CoMo: Efficient Video Transformer via Codec Residual and Motion-Field Geometry

Anonymous authors
Paper under double-blind review

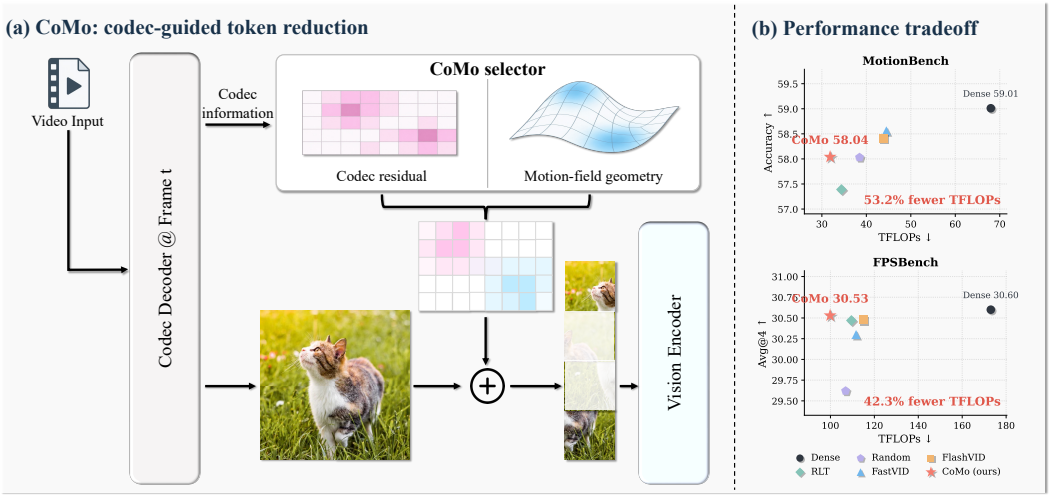


Figure 1: **Efficient video understanding with CoMo.** (a) CoMo combines codec residuals and motion-field geometry to select informative tokens before the vision encoder, without additional training or architectural modifications. (b) On Qwen3-VL$_{8B}$, CoMo uses 53.2% fewer TFLOPs on MotionBench and 42.3% fewer on FPSBench than dense inference, with comparable accuracy.

Abstract

Transformers have demonstrated remarkable capabilities in video understanding, but their computational cost grows rapidly with the large number of spatiotemporal tokens. Existing token reduction approaches commonly rely on learned selection modules, intermediate model features, or comparisons between spatially co-located patches, introducing additional overhead, requiring redundant tokens to be partially processed, or failing to recognize redundancy under spatial displacement. Video codecs, in contrast, have already characterized temporal redundancy through motion compensation and residual coding. In this paper, we systematically investigate codec-derived residuals and motion vectors as lightweight signals for video token reduction. We find that residuals provide an effective measure of information not explained by motion-compensated prediction, whereas raw motion-vector magnitude captures absolute displacement but overlooks spatial variations in the motion field. Based on these findings, we introduce **CoMo**, a training-free and model-agnostic framework that combines codec residuals with motion-field geometry to adaptively remove motion-compensated redundancy before the vision encoder. Across action recognition and video language understanding benchmarks, CoMo consistently improves the tradeoff between performance and efficiency, reducing VLM computation by up to 53.2% and accelerating inference by up to 1.66× while retaining accuracy comparable to dense inference.

1 Introduction

Recently, vision transformers (Dosovitskiy et al., 2020) have achieved remarkable success across a wide range of visual understanding tasks (Feichtenhofer et al., 2022; Caron et al., 2021), yet extending them efficiently to video remains challenging. Due to the massive computation of spatiotemporal

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



Figure 2: **Motivation for CoMo.** Given RGB frames (a), RLT differences (b) respond broadly to displacement, while optical-flow compensation (c) reduces these responses with additional computational cost. Codec residuals (d) and motion vectors (e) provide motion and prediction-error information directly during decoding.

self-attention and excessive number of tokens generated across video frames, the computational cost scales quadratically with the sequence length. Consequently, processing dense video tokens incurs prohibitive memory overhead and high inference latency. As a result, the majority of research on video understanding is still restricted to short, low resolution videos and far from fully capturing complex temporal dynamics and motions.

To solve this efficiency bottleneck, several works have attempted to address this problem from different directions. Firstly, some works aim to replace the complete self-attention architecture with sparse attention (Yang et al., 2026b) or adaptive window mechanisms (Zhang et al., 2022) to mitigate the sheer computational cost driven by long video sequence lengths. However, these architectural modifications often require large-scale training or non-trivial fine-tuning to retain model capability and lack flexibility for off-the-shelf models. In contrast, token reduction techniques remove or merge uninformative tokens before (Choudhury et al., 2025) or during network processing (Rao et al., 2021; Bolya et al., 2023; Bolya & Hoffman, 2023) without substantially modifying the underlying transformer architecture. By preserving the pretrained backbone and its parameters, these methods can retain much of the original model's capabilities while reducing the need for extensive adaptation.

Despite their flexibility, existing token reduction methods can generally be divided into two categories and have their own limitations. (1) Methods that progressively prune or merge tokens within the transformer (Rao et al., 2021; Bolya et al., 2023; Bolya & Hoffman, 2023) either require additional training or process redundant tokens in the early layers before removing them. (2) Pre model reduction methods, such as RLT (Choudhury et al., 2025), remove redundant tokens before model processing by comparing pixel differences between temporally neighboring patches. However, as shown in Figure 2 (b), such comparisons do not account for object or camera motion and can therefore mistake displaced but otherwise redundant content for new visual information, limiting the achievable token reduction and retaining unnecessary computation.

To successfully identify temporal redundancy under motion, the correspondence between content across frames must be considered. Although optical flow can establish such correspondence and reduce displacement-induced differences, as illustrated in Figure 2 (c), running an additional optical-flow model, even with modern GPU acceleration (Teed & Deng, 2020; Wang et al., 2024), introduces overhead that can partially offset the efficiency gains from token reduction. This naturally leads to another question: *Can motion information be obtained without running an additional motion-estimation model?* The answer lies in video codecs. During compression, video codecs predict

frames from reference frames using block-wise motion compensation and encode both the resulting motion vectors and prediction residuals in the bitstream. Together, these codec signals provide a readily available and computationally inexpensive description of motion correspondence and unexplained visual changes, as illustrated in Figure 2 (d–e), motivating their reuse for token selection.

Motivated by this observation, we propose **CoMo**, a training-free token reduction framework that combines **Co**dec residuals with **Mo**tion-field geometry while preserving the original architecture of the video transformer. As illustrated in Figure 1 (a), CoMo combines codec residuals, which quantify visual information unexplained by motion-compensated prediction, with motion-field geometry, which characterizes local expansion and contraction through an area-change score. CoMo retains a patch when either signal exceeds its corresponding threshold, enabling adaptive token selection before the vision encoder. As shown in Figure 1(b), CoMo reduces computation on Qwen3-$\text{VL}_{8\text{B}}$ by 53.2% on MotionBench (Hong et al., 2025a) and 42.3% on FPSBench (Choudhury et al., 2026) relative to dense inference while maintaining comparable accuracy.

In summary, our contributions are as follows:

- We systematically analyze codec residuals and motion vectors as token-selection signals and compare them with alternative information. Our results show that codec-derived signals provide a robust and efficient solution across others.
- We introduce **CoMo**, a training-free and model-agnostic token-reduction framework that combines codec residuals with motion-field geometry to remove motion-compensated redundancy before the vision encoder, without modifying the underlying model architecture.
- We conduct extensive experiments and ablation studies across video recognition and video-language models, demonstrating that CoMo consistently improves the accuracy–efficiency trade-off over existing training-free token reduction methods.

2 Related work

Efficient Vision Transformers. Vision transformers (Dosovitskiy et al., 2020; Vaswani et al., 2017) have demonstrated strong results on images and video (Arnab et al., 2021; Bertasius et al., 2021; Fan et al., 2021; Li et al., 2022; Ryali et al., 2023), but dense spatiotemporal tokenization makes their computational cost grow rapidly with video duration, frame rate, and spatial resolution. One line of works attempt to address the issue through specialized architectures, including multi-rate processing (Feichtenhofer et al., 2019), temporal memory (Ravi et al., 2024; Wu et al., 2022), linear-complexity sequence models (Li et al., 2024), and content-dependent routing (Raposo et al., 2024). Although effective, these approaches generally require architectural modifications or task-specific training. Therefore, a complementary direction improves efficiency by reducing the number of visual tokens processed by the model. Learned pruning methods estimate token importance from intermediate features (Rao et al., 2021; Yin et al., 2022; Meng et al., 2022), whereas training-free methods merge or discard tokens according to their feature similarity or attention scores (Bolya et al., 2023; Bolya & Hoffman, 2023; Shen et al., 2026; Fan et al., 2026). However, because these methods operate within or after the vision encoder, the discarded tokens still incur part or all of the encoding cost. In comparison, RLT (Choudhury et al., 2025) instead removes temporally redundant patches before the encoder by comparing RGB patches. Such comparisons, however, use appearance similarity only as an indirect proxy for redundancy and may fail to identify content that remains predictable after spatial displacement, motivating us to seek a more direct, motion-aware measure of redundancy from signals already available in video codecs.

Motion Cues and Compressed-Domain Video Understanding. Optical flow is widely used for motion representation (Horn & Schunck, 1981; Efros et al., 2003; Laptev et al., 2008; Wang & Schmid, 2013; Wang et al., 2013; Teed & Deng, 2020; Wang et al., 2024) and can reveal temporal redundancy under spatial displacement (shown in fig. 2). However, optical flow estimation introduces additional computation that can offset the efficiency gains from token reduction. As illustrated in table 1, Codec motion vectors and residuals provide lower-cost temporal cues and have been exploited for efficient action recognition (Zhang et al., 2016; Wu et al.,

Motion Source	Device	Time (ms)
Optical flow (Farneback)	CPU	104.45
Optical flow (RAFT)	GPU	49.19
Codec MV	CPU	6.46

Table 1: Motion extraction cost.

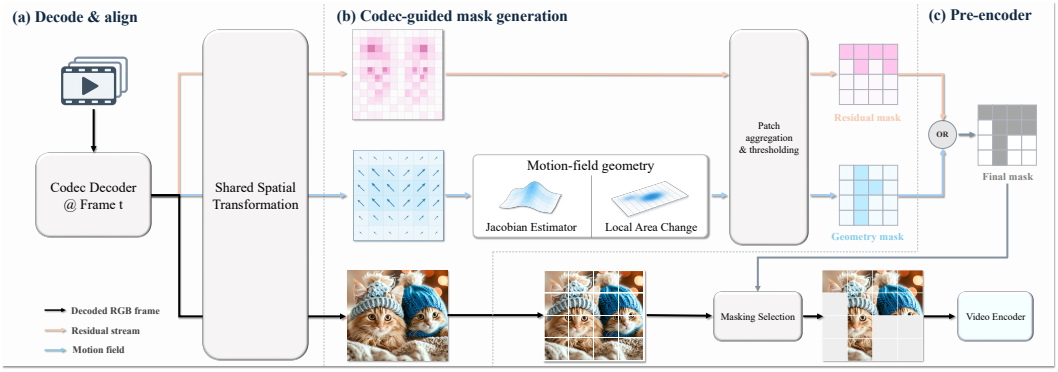


Figure 3: **Overview of CoMo.** (a) RGB frames, codec residuals, and motion vectors are extracted during decoding and aligned through shared spatial transformations. (b) Residual and local area-change scores are aggregated and thresholded into patch-selection masks. (c) OR fusion combines the masks to select patches before the vision encoder, without model training or modifications.

2018; Chen & Ho, 2022). More recently, Mage-VL (Yang et al., 2026a) develops a codec-native multimodal model supporting streaming video understanding. A more detailed discussion of concurrent work can be found in the Appendix.

3 METHOD

Consider a video transformer that takes a video $V \in \mathbb{R}^{C\times T\times H\times W}$ as input. Standard tokenization partitions V into a set of non-overlapping spatiotemporal patches $\mathbf{P} = \{P_i\}_{i=1}^{N_P}$, each of size $C \times D_t \times D_h \times D_w$. A patch embedding layer $\mathcal{E}$ then projects each patch into a d_{embed}-dimensional embedding, producing N_P tokens, each corresponding to a distinct spatiotemporal location. Consequently, the number of tokens is determined entirely by the input dimensions and remains fixed regardless of the temporal redundancy in the video.

In this work, we aim to construct an input-dependent masking function that assigns a binary decision $\mathcal{M}(P_i) \in \{0, 1\}$ to each patch before the transformer encoder, where 1 indicates retention and 0 indicates removal. This produces a compact subset $\widetilde{\mathbf{P}} = \{P_i \in \mathbf{P} \mid \mathcal{M}(P_i) = 1\}$ while preserving downstream performance without modifying or fine-tuning the underlying transformer. Although video compression targets reconstruction and token reduction targets semantic understanding, both rely on identifying temporal redundancy, motivating us to use codec-derived signals for token selection.

3.1 PRELIMINARY – VIDEO CODEC

Modern video codecs, such as H.264 (Wiegand et al., 2003), reduce spatial and temporal redundancy through block-based predictive coding. Each frame is partitioned into different coding blocks. Intra-coded blocks are predicted from previously reconstructed neighboring samples within the same frame, whereas inter-coded blocks are predicted from reconstructed reference frames through motion compensation. We use $V_t^{\text{pred}}(\boldsymbol{x})$ to denote the prediction produced by either coding mode. For an inter-coded block, the prediction at spatial location $\boldsymbol{x} = (x, y)$ in frame t is given by

$$V_t^{\text{pred}}(\boldsymbol{x}) = V_r^{\text{rec}}\big(\boldsymbol{x} + \boldsymbol{d}_t(\boldsymbol{x})\big), \tag{1}$$

where V_r^{rec} denotes the reconstructed reference frame selected for the block, and $\boldsymbol{d}_t(\boldsymbol{x})$ denotes the block-wise displacement from $\boldsymbol{x}$ in the current frame to its sampling location in the reference frame. For either coding mode, the encoder computes the prediction residual as

$$\mathbf{e}_t(\boldsymbol{x}) = V_t(\boldsymbol{x}) - V_t^{\text{pred}}(\boldsymbol{x}), \tag{2}$$

which is transformed and quantized into residual coefficients stored in the compressed bitstream. During decoding, dequantization and inverse transformation produce the decoded residual $\widetilde{\mathbf{e}}_t$, which

is added to the prediction to reconstruct the frame:

$$V_t^{\mathrm{rec}}(\boldsymbol{x}) = V_t^{\mathrm{pred}}(\boldsymbol{x}) + \widetilde{\mathbf{e}}_t(\boldsymbol{x}). \tag{3}$$

Residuals and motion vectors provide complementary information for token selection. Residuals quantify information unexplained by codec prediction for both intra- and inter-coded blocks, whereas motion vectors describe temporal correspondence for inter-coded blocks. CoMo therefore uses residuals from both coding modes and motion vectors at valid inter-coded locations. Because these signals are already computed by the codec, using them does not require an additional neural motion estimator.

3.2 CoMo

As illustrated in Figure 3, CoMo constructs an input-dependent token masking function $\mathcal{M}$ from decoded residuals and codec motion vectors. During decoding, we extract the signed residual map $\widetilde{\mathbf{e}}_t$ and block-wise displacement $\boldsymbol{d}_t$, together with a binary validity map q_t indicating locations with valid inter-frame motion information. Locations without an inter-frame reference, including intra-coded regions, are marked as invalid rather than assigned zero motion. We convert the block-wise quantities into spatial maps and apply the same temporal sampling, resizing, and cropping operations used for the RGB frames, thereby aligning all codec signals with the transformer patches. Finally, we obtain a temporally normalized motion field by dividing each valid displacement by its signed reference-frame distance:

$$\boldsymbol{v}_t(\boldsymbol{x}) = \frac{\boldsymbol{d}_t(\boldsymbol{x})}{\Delta_t(\boldsymbol{x})}, \tag{4}$$

where $\Delta_t(\boldsymbol{x})$ denotes the signed frame distance from frame t to the corresponding reference frame.

Residual score. The decoded residual $\widetilde{\mathbf{e}}_t(\boldsymbol{x})$ quantifies the visual information not captured by codec prediction: For an intra-coded block, it reflects the error of spatial prediction within the current frame; for an inter-coded block, it represents the error remaining after motion-compensated prediction. We apply the linear YUV-to-RGB transformation to $\widetilde{\mathbf{e}}_t$ and denote the resulting aligned RGB residual by $\widetilde{e}_{t,c}^{\mathrm{RGB}}(\boldsymbol{x}) \in [-255, 255]$. For a spatiotemporal patch P_i, we define its residual score as

$$r_i = \max_{t \in \mathcal{T}_i} \frac{1}{|\Omega_i| C} \sum_{\boldsymbol{x} \in \Omega_i} \sum_{c=1}^{C} \left| \widetilde{e}_{t,c}^{\mathrm{RGB}}(\boldsymbol{x}) \right|, \tag{5}$$

where Ω_i and $\mathcal{T}_i$ denote the spatial and temporal region of P_i, respectively, and $C = 3$ is the number of RGB channels. Spatial averaging measures the mean residual magnitude within each token, while temporal max pooling preserves brief but informative prediction errors within the temporal patch.

Motion-geometry score. The motion-vector magnitude measures absolute displacement, but cannot distinguish globally consistent translation from locally varying motion. We therefore use spatial variation among neighboring vectors in the temporally normalized motion field as the basis for our motion-geometry score. Writing the motion field as $\boldsymbol{v}_t(\boldsymbol{x}) = [v_t^x(\boldsymbol{x}), v_t^y(\boldsymbol{x})]^\top$, we characterize its local spatial variation using the Jacobian

$$\mathbf{J}_t(\boldsymbol{x}) = \nabla \boldsymbol{v}_t(\boldsymbol{x}) = \begin{bmatrix} \dfrac{\partial v_t^x}{\partial x} & \dfrac{\partial v_t^x}{\partial y} \\ \dfrac{\partial v_t^y}{\partial x} & \dfrac{\partial v_t^y}{\partial y} \end{bmatrix}. \tag{6}$$

A spatially constant motion field yields $\mathbf{J}_t(\boldsymbol{x}) = \mathbf{0}$ regardless of its motion magnitude, whereas spatially varying motion produces nonzero derivatives. A straightforward scalar summary of the Jacobian is its Frobenius norm. However, the Frobenius norm weights all derivative components equally and therefore responds to expansion, rotation, and shear alike. We instead seek a more selective descriptor that measures whether neighboring trajectories expand or contract the local area they occupy. To quantify this effect, we consider the mapping induced by the normalized motion field over one frame step. Each location $\boldsymbol{x}$ is mapped to its displaced position

$$\boldsymbol{\phi}_t(\boldsymbol{x}) = \boldsymbol{x} + \boldsymbol{v}_t(\boldsymbol{x}). \tag{7}$$

Differentiating this mapping yields the local deformation gradient

$$\mathbf{F}_t(\boldsymbol{x}) = \nabla \boldsymbol{\phi}_t(\boldsymbol{x}) = \mathbf{I} + \mathbf{J}_t(\boldsymbol{x}), \tag{8}$$

where $\det(\mathbf{F}_t)$ gives the signed local area-scaling factor. We define the pixel-level area-change response as

$$A_t(\boldsymbol{x}) = |\det(\mathbf{F}_t(\boldsymbol{x})) - 1| = |\det(\mathbf{I} + \mathbf{J}_t(\boldsymbol{x})) - 1|, \tag{9}$$

which is zero for locally area-preserving motion and increases with local expansion or contraction. As shown in Table 2, local area change achieves the highest Top-1 accuracy and competitive Top-5 accuracy. We therefore use $A_t(\boldsymbol{x})$ as the pixel-level motion-geometry response and obtain the token-level score a_i through spatial averaging followed by temporal max pooling, as for r_i. Spatial averaging includes only locations with valid motion responses, and a_i is set to zero when none are available within P_i.

Motion descriptor	Top-1 ↑	Top-5 ↑
Jacobian norm	68.77	91.27
Divergence	68.84	**91.30**
Curl	68.80	91.29
Shear	68.77	91.23
Local area change	**68.86**	91.25

Table 2: Motion-field descriptors.

Signal fusion and token selection. The residual and motion-geometry scores capture complementary information: r_i measures visual content unexplained by codec prediction, while a_i measures motion-induced local area change. To accommodate their different numerical scales, CoMo thresholds each score separately and retains a patch if either threshold is exceeded:

$$\mathcal{M}(P_i) = [r_i > \tau_r \text{ or } a_i > \tau_a], \tag{10}$$

where τ_r and τ_a denote the residual and motion-geometry thresholds, respectively. The union preserves patches containing either substantial prediction error or informative local motion. Because CoMo uses fixed score thresholds rather than a fixed Top-K budget, the number of retained tokens adapts to the information content of each video.

4 Experiments

4.1 Experimental Setup

To evaluate the effectiveness and generality of CoMo, we conduct experiments on standard action-recognition benchmarks and extend our evaluation to video-language models on MotionBench (Hong et al., 2025a) and FPSBench (Choudhury et al., 2026). For action recognition, we use fixed Top-K token budgets to isolate the quality of the selected tokens under identical computational constraints; for video-language understanding, we employ threshold-based selection to adapt the number of retained tokens to the information content and temporal complexity of each video. For a fair comparison and to reflect practical deployment, all videos are used in their original compressed form without re-encoding, so the codec signals are exactly those produced by the source bitstreams and no encoding cost is incurred at inference time.

Metrics. For action recognition, we report top-1 and top-5 classification accuracy. For video-language understanding, we report accuracy on MotionBench and Average@4 on FPSBench. We measure computational efficiency using estimated TFLOPs per video across all model components and report inference speedup relative to dense-token inference. All runtime measurements are conducted on a single NVIDIA A100 80GB GPU under the same evaluation configuration.

Implementation details. We build our implementation on the official VideoMAE (Tong et al., 2022; Wang et al., 2023), Qwen3-VL (Bai et al., 2025), and GLM-4.1 (Hong et al., 2025b) checkpoints without modifying or fine-tuning their model parameters. To obtain codec information, we lightly modify the H.264 and VP9 decoders in FFmpeg and expose the extracted signals through a modified `decord` video reader. During decoding, we export the signed prediction residuals in YUV space, block-wise motion vectors, reference-frame information, and block coding modes. These signals are generated as part of the standard decoding process and require neither pixel-based motion recomputation nor an additional neural motion estimator. To further reduce overhead, side information is materialized only for frames selected by the model's temporal sampler. Unless otherwise stated, CoMo uses the same residual and motion-geometry thresholds, $(\tau_r, \tau_a) = (0.6, 0.2)$, for all threshold-based evaluations, without model or dataset specific retuning. Additional codec-specific implementation details are provided in the Appendix.

ID	Method	Kinetics-400		Something-Something V2	
		50%	60%	50%	60%
–	Dense inference	79.68		70.53	
(a)	RGB difference (RLT)	78.85	79.18	69.46	69.89
(b)	Codec residual	78.89 (+0.04)	79.22 (+0.04)	69.76 (+0.30)	69.99 (+0.10)
(c)	Motion-vector magnitude	77.57	78.60	68.16	69.16
(d)	Motion Jacobian norm	77.81 (+0.24)	78.82 (+0.22)	69.71 (+1.55)	70.16 (+1.00)
(e)	Local area change	77.83 (+0.26)	78.92 (+0.32)	69.72 (+1.56)	70.18 (+1.02)

Table 3: **Roadmap from codec signals to effective token selection.** We report top-1 accuracy under fixed top-K token budgets. Rows (a) – (b) compare residual-based signals, while rows (c) – (e) progressively transform codec motion vectors into motion-geometry descriptors. Each signal is evaluated independently without score fusion.

4.2 Performance Analysis

Roadmap to effective token selection. Before evaluating the complete CoMo framework, we first isolate the effectiveness of each selection signal under a controlled Top-K setting. For each signal, we rank the spatiotemporal patches within every video and retain the same fraction of tokens. Therefore, differences in accuracy reflect which tokens are selected rather than how many are retained. As shown in Table 3, codec residuals consistently outperform RGB differences (Choudhury et al., 2025) across both dataset and token budget. For motion-based selection, directly ranking tokens by motion-vector magnitude produces relatively weak results because it captures absolute displacement without accounting for spatial variation. Characterizing the motion field using the Jacobian norm substantially improves accuracy, where recognizing actions relies strongly on temporal dynamics. Local area change offers a clear geometric interpretation through the determinant of the deformation gradient, while providing slight but consistent improvements over the Jacobian norm across the evaluated settings. Furthermore, we find that the relative effectiveness of the two signal families also varies across datasets and token budgets: residuals perform better on Kinetics-400 (Kay et al., 2017), whereas motion geometry becomes more competitive on SSv2 (Goyal et al., 2017) and outperforms residuals at 60% token retention, which motivates the fusion between two signals.

Benefits of combining signals. We evaluate the contribution of combining codec residuals and motion-field geometry using Qwen3-VL$_{8B}$ on MotionBench (Hong et al., 2025a) and FPSBench (Choudhury et al., 2026). We compare residual-only selection, geometry-only selection, and CoMo at comparable computational costs. As shown in Table 4, the relative performance of the individual signals varies across benchmarks: residual-only selection performs better on MotionBench, whereas geometry-only selection achieves a higher score on FPSBench. Nevertheless, by combining the two signals, CoMo matches the stronger individual signal on both benchmarks, achieving 58.04 accuracy on MotionBench and 30.52 Average@4 on FPSBench without requiring benchmark-specific selection of a single signal. Additional comparisons of fusion strategies are provided in the Appendix.

	MotionBench		FPSBench	
Signal	Acc.	TFLOPs	Avg.@4	TFLOPs
Residual Only	58.04	31.21	30.10	98.05
Geometry Only	57.34	30.76	30.50	99.57
CoMo (Ours)	**58.04**	31.85	**30.52**	99.84

Table 4: Signal fusion on Qwen3-VL$_{8B}$ at comparable computational costs.

Main comparison. As shown in Table 5, we compare CoMo with representative training-free token-reduction methods across three VLM architectures on MotionBench (Hong et al., 2025a) and FPSBench (Choudhury et al., 2026). We select VLMs that natively support variable-resolution visual inputs, allowing us to apply the same input configuration and hyperparameters across architectures. Among methods that remove tokens before the vision encoder, CoMo consistently outperforms RLT (Choudhury et al., 2025) and random selection in all models, demonstrating that codec residuals and motion geometry provide reliable signals to identify informative visual tokens. Compared with ToMe (Bolya et al., 2023), FastVID (Shen et al., 2026), and FlashVID (Fan et al., 2026), which reduce tokens within or after the vision encoder, CoMo achieves comparable or better accu-

Model	MotionBench			FPSBench		
	Acc.↑	TFLOPs↓	Speedup↑	Avg@4↑	TFLOPs↓	Speedup↑
Qwen3-VL$_{8B}$	59.01	67.99	1.00×	30.60	173.03	1.00×
ToMe*	57.59	47.53	0.60×	29.60	151.34	0.57×
FastVID*	**58.56**	44.50	1.17×	30.30	111.67	1.16×
FlashVID*	58.41	43.87	0.98×	30.48	114.96	1.06×
Random	58.03	38.47	1.33×	29.62	106.90	1.29×
RLT	57.39	34.40	1.33×	30.47	109.54	1.18×
CoMo (ours)	58.04	**31.85**	**1.35×**	**30.53**	**99.84**	**1.30×**
GLM-4.1V$_{9B}$	57.07	152.10	1.00×	29.85	547.70	1.00×
ToMe*	56.77	126.11	0.57×	29.35	517.38	0.59×
FastVID*	**56.94**	115.30	1.24×	29.45	389.15	1.43×
FlashVID*	56.89	114.61	1.00×	29.57	398.71	1.27×
Random	56.05	87.17	1.55×	28.60	329.02	1.62×
RLT	55.38	77.03	1.52×	29.20	327.34	1.53×
CoMo (ours)	56.57	**71.81**	**1.58×**	**29.58**	**302.14**	**1.64×**
Qwen2.5-VL$_{7B}$	57.14	86.81	1.00×	30.00	402.97	1.00×
ToMe*	56.12	64.89	0.49×	29.72	376.61	0.32×
FastVID*	56.42	58.68	1.20×	29.50	241.29	1.42×
FlashVID*	**56.45**	58.06	0.91×	**29.95**	256.49	1.21×
Random	55.80	50.78	**1.54×**	29.58	238.90	1.60×
RLT	56.27	44.79	1.47×	28.55	232.86	1.56×
CoMo (ours)	56.35	**41.88**	1.49×	29.55	**216.29**	**1.66×**

Table 5: **Comparison with training-free token-reduction methods.** Following the standard settings of each benchmark, we sample videos at 1 FPS on MotionBench and 8 FPS on FPSBench. * denotes methods that reduce tokens within or after the vision encoder.

Signal	H.264	VP9
Res.	70.20	69.99
Area	69.98	70.18

(a) Codec type

τ_r	0.50	0.60	0.70
Acc.	57.34	57.12	56.60
TFLOPs	28.24	25.76	23.68

(b) Residual-only threshold

τ_a	0.15	0.20	0.25
Acc.	56.99	56.97	56.30
TFLOPs	28.90	26.15	24.17

(c) Geometry-only threshold

Stage	RLT	CoMo	Δ
Post-ViT	58.54	58.49	-0.05
Pre-ViT	57.39	58.04	+0.65

(d) Pruning stage

Table 6: **Ablation studies of CoMo.**

racy while requiring substantially less computation. In MotionBench (Hong et al., 2025a), CoMo requires more than 20% fewer compute costs than FastVID (Shen et al., 2026) and FlashVID (Fan et al., 2026), two recent state-of-the-art video token-reduction methods, while maintaining comparable accuracy across all three VLM architectures. Under the high-frame-rate setting of FPSBench (Choudhury et al., 2026), CoMo demonstrates an even stronger advantage: it achieves the highest accuracy among all token reduction methods in Qwen3-VL (Bai et al., 2025) and GLM-4.1V (Hong et al., 2025b) and lags behind the best-performing method by only 0.40% on Qwen2.5-VL (Bai et al., 2025), while consistently requiring the fewest TFLOPs across all three architectures.

4.3 More Analysis

Ablation studies. As shown in Table 6, We conduct the codec ablation in Table 6a using VideoMAE-B (Tong et al., 2022) on SSv2 (Goyal et al., 2017), retaining 60% of tokens through Top-K selection. The threshold and pruning-stage ablations in Tables 6b to 6d use Qwen3-VL$_{8B}$ (Bai et al., 2025) on MotionBench (Hong et al., 2025a). As summarized in table 6a , we encode the same source videos using H.264 and VP9 while keeping temporal sampling, spatial preprocessing, and the token budget fixed. Although their relative performance changes with the codec, both signals achieve similar accuracy across H.264 and VP9. Next, as shown in Tables 6b and 6c, increasing either threshold progressively reduces both computational cost and accuracy. The intermediate settings $\tau_r = 0.60$ and $\tau_a = 0.20$ offer a favorable trade-off, reducing the computation by 2.48 and 2.75 TFLOPs with accuracy drops of only 0.22 and 0.02, respectively. Therefore, we adopt $\tau_r = 0.60$ and $\tau_a = 0.20$ as the default thresholds for all threshold-based experiments. Finally, Table 6d compares the pruning before and after the vision encoder with matched token budgets. CoMo matches

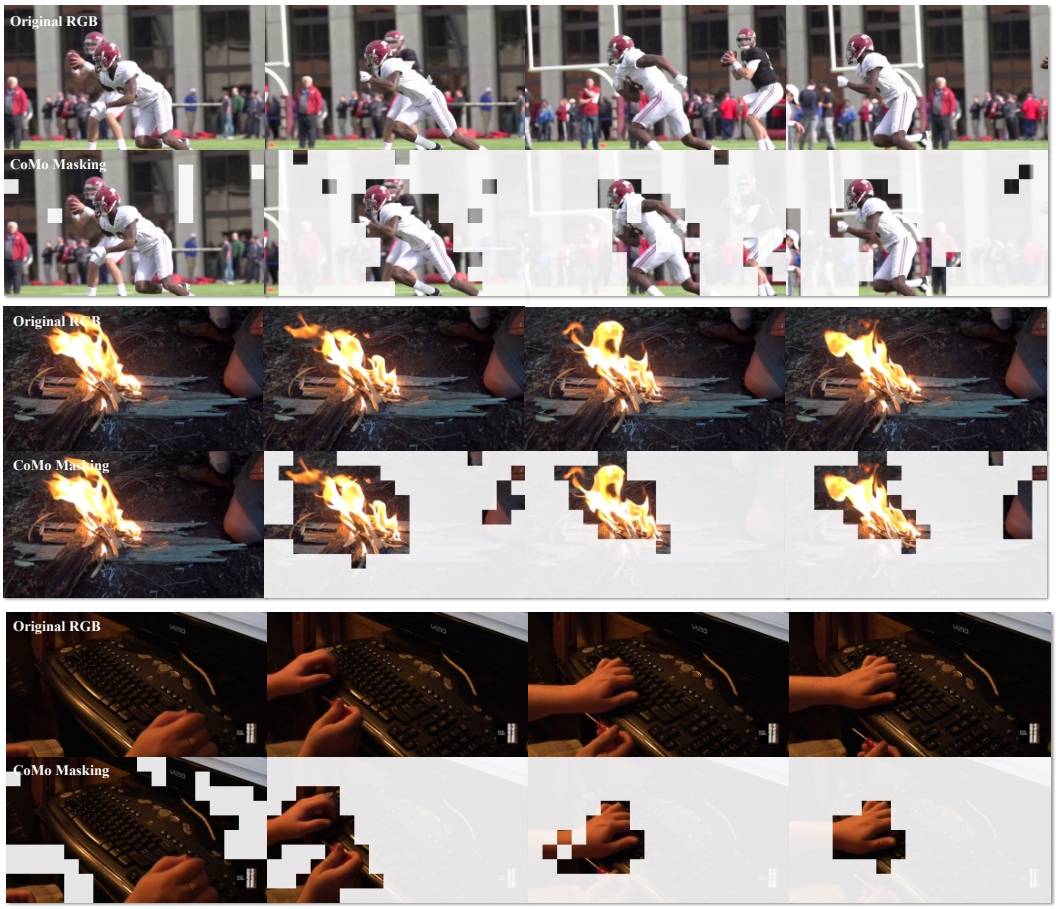


Figure 4: **Sample visualization.** Top: original RGB frames. Bottom: tokens selected by CoMo, with discarded tokens faded.

RLT under post-ViT pruning but outperforms it under pre-ViT pruning. We therefore adopt pre-ViT pruning, which provides stronger input-level token selection while eliminating vision-encoder computation for discarded tokens. Further representation analysis and additional ablation studies are provided in the Appendix.

Visualization. As visualized in Figure 4, our method consistently retains tokens on moving objects, such as the typing hands, the running player, and the flickering flame, while discarding most of the static background.

5 CONCLUSION

In this work, we present CoMo, a training-free framework that reuses codec residuals and motion-field geometry to select informative tokens before the vision encoder. Our analysis shows that residuals measure content unexplained by motion-compensated prediction and match or exceed RGB differences, while motion-field geometry captures local variations overlooked by raw motion-vector magnitude. Since neither signal dominates across datasets, fusing them with a single pair of thresholds matches the stronger individual signal on every benchmark without per-dataset signal selection or retuning. Experiments demonstrate improved accuracy–efficiency trade-offs across multiple pre-trained models, with up to 53.2% lower computation and $1.66\times$ inference speedup.

Limitations. Although CoMo uses a single set of thresholds across the evaluated datasets, it relies on hand-designed selection criteria rather than learned task-specific importance. Consequently, it may discard semantically relevant tokens with weak residual and motion-geometry responses.

AI USE STATEMENT

In this work, we used generative AI tools to assist in writing the derivations in Appendix C and to provide feedback on the research methodology and experimental design. We did not use generative AI tools to propose or refine hypotheses, formulate mathematical claims, implement methods, or interpret results. The remaining tasks requiring disclosure, namely generating synthetic datasets, developing theoretical models, translation, data cleaning, and qualitative analysis, are not applicable to this work. Additionally, we used generative AI tools to assist with coding, draft and edit parts of the paper to improve readability. We did not use generative AI tools to create or modify figures or to format references; all references were exported from the original publication records and verified by the authors. We have reviewed all AI-assisted work: the derivations in Appendix C were independently checked by the authors, AI-assisted code was tested and verified by the authors, all AI-suggested text was revised by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work investigates efficient video understanding using existing benchmarks and pretrained models. CoMo does not explicitly address biases or privacy concerns associated with the underlying datasets and models. Since token reduction can affect model predictions, deployment in sensitive applications requires evaluating its impact on reliability and fairness.

REPRODUCIBILITY STATEMENT

Section 3 describes the codec-derived signals, motion-field geometry, and token-selection procedure used by CoMo. Section 4 specifies the evaluated models and benchmarks, evaluation metrics, and experimental settings. All experiments use pretrained model checkpoints without modifying or fine-tuning their parameters. The Appendix provides additional implementation details on codec-signal extraction, motion-field processing, Jacobian estimation, and latency measurement, as well as analyses of fusion strategies, selection thresholds, temporal sampling rates, and model sizes. Together, these details specify how codec signals are processed, how token-selection masks are constructed, and how performance and efficiency are evaluated. We will release our implementation, including the modified video decoders and evaluation configurations, to facilitate reproducibility.

REFERENCES

Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6836–6846, 2021.

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 preprint arXiv:2506.09985*, 2025.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *Proceedings of the International Conference on Machine Learning (ICML)*, July 2021.

Daniel Bolya and Judy Hoffman. Token merging for fast stable diffusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4599–4603, 2023.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your ViT but faster. In *International Conference on Learning Representations*, 2023.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *2021 IEEE/CVF international conference on computer vision (ICCV)*, pp. 9630–9640. IEEE, 2021.

Jiawei Chen and Chiu Man Ho. Mm-vit: Multi-modal video transformer for compressed video action recognition. In *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 786–797. IEEE, 2022.

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*, 37:28127–28149, 2025.

Rohan Choudhury, Jean-Sebastien Dandurand, Kai Qiu, Kshitij Madhav Bhat, Kartik Sharma, Liza Dahiya, Yizhou Zhao, Souraja Kundu, Chun-Hsien Lin, Kris M Kitani, et al. Fps-bench: A benchmark for high frame-rate video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18598–18608, 2026.

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 preprint arXiv:2010.11929*, 2020.

Alexei A. Efros, Alexander C. Berg, Greg Mori, and Jitendra Malik. Recognizing action at a distance. In *IEEE International Conference on Computer Vision*, pp. 726–733, Nice, France, 2003.

Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6824–6835, 2021.

Ziyang Fan, Keyu Chen, Ruilong Xing, Yulin Li, Li Jiang, and Zhuotao Tian. Flashvid: Efficient video large language models via training-free tree-based spatiotemporal token merging. *arXiv preprint arXiv:2602.08024*, 2026.

Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6202–6211, 2019.

Christoph Feichtenhofer, Haoqi Fan, Yanghao Li, and Kaiming He. Masked autoencoders as spatiotemporal learners. *arXiv:2205.09113*, 2022.

Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The" something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE international conference on computer vision*, pp. 5842–5850, 2017.

Wenyi Hong, Yean Cheng, Zhuoyi Yang, Weihan Wang, Lefan Wang, Xiaotao Gu, Shiyu Huang, Yuxiao Dong, and Jie Tang. Motionbench: Benchmarking and improving fine-grained video motion understanding for vision language models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8450–8460. IEEE, 2025a.

Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, Guobing Gan, Haomiao Tang, Jiale Cheng, Ji Qi, Junhui Ji, Lihang Pan, et al. Glm-4.5 v and glm-4.1 v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning. *arXiv preprint arXiv:2507.01006*, 2025b.

Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3): 185–203, 1981.

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.

Ivan Laptev, Marcin Marszalek, Cordelia Schmid, and Benjamin Rozenfeld. Learning realistic human actions from movies. In *2008 IEEE conference on computer vision and pattern recognition*, pp. 1–8. IEEE, 2008.

Kunchang Li, Xinhao Li, Yi Wang, Yinan He, Yali Wang, Limin Wang, and Yu Qiao. Videomamba: State space model for efficient video understanding. In *European Conference on Computer Vision*, pp. 237–255. Springer, 2024.

Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Mvitv2: Improved multiscale vision transformers for classification and detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4804–4814, 2022.

Lingchen Meng, Hengduo Li, Bor-Chun Chen, Shiyi Lan, Zuxuan Wu, Yu-Gang Jiang, and Ser-Nam Lim. Adavit: Adaptive vision transformers for efficient image recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12309–12318, 2022.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *Advances in neural information processing systems*, 34:13937–13949, 2021.

David Raposo, Sam Ritter, Blake Richards, Timothy Lillicrap, Peter Conway Humphreys, and Adam Santoro. Mixture-of-depths: Dynamically allocating compute in transformer-based language models. *arXiv preprint arXiv:2404.02258*, 2024.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. URL https://arxiv.org/abs/2408.00714.

Chaitanya Ryali, Yuan-Ting Hu, Daniel Bolya, Chen Wei, Haoqi Fan, Po-Yao Huang, Vaibhav Aggarwal, Arkabandhu Chowdhury, Omid Poursaeed, Judy Hoffman, et al. Hiera: A hierarchical vision transformer without the bells-and-whistles. In *International Conference on Machine Learning*, pp. 29441–29454. PMLR, 2023.

Leqi Shen, Guoqiang Gong, Tao He, Yifeng Zhang, Pengzhang Liu, Sicheng Zhao, et al. Fastvid: Dynamic density pruning for fast video large language models. *Advances in Neural Information Processing Systems*, 38:123553–123581, 2026.

Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pp. 402–419. Springer, 2020.

Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *Advances in neural information processing systems*, 35:10078–10093, 2022.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *Proceedings of the IEEE international conference on computer vision*, pp. 3551–3558, 2013.

Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Dense trajectories and motion boundary descriptors for action recognition. *International journal of computer vision*, 103:60–79, 2013.

Limin Wang, Bingkun Huang, Zhiyu Zhao, Zhan Tong, Yinan He, Yi Wang, Yali Wang, and Yu Qiao. Videomae v2: Scaling video masked autoencoders with dual masking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14549–14560, 2023.

Yihan Wang, Lahav Lipson, and Jia Deng. Sea-raft: Simple, efficient, accurate raft for optical flow. In *European Conference on Computer Vision*, pp. 36–54. Springer, 2024.

Thomas Wiegand, Gary J Sullivan, Gisle Bjontegaard, and Ajay Luthra. Overview of the h. 264/avc video coding standard. *IEEE Transactions on circuits and systems for video technology*, 13(7): 560–576, 2003.

Chao-Yuan Wu, Manzil Zaheer, Hexiang Hu, R Manmatha, Alexander J Smola, and Philipp Krähenbühl. Compressed video action recognition. In *CVPR*, 2018.

Chao-Yuan Wu, Yanghao Li, Karttikeya Mangalam, Haoqi Fan, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Memvit: Memory-augmented multiscale vision transformer for efficient long-term video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13587–13597, 2022.

Senqiao Yang, Kaichen Zhang, Zhaoyang Jia, Jinghao Guo, Yifei Shen, Xinjie Zhang, Xiaoyi Zhang, Haoqing Wang, Xiao Li, Peng Zhang, et al. Mage-vl: An efficient codec-native streaming multimodal foundation model. *arXiv preprint arXiv:2607.24904*, 2026a.

Shuo Yang, Haocheng Xi, Yilong Zhao, Muyang Li, Jintao Zhang, Han Cai, Yujun Lin, Xiuyu Li, Chenfeng Xu, Kelly Peng, et al. Sparse videogen2: Accelerate video generation with sparse attention via semantic-aware permutation. *Advances in Neural Information Processing Systems*, 38:96965–96991, 2026b.

Hongxu Yin, Arash Vahdat, Jose M Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. A-vit: Adaptive tokens for efficient vision transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10809–10818, 2022.

Bowen Zhang, Limin Wang, Zhe Wang, Yu Qiao, and Hanli Wang. Real-time action recognition with enhanced motion vector cnns. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2718–2726, 2016.

Qiming Zhang, Yufei Xu, Jing Zhang, and Dacheng Tao. Vsa: Learning varied-size window attention in vision transformers. In *European conference on computer vision*, pp. 466–483. Springer, 2022.

Descriptor	Formula	Geometric interpretation
Jacobian norm	$\|\mathbf{J}_t(\boldsymbol{x})\|_F$	Overall local motion variation, including expansion, rotation, and shear.
Divergence	$\left\|\frac{\partial v_t^x}{\partial x} + \frac{\partial v_t^y}{\partial y}\right\|$	First-order local expansion or contraction.
Curl	$\left\|\frac{\partial v_t^y}{\partial x} - \frac{\partial v_t^x}{\partial y}\right\|$	Local rotational tendency of the motion field.
Shear	$\sqrt{\left(\frac{\partial v_t^x}{\partial x} - \frac{\partial v_t^y}{\partial y}\right)^2 + \left(\frac{\partial v_t^x}{\partial y} + \frac{\partial v_t^y}{\partial x}\right)^2}$	Anisotropic deformation through differential stretching and shearing.
Local area change	$\lvert\det(\mathbf{I} + \mathbf{J}_t(\boldsymbol{x})) - 1\rvert$	Local expansion or contraction of area under the unit-step motion mapping.

Table 8: Motion-field descriptors and their geometric meanings.

A SUMMARY

In the appendix, we provide implementation details and ablations of Jacobian estimation, analyze the geometric relationships among motion-field descriptors, and compare signal fusion strategies in sections B to D. We further evaluate CoMo on egocentric videos, across temporal sampling rates and model sizes, under different selection thresholds, and provide analysis in representation learning in sections E to I. Finally, in sections J to L, we continue to discuss differences from the concurrent work and provide additional visualizations of codec signals and token selection.

B JACOBIAN ESTIMATION AND MOTION-FIELD IMPLEMENTATION

For simplicity, we represent codec motion vectors on a uniform 4×4 grid, matching the smallest motion-compensation partition size in H.264 (Wiegand et al., 2003). Motion vectors from larger partitions are replicated across their constituent grid cells and normalized by the signed temporal distance to their reference frames. For blocks with bidirectional prediction, we use the list-0 vector when available and otherwise fall back to list-1. Locations without an inter-frame reference are marked as invalid. We compute the spatial derivatives of each motion-field component using central differences between adjacent grid cells. The finite differences account for this spatial spacing, keeping displacements and coordinates in consistent units. When only one neighboring sample is valid, we use a one-sided difference; locations without sufficient valid support are excluded. To isolate the effect of the derivative operator, we compare central differences with the Gaussian and Sobel estimators using the Jacobian norm for token ranking under a fixed token budget. As shown in Table 7, central differences achieve the highest Top-1 accuracy and tie with Sobel in Top-5 accuracy. This may be because central differences preserve sharp local motion changes that help identify informative tokens, whereas smoothing can weaken these responses. Given their performance and simplicity, we use central differences in all subsequent experiments.

Estimator	Top-1 ↑	Top-5 ↑
Gaussian derivative	70.30	92.10
Sobel	70.34	**92.17**
Central difference	**70.40**	**92.17**

Table 7: Jacobian estimation ablation.

C GEOMETRIC ANALYSIS OF MOTION-FIELD DESCRIPTORS

Using the Jacobian estimator as described above, we examine how different scalar descriptors characterize local motion variation. Table 8 summarizes their definitions and geometric interpretations. Specifically, the squared Frobenius norm decomposes into contributions from divergence, curl, and

shear:

$$\begin{aligned}\|\mathbf{J}_t(\boldsymbol{x})\|_F^2 &= \left(\frac{\partial v_t^x}{\partial x}\right)^2 + \left(\frac{\partial v_t^y}{\partial y}\right)^2 + \left(\frac{\partial v_t^x}{\partial y}\right)^2 + \left(\frac{\partial v_t^y}{\partial x}\right)^2 \\ &= \frac{1}{2}\underbrace{\left(\frac{\partial v_t^x}{\partial x} + \frac{\partial v_t^y}{\partial y}\right)^2}_{\text{divergence}^2} + \frac{1}{2}\underbrace{\left(\frac{\partial v_t^y}{\partial x} - \frac{\partial v_t^x}{\partial y}\right)^2}_{\text{curl}^2} + \frac{1}{2}\underbrace{\left[\left(\frac{\partial v_t^x}{\partial x} - \frac{\partial v_t^y}{\partial y}\right)^2 + \left(\frac{\partial v_t^x}{\partial y} + \frac{\partial v_t^y}{\partial x}\right)^2\right]}_{\text{shear}^2}.\end{aligned} \tag{11}$$

This decomposition shows that the squared Jacobian norm weights the magnitudes of divergence, curl, and shear equally, allowing motion fields with different geometric properties to receive the same score. However, Table 2 shows that divergence and curl outperform shear, suggesting that these components differ in their usefulness for token selection. This motivates us to explore a scalar descriptor considering how they jointly affect local area, rather than simply summing their magnitudes. We therefore examine the local area-change response and derive its relationship to divergence, curl, and shear by expanding the determinant in the following:

$$\begin{aligned}A_t(\boldsymbol{x}) &= |\det(\mathbf{I} + \mathbf{J}_t(\boldsymbol{x})) - 1| \\ &= \left|\left(1 + \frac{\partial v_t^x}{\partial x}\right)\left(1 + \frac{\partial v_t^y}{\partial y}\right) - \frac{\partial v_t^x}{\partial y}\frac{\partial v_t^y}{\partial x} - 1\right| \\ &= \left|\frac{\partial v_t^x}{\partial x} + \frac{\partial v_t^y}{\partial y} + \frac{\partial v_t^x}{\partial x}\frac{\partial v_t^y}{\partial y} - \frac{\partial v_t^x}{\partial y}\frac{\partial v_t^y}{\partial x}\right| \\ &= \left|\underbrace{\frac{\partial v_t^x}{\partial x} + \frac{\partial v_t^y}{\partial y}}_{\text{signed divergence}} + \frac{1}{4}\underbrace{\left(\frac{\partial v_t^x}{\partial x} + \frac{\partial v_t^y}{\partial y}\right)^2}_{\text{divergence}^2} + \frac{1}{4}\underbrace{\left(\frac{\partial v_t^y}{\partial x} - \frac{\partial v_t^x}{\partial y}\right)^2}_{\text{curl}^2} - \frac{1}{4}\underbrace{\left[\left(\frac{\partial v_t^x}{\partial x} - \frac{\partial v_t^y}{\partial y}\right)^2 + \left(\frac{\partial v_t^x}{\partial y} + \frac{\partial v_t^y}{\partial x}\right)^2\right]}_{\text{shear}^2}\right|.\end{aligned} \tag{12}$$

D Combining Residual and Motion-Geometry Signals

We compare AND and OR fusion on MotionBench using Qwen3-VL$_{8B}$. For each fusion rule, we scale the residual and motion-geometry thresholds by a common factor, keeping their ratio fixed, and select operating points with comparable computational costs. As shown in Table 9, OR achieves slightly higher accuracy at both operating points, supporting its use to preserve tokens that are informative under either signal.

τ_r	τ_a	Fusion	Acc. ↑	TFLOPs ↓
0.15	0.05	AND	56.99	24.64
0.9	0.3	OR	57.07	25.48
0.08	0.027	AND	57.79	32.05
0.6	0.2	OR	58.04	31.85

Table 9: Signal fusion under different thresholds.

E Egocentric Video Understanding

We further evaluate CoMo on EgoSchema using Qwen3-VL$_{8B}$. As shown in Table 10, RLT with its default threshold of 0.1 provides limited computational savings and does not accelerate inference ($0.97\times$). We therefore increase its threshold to 0.4, obtaining a $1.22\times$ speedup at 70.40 accuracy. CoMo achieves higher accuracy (70.60) with 13.2% fewer TFLOPs and a greater speedup of $1.38\times$, demonstrating a better performance on egocentric videos.

Method	Acc. ↑	TFLOPs ↓	Speedup ↑
Qwen3-VL$_{8B}$	72.00	266.64	$1.00\times$
RLT$_{0.1}$	71.05	252.43	$0.97\times$
RLT$_{0.4}$	70.40	192.14	$1.22\times$
CoMo (ours)	**70.60**	**166.71**	$\mathbf{1.38}\times$

Table 10: EgoSchema performance.

F Higher FPS

We further examine whether CoMo remains effective under different temporal sampling rates. Using the same selection thresholds, we vary the sampling rate from 4 to 30 FPS on FPSBench. As

	4 FPS		8 FPS		16 FPS		30 FPS	
Method	Avg.@4	Speedup	Avg.@4	Speedup	Avg.@4	Speedup	Avg.@4	Speedup
Qwen3-VL$_{8B}$	29.90	1.00×	30.60	1.00×	31.65	1.00×	31.52	1.00×
+ CoMo	29.85	1.36×	30.53	1.30×	31.80	1.42×	31.90	1.27×

Table 11: Avg.@4 and speedup of Qwen3-VL$_{8B}$ with CoMo at different sampling rates.

shown in Table 11, CoMo closely matches dense inference at 4 and 8 FPS and slightly improves Avg.@4 at 16 and 30 FPS. Across all four sampling rates, CoMo achieves end-to-end inference speedups of 1.27×–1.42×. These results demonstrate that CoMo remains effective at different temporal sampling rates without threshold retuning.

G Scaling with Model Size

We evaluate CoMo on Qwen3-VL models on MotionBench ranging from 2B to 32B parameters using the same input configuration and selection thresholds. Because token selection is independent of the downstream model, CoMo produces the same token mask for a given video across all model sizes. As shown in Table 12, we find that our method consistently accelerates inference while maintaining accuracy comparable to dense inference, with only minor performance degradation in all model sizes.

Model	Dense	CoMo	Speedup
Qwen3-VL$_{2B}$	55.18	54.28	1.20×
Qwen3-VL$_{4B}$	57.74	55.85	1.34×
Qwen3-VL$_{8B}$	59.01	58.04	1.35×
Qwen3-VL$_{32B}$	61.55	60.65	1.66×

Table 12: Scaling across Qwen3-VL model parameters.

H Threshold Tuning

We analyze the sensitivity of CoMo to its selection thresholds and compare its accuracy–compute trade-off with RLT and random selection. Starting from the default thresholds $(\tau_r, \tau_a) = (0.6, 0.2)$, we jointly scale both values by the same factor, preserving their relative weighting while varying the number of retained tokens. We analogously vary the selection threshold for RLT and match random selection to comparable compute budgets. As shown in Figure 5, all methods approach dense performance when more tokens are retained. Under increasingly constrained budgets, however, CoMo degrades substantially more slowly than RLT and random selection, and its advantage widens as the number of retained tokens decreases. This trend indicates that codec residuals and motion geometry become particularly effective when the token budget is limited, as they prioritize action-relevant content more reliably than RGB differences or random selection.

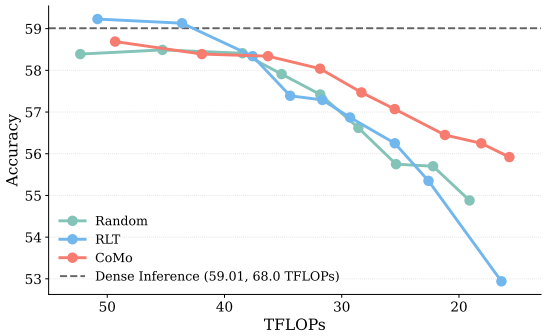


Figure 5: Effect of selection thresholds.

I Representation Analysis

To examine what each selection signal preserves in the learned representation, we visualize V-JEPA 2 (Assran et al., 2025) attentive-probe features on a 600-video SSv2 subset. As shown in Figure 6, we divide the subset into three levels: (a) different verbs, largely separable from appearance; (b) the same verb with a different object or motion direction; and (c) real actions paired with their pretended counterparts, which differ primarily in whether the object is actually affected.

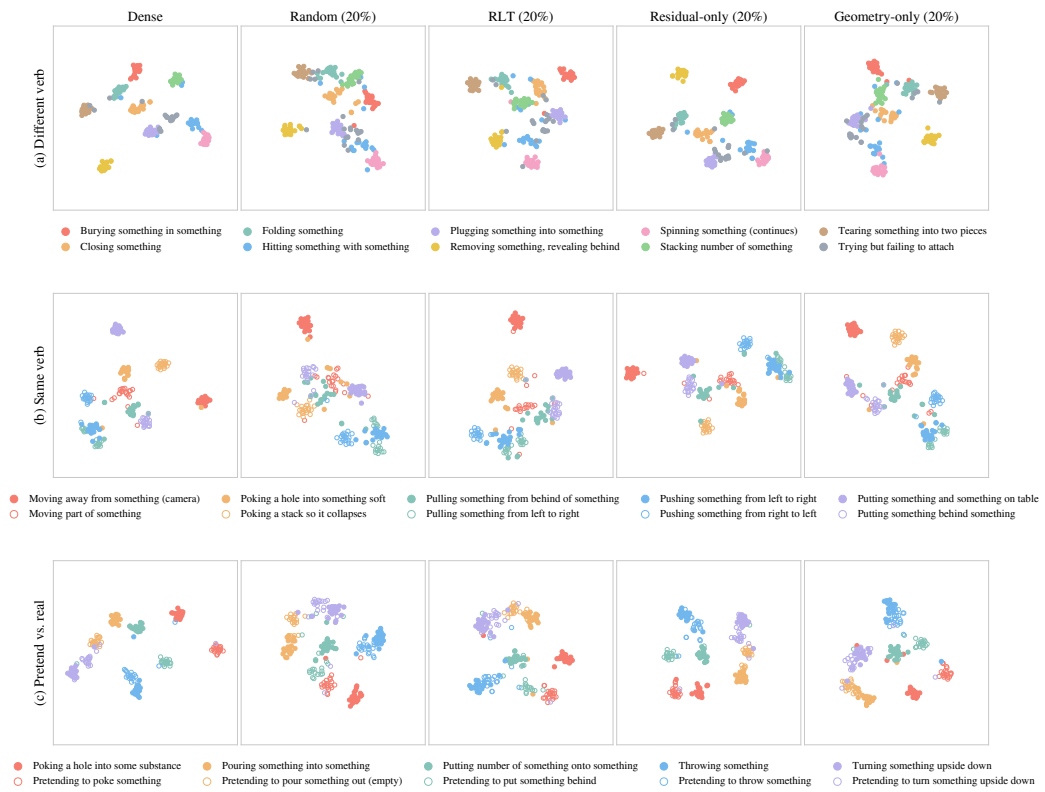


Figure 6: **Representation analysis under 20% token retention.** t-SNE of V-JEPA 2 probe features on SSv2 classes grouped by (a) different verbs, (b) same verb with different object or motion direction, and (c) real vs. pretended actions; paired classes share a hue (filled vs. open) and all panels share one coordinate scale.

Following our main evaluation protocol, we remove tokens before the encoder and compare dense inference with random, RLT, residual-only, and geometry-only selection in a matched budget of 20% retained tokens. Residual selection preserves the appearance-driven structure of levels (a) and (c): its clusters remain compact and well separated. Geometry selection is strongest on level (b), where classes differ in motion direction (e.g., pushing left-to-right vs. right-to-left) that appearance alone cannot resolve. In contrast, random selection inflates every cluster, and RLT collapses the real/pretended pairs, since frame differencing responds to the hand motion shared by both.

J More Visualization

As shown in Figure 7, we provide additional token-selection visualizations across sports and animated scenes. CoMo retains regions around moving players, characters, and their interactions while removing much of the static background.

K Codec Visualization

As shown in Figure 8, codec residuals and motion vectors reveal different aspects of temporal change across real-world and animated scenes. Residual responses are often concentrated around contours, fine textures, and changing facial features, while motion vectors exhibit broader patterns of displacement across subjects and backgrounds. Regions with substantial motion can still have weak residuals when their appearance is well predicted through motion compensation. These examples illustrate why CoMo considers both prediction errors and motion-field geometry when selecting tokens.



Figure 7: Additional qualitative examples of token selection by CoMo across diverse video scenes.

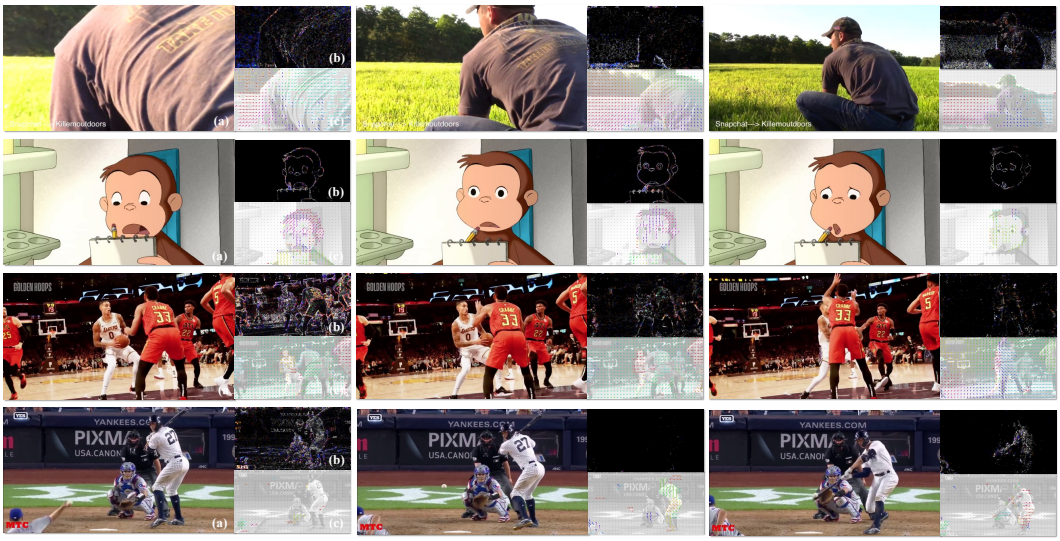


Figure 8: **Visualization of codec residuals and motion vectors across diverse video scenes.** Each row shows a video sequence, with time progressing from left to right. Each group contains (a) an RGB frame, (b) its codec residual, and (c) the corresponding motion vectors.

L CONCURRENT WORK

Recently, Mage-VL (Yang et al., 2026a) has also explored codec signals for efficient video understanding. Mage-VL develops a codec-native model for streaming video understanding through dedicated training, using compressed-bitstream information as complementary of the codec residual and motion vectors to replace part of the dense RGB processing. Our work focuses on training-free inference acceleration for existing video transformers and provides a clear and systematic account of how codec prediction residuals and motion vectors translate into token-selection criteria.