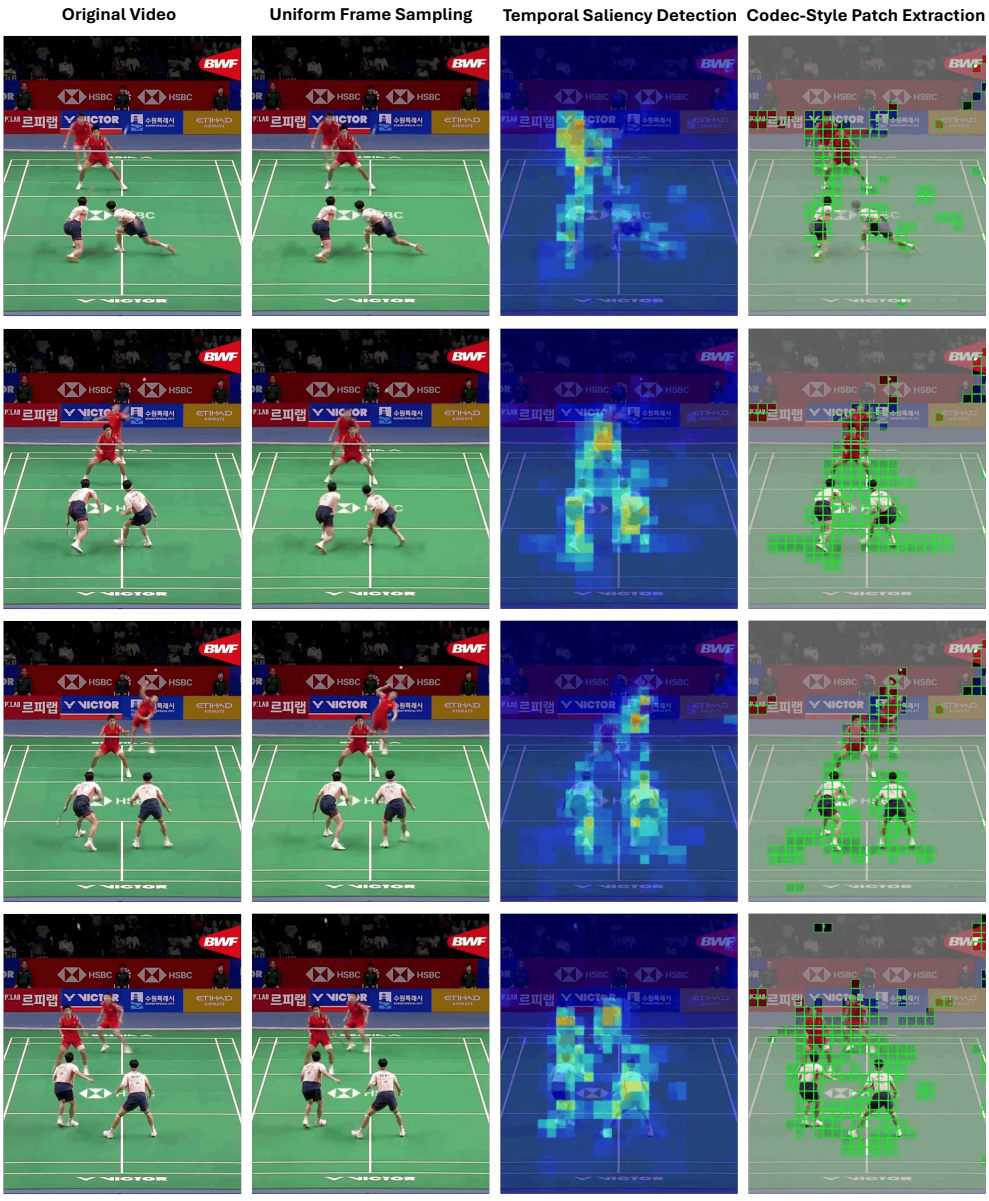


Figure 11: Comparison of video processing pipelines for spatiotemporal representation learning. (a) original dense video input with full temporal context, (b) uniform frame sampling that sparsely selects evenly spaced frames, (c) temporal saliency detection that identifies motion- and event-centric regions across all frames, and (d) Codec patch extraction that selectively retains temporally salient patches under a fixed token budget.

Justification: In the section of supplementary materials on limitations.

Guidelines: We discuss the limitations of our limitations in the last section, Conclusion&Limitation.

- The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: [N/A]

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: **[TODO]**

   Guidelines: We clearly provide the information of our experiment setting in the section, Implementation Details.

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We will open source data, models and data.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: **[TODO]**

   Guidelines: We clearly provide our experiment setting in the section, Baseline and Implementation details.

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [N/A]

   Justification: [N/A]

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: **[TODO]**

   Guidelines: We provide sufficient information on the computation resources in Implementation details.

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This paper strictly adheres to all requirements of the NeurIPS Code of Ethics, including transparency in data usage, fairness in research methods.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: [N/A]

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: [N/A]

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We explicitly acknowledge the original owners of the assets, including code, data, and models.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: [N/A]

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: [N/A]

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: [N/A]

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: We describe the details of how our model is used as the vision encoder in the LLM-based MLLM, along with the corresponding experimental settings, in UniViT as an Encoder of MLLM.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.