RETHINKING VIDEO TOKEN COMPRESSION WITH A GLOBAL CODEBOOK: LEARNING ONCE, COMPRESSING EVERYWHERE

Anonymous authors
Paper under double-blind review

ABSTRACT

Video large language models (Video-LLMs) represent videos as dense sequences of visual tokens, whose length grows with the temporal and spatial extent of the input. These tokens often contain substantial redundancy arising from repeated visual patterns, leading to unnecessary computation in the subsequent language-model processing. Existing token compression methods, including pruning and merging, perform compression online during inference, repeatedly incurring additional computation for each input video and often relying on model-specific designs that limit their generality, we instead rethink this paradigm by shifting the costly compression process offline. We propose ONCE, a plug-in video token compression framework that introduces an offline-to-online paradigm: a frequency-aware global codebook is learned once in the visual feature space and reused for lightweight online compression through codebook lookup and aggregation, reducing repeated per-video computation and the need for model-specific compression designs. Extensive experiments across multiple video understanding benchmarks and against diverse compression baselines demonstrate that our approach achieves a strong accuracy-efficiency trade-off, maintaining competitive performance while achieving the lowest inference latency among compared methods.

1 INTRODUCTION

Video understanding with large multimodal models produces dense visual-token sequences whose length grows with the temporal and spatial extent of a video. Repeated content across nearby frames and patches makes these sequences redundant, while their length increases language-model prefill cost Lin et al. (2024); Li et al. (2025a); Yang et al. (2025). Effective compression must therefore shorten the sequence without losing evidence from the current video.

Existing methods mainly make selection or aggregation decisions separately for each input after dense visual features have been formed Chen et al. (2024a); Tao et al. (2025); Li et al. (2024b). They reduce the sequence delivered to the language model, but repeat the input-specific grouping step and usually leave visual encoding unchanged. Other methods move merging into the visual encoder, while their merge pattern remains tied to the current input Wang et al. (2026). This leads to the central design question: can partition geometry be learned once within a frozen feature space and reused across videos while the output remains grounded in the current video and the Video-LLM stays unchanged?

We propose **ONCE** (Figure 1), a plug-in framework that learns the reusable partition before evaluation and constructs each shortened sequence from the current video's features at the frozen backbone interface. The offline stage amortizes corpus-level partition fitting, while the online stage instantiates the shared geometry with each video's features and performs the corresponding aggregation.

Our main contributions are summarized as follows:

- **Reusable partition formulation.** We recast video-token compression as learning partition geometry in a fixed visual feature space, separating corpus-level structure from construction of each video's compressed representation.

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.

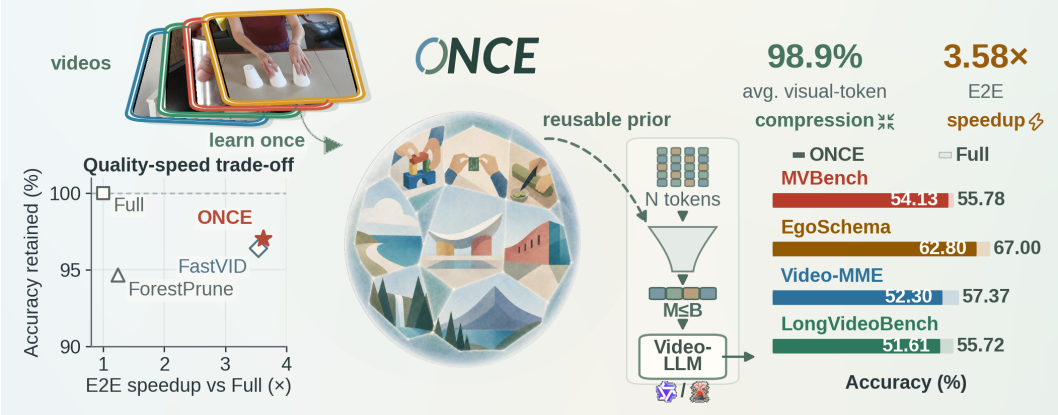


Figure 1: ONCE learns a reusable corpus-level codebook from an offline video corpus. For each test video, codebook-based grouping and continuous source-feature aggregation replace N projected visual tokens with at most B pooled tokens before language-model processing. The quality-speed panel uses the LLaVA-OneVision-7B $B = 256$ operating point; the headline E2E speedup reports the $B = 512$ operating point in Table 3.

- **Offline codebook construction.** We use temporally stratified, source-count-weighted sketches to build a codebook that covers the video timeline while preserving the source-token mass represented by each sketch.
- **Budget-flexible inference interface.** Current-video features select supported groups, complete reassignment covers the source sequence, and source means form continuous inputs to the frozen Video-LLM. A backbone-specific codebook can therefore serve multiple token budgets without Video-LLM fine-tuning or budget-specific refitting.
- **Cross-feature-space evaluation.** We instantiate the separation between reusable grouping and source-feature construction in two distinct frozen feature spaces, and evaluate accuracy and end-to-end efficiency under matched token and runtime protocols.

2 Related Work

2.1 Video Large Language Models

Recent advances in multimodal large language models have enabled Video-LLMs to perform complex video understanding tasks by integrating pretrained visual encoders with large language models. Early Video-LLMs, such as Video-ChatGPT and Video-LLaVA, establish a modular architecture that connects pretrained vision encoders with LLMs through lightweight projection modules Maaz et al. (2024); Lin et al. (2024); Zhang et al. (2023); Li et al. (2025b). Recent works, including LongVILA, LongLLaVA, and LLaVA-OneVision, further scale Video-LLMs toward long-video understanding by extending visual contexts, improving temporal reasoning, and leveraging large-scale video instruction data Chen et al. (2025); Wang et al. (2025b); Li et al. (2025a); An et al. (2025). Meanwhile, general-purpose vision-language models such as InternVL and Qwen3-VL provide stronger multimodal representations and flexible visual token processing, serving as powerful backbones for video understanding tasks Chen et al. (2024b); Wang et al. (2025a); Bai et al. (2025). However, these advances rely on increasingly dense visual-token sequences, where the growing number of visual tokens becomes a major bottleneck for long-video inference.

2.2 Video Token Compression

Training-free methods prune or aggregate tokens using spatial and temporal structure Ju et al. (2026); Guo et al. (2026); Shao et al. (2025), feature redundancy Shen et al. (2025); Tao et al. (2025); Kim et al. (2024); Fu et al. (2025b), query relevance Huang et al. (2025); Shang et al. (2025), or

information-aware criteria Liu et al. (2026). Unified spatiotemporal compression selects from a video-wide token pool and merges unselected evidence, while local–global optimal transport aggregates discarded context into retained anchors Du et al. (2026); Li et al. (2026). UniComp uses informational uniqueness for per-video frame fusion, budget allocation, and spatial compression Yuan et al. (2026). EarlyTom moves merging inside the visual encoder to reduce encoder cost Wang et al. (2026). These methods all adapt their decisions to the current input. ONCE instead amortizes grouping geometry offline with a corpus-fitted codebook; each video uses that geometry to construct continuous output representations from its own features.

Training-integrated methods learn compact video representations, as in LLaMA-VID and VQToken Li et al. (2024b); Zhang & Fu (2025), or adapt the model under compression. DUET-VLM combines visual-output compression, layer-wise text-guided dropping, and end-to-end adaptation Singh et al. (2026). These methods adapt the representation jointly with the compressed model. ONCE complements this line with a frozen-checkpoint interface and a backbone-specific codebook fitted from unlabeled visual features.

Comparison context. The main numerical rows therefore share the backbone, input contract, insertion point, and token budget. Representative source-paper operating points for the other aggregation-oriented methods are summarized in the supplement with their training and interface conventions preserved, providing context for the design space alongside the matched latency comparison.

2.3 Codebook-based Representation Learning

Codebook-based representation learning learns a finite set of representative prototypes to discretize continuous visual features into compact latent spaces. Early approaches such as VQ-VAE and VQ-GAN introduce vector quantization to construct discrete visual representations for image reconstruction and generation van den Oord et al. (2017); Esser et al. (2021). Recent works extend this idea to multimodal models by learning discrete visual vocabularies, where continuous visual features are mapped to compact token representations for efficient visual understanding and generation Zhang & Fu (2025); Zhang et al. (2025a).

ONCE uses prototypes as reusable geometric anchors for grouping, while the sequence passed onward is formed from the current video's continuous features.

3 Method

3.1 ONCE Framework Overview

ONCE operates at the visual-to-language interface of a frozen backbone. For each frozen feature space, its learned artifact is a reusable codebook; the output content is constructed from the input video, and the Video-LLM weights remain unchanged.

Given a video V, the frozen visual encoder E_{vis} and visual projection or merger P_{vis} produce dense visual tokens:

$$X = P_{\text{vis}}(E_{\text{vis}}(V)) \in \mathbb{R}^{N \times D}. \tag{1}$$

ONCE fits a corpus-level codebook $C = [c_1, \ldots, c_K]^{\mathsf{T}}$ offline in this same feature space and returns

$$Z = \text{ONCE}(X, C, B) \in \mathbb{R}^{M \times D}, \qquad M \leq \min(N, B). \tag{2}$$

Here B bounds the number of pooled visual-content tokens, excluding model-specific special and layout tokens, and M is the input-dependent output length. We write $\nu(v) = v/\max(\|v\|_2, \delta)$, with $\delta > 0$, for ℓ_2 normalization in cosine assignments. The backbone interface adds its special or layout tokens separately, so the total visual-token count can exceed B.

The codebook provides a reusable geometric scaffold for content-preserving aggregation. It identifies active groups in the current sequence, and ONCE pools the corresponding continuous embeddings to form output tokens. This combines cross-video reuse of grouping geometry with video-conditioned representation construction; all assignment and aggregation costs are included in end-to-end latency while the shared geometry is amortized.

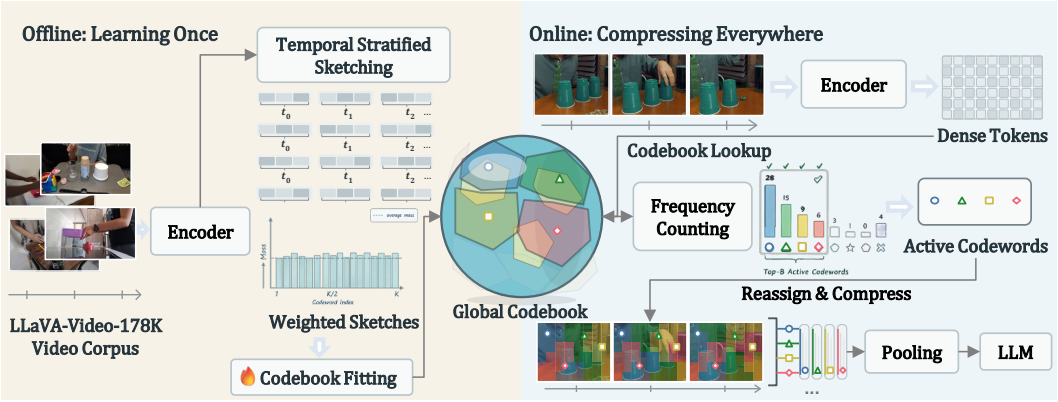


Figure 2: Overview of ONCE. (a) Offline, ONCE fits a corpus-level codebook from temporally stratified, source-count-weighted corpus sketches. (b) At inference, the codebook induces groups supported by the current video, and each output token is obtained by averaging the continuous features assigned to a group.

3.2 Corpus-Level Codebook Learning

ONCE learns a codebook from large-scale video data in two stages: temporally stratified corpus sketching and capacity-regularized codeword fitting.

Temporally Stratified Corpus Sketch For each training video, we divide the normalized timeline into L equal bins, with $\tau = 1$ denoting the final bin. Given N source tokens, the sketch size is $m = \min\{N, \mathrm{clip}(\lceil N/R \rceil, m_{\min}, m_{\max})\}$, where R is the target number of source tokens per sketch token. We distribute m representatives across nonempty bins as evenly as capacity permits, with at least one representative per bin whenever possible.

Within each bin, spherical Lloyd updates Dhillon & Modha (2001) partition the normalized tokens into local groups A_j. Each group stores a representative and the amount of source-token mass it summarizes:

$$r_j = \nu\left(\sum_{i \in A_j} \nu(x_i)\right), \qquad w_j = |A_j|, \qquad \sum_j w_j = N. \tag{3}$$

Aggregating all videos gives weighted representatives $\mathcal{D} = \{(r_u, w_u, g_u)\}_{u=1}^{U}$, where g_u is the source-video category when available. The representative preserves local visual content, while w_u preserves the number of source tokens it summarizes.

The temporal labels distribute representatives across the offline cache, providing coverage across the normalized timeline while preserving feature geometry and source-token mass. They guide codebook construction and remain distinct from inference-time support selection.

Capacity-Regularized Codebook Fitting Naive clustering can allocate many prototypes to frequent visual patterns. We therefore fit K normalized codewords with weighted, capacity-regularized assignments. For representative r_u, its effective weight is

$$\omega_u = w_u p_{g_u}^{-\alpha}, \tag{4}$$

where p_{g_u} is the frequency of its source category and α controls long-tail correction. When category labels are unavailable, $\alpha = 0$. The source-count factor preserves the contribution of high-mass representatives while the correction limits domination by frequent categories.

For a fitting batch, let $s_{uk} = \nu(r_u)^\top \nu(c_k)$ be the cosine similarity and let $a_u = \omega_u / \sum_v \omega_v$ and $b_k = 1/K$ be the desired row and column marginals, with $S = (s_{uk})$. The capacity constraint is

imposed on assignment mass:

$$\Gamma^* = \arg\max_{\Gamma \geq 0} \langle \Gamma, S \rangle + \varepsilon H(\Gamma), \quad \Gamma \mathbf{1} = a, \qquad \Gamma^\top \mathbf{1} = b, \qquad H(\Gamma) = -\sum_{u,k} \Gamma_{uk} \log \Gamma_{uk}. \tag{5}$$

Thus, the uniform target spreads soft assignment mass across the fitting batch, while hard occupancy follows the learned feature geometry. Sinkhorn is a matrix- scaling algorithm for solving entropy-regularized optimal-transport problems. We use it to solve the capacity-constrained assignment efficiently during global codebook fitting, preserving the prescribed assignment mass while allowing feature geometry to determine hard occupancy. A straight-through Gumbel–Softmax parameterization Jang et al. (2017) completes codeword fitting, and the objective combines weighted reconstruction, commitment, and usage regularization. The resulting codebook defines reusable grouping geometry for each frozen feature space, while the input video supplies the continuous features used for downstream aggregation. It is held fixed during inference and supports video-conditioned assignment and aggregation.

3.3 Inference-Time Token Compression

For a new video, the constructed codebook is applied to its feature sequence through the following assignment and aggregation steps.

Active Codeword Selection Each source token is first assigned to its nearest codeword:

$$\widehat{k}_i = \arg\max_{k \in [K]} \nu(x_i)^\top \nu(c_k), \qquad n_k = \sum_i \mathbf{1}[\widehat{k}_i = k]. \tag{6}$$

The count n_k measures how much of the current video is associated with codeword k. ONCE retains the nonzero codewords with the largest counts, forming $\mathcal{K}_B$ with $|\mathcal{K}_B| \leq \min(B, K)$. This step instantiates the global geometry as a video-conditioned support set under the chosen budget.

Complete Reassignment and Token Pooling After inactive codewords are removed, every source token is reassigned to the retained set:

$$k_i = \arg\max_{k \in \mathcal{K}_B} \nu(x_i)^\top \nu(c_k), \qquad z_k = \frac{1}{|\mathcal{G}_k|} \sum_{i \in \mathcal{G}_k} x_i, \tag{7}$$

where $\mathcal{G}_k$ contains the source tokens assigned to active codeword k. The second assignment ensures complete source coverage. Mean pooling aggregates the original continuous embeddings, making each output token a content-specific summary of the current video. Output tokens are emitted in a deterministic active-group order, and the partition property gives $M \leq \min(N, B)$. In the whole-video path, active-group order provides a deterministic sequence over jointly supported groups, and each group is represented by its current-video source features. The backbone adapter then places the shortened visual sequence according to its native visual layout, keeping the grouping rule compatible with backbone-specific positional interfaces.

Inference complexity. The offline stage fits the codebook once for each frozen feature space; let T_{off} denote this one-time cost. It is excluded from per-video online inference. After X is available, the online compression stage for a video with N source tokens of dimension D and K codebook entries performs an $O(NKD)$ lookup, adds $O(N + K \log B)$ for counting and support selection, and costs $O(N|\mathcal{K}_B|D)$ for reassignment plus $O(ND)$ for pooling, where $|\mathcal{K}_B| \leq B$. Hence,

$$C_{\text{on}} = O(NKD + N|\mathcal{K}_B|D + ND + N + K \log B)\,. \tag{8}$$

The matched end-to-end profile includes visual encoding, this online stage, language-model prefill, and decoding, but excludes T_{off}. For V downstream videos, T_{off}/V can be reported separately as an amortized construction cost; prefill FLOPs isolate language-model computation after compression.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Models, data, and baselines. We use Dense for uncompressed inference. In the Qwen table, Standard denotes the same uncompressed reference; Full denotes the complete ONCE configuration in ablations and supplementary tables.

For the primary evaluation, we select a four-benchmark core from publicly available video-understanding suites using a consistent evaluation interface and complementary coverage of short-form temporal reasoning, long-video comprehension, and broad multimodal video understanding. We evaluate ONCE with the frozen LLaVA-OneVision-7B backbone Li et al. (2025a) on MVBench Li et al. (2024a), EgoSchema Mangalam et al. (2023), Video-MME Fu et al. (2025a), and LongVideoBench Wu et al. (2024). We also instantiate ONCE with Qwen3.5-9B Qwen Team (2026) on MVBench and LongVideoBench, fitting a separate codebook for that backbone-specific visual feature space. These backbones have comparable parameter scale but distinct visual-language architectures, so this pairing tests whether the same separation between reusable grouping and source-feature construction can be instantiated across feature spaces without making model capacity the main experimental variable. The codebooks are fitted independently; the experiment tests the plug-in design across architectures rather than codebook transfer or a capacity and throughput comparison against a smaller VLM.

For LLaVA-OneVision-7B, we compare ONCE with two representative recent video-token compression methods, ForestPrune Ju et al. (2026) and FastVID Shen et al. (2025), using the same frozen backbone, post-encoder visual-to-language interface, and token budgets; the Video-LLM remains frozen for these matched rows. ForestPrune models spatial and temporal token structure, while FastVID uses dynamic density-based pruning. The direct table therefore contains only these matched rows. Related aggregation and representation methods, including FrameFusion, DyCoke, HoliTom, UniComp, and VQToken, are compared in the supplement by their principal mechanisms and source evaluation protocols. The latency comparison below uses this matched frozen-backbone contract. At a smaller LLaVA-OneVision scale, the supplement compares ONCE with VQToken under extreme token compression: ONCE maintains comparable task performance while achieving higher reported end-to-end speedups.

Protocols and metrics. We evaluate video understanding performance under different token budgets B by varying the upper bound on pooled visual-content tokens retained by ONCE. The realized number M can be smaller because empty codeword groups are removed, and the total visual-token count additionally includes any model-specific special or layout tokens.

For efficiency evaluation, the main text reports the selected LLaVA-OneVision MVBench operating point at $B = 512$, with batch size 1 on a single NVIDIA GPU. The full budget sweep is given in the supplement. Each compressed run is paired with uncompressed inference under the same selected evaluation contract. End-to-end latency includes visual encoding, codebook-based grouping and feature aggregation, language-model prefill, and decoding. Prefill speedup isolates the language-model computation after compression and therefore excludes visual encoding, grouping, feature aggregation, and decoding.

Implementation details. For each frozen backbone, we fit a $K = 8192$ corpus-level codebook on the 0–30 s subset of LLaVA-Video-178K Zhang et al. (2025b). The codebook produced under this fitting configuration is reused across all token budgets and downstream datasets. The fitting set contains 94,581 successfully encoded videos and uses only frozen visual features, not downstream questions, answers, labels, or predictions. The fitting subset is disjoint from the videos in MVBench, EgoSchema, Video-MME, and LongVideoBench; a complete identity audit found no direct matches. Online assignment uses cosine similarity, and no Video-LLM parameters are updated. The primary runs use a fixed precision and sampling contract for each backbone; complete runtime settings are given in the supplement.

For LLaVA and Qwen, codeword selection follows the same whole-video rule. The backbone-specific temporal-bin configurations are detailed in the supplement.

Table 1: LLaVA-OneVision-7B accuracy (%) across visual-token budgets under the matched frozen-checkpoint contract. B denotes the pooled visual-content token budget; bold marks the best compressed result within each budget.

Token budget	Method	MVBench	EgoSchema	Video-MME	LongVideoBench
23329	Dense	55.78	67.00	57.37	55.72
32	ForestPrune	44.85	47.80	45.93	46.15
	FastVID	40.67	39.60	43.85	43.08
	ONCE (Ours)	**47.93**	**55.20**	**48.44**	**47.49**
64	ForestPrune	47.67	52.80	47.96	46.45
	FastVID	45.06	49.20	46.41	44.95
	ONCE (Ours)	**50.58**	**57.00**	**50.30**	**47.64**
128	ForestPrune	49.33	54.40	50.33	47.94
	FastVID	49.32	54.80	49.44	47.42
	ONCE (Ours)	**52.20**	**60.40**	**51.07**	**49.89**
256	ForestPrune	52.80	57.20	52.22	50.26
	FastVID	53.78	60.40	**53.67**	**52.36**
	ONCE (Ours)	**54.13**	**62.80**	52.30	51.61
512	ForestPrune	55.75	61.20	53.56	53.55
	FastVID	**56.45**	59.80	**56.56**	**54.75**
	ONCE (Ours)	54.50	**62.40**	53.26	52.95
1024	ForestPrune	**57.83**	62.40	56.15	**56.62**
	FastVID	57.65	59.40	**57.26**	56.17
	ONCE (Ours)	54.28	**62.60**	53.85	54.00

4.2 Main Results

LLaVA-OneVision-7B. At $B \in \{32, 64, 128\}$, ONCE attains the highest accuracy among compressed methods on all four benchmarks (Table 1). This pattern is consistent with the reusable grouping prior being most useful when token capacity is scarce. As B grows, pruning baselines close or reverse the gap; the result is therefore low-budget robustness rather than uniform dominance.

Qwen3.5-9B. We instantiate the same design with a Qwen-specific codebook, with $B = 512$ giving the best scores on both benchmarks (Table 2). The trend suggests that downstream performance under codebook compression need not improve linearly with the retained-token budget, so B is selected empirically rather than treated as a direct proxy for representation quality. The codebook constructed for this Qwen profile is reused across all four budgets, so the trend reflects the budget choice rather than budget-specific retraining.

Table 2: Qwen3.5-9B accuracy (%) across visual-token budgets. Standard is the uncompressed reference.

Token budget	Method	MV	LVB
Standard	Qwen3.5-9B	70.18	65.37
128	ONCE	56.23	51.46
256	ONCE	58.90	53.78
512	ONCE	**63.25**	**56.40**
1024	ONCE	61.95	54.43

Efficiency. At the selected $B = 512$ operating point, ONCE has the lowest reported latency among the compared LLaVA methods in the established MVBench profile (Table 3). Its inference procedure uses the same constructed-codebook grouping and source-feature aggregation, so end-to-end speedup depends on the share of language-model prefill in the workload. Memory measurements are provided in the supplement.

Table 3: Selected LLaVA-OneVision-7B MVBench efficiency profile at $B = 512$. Latency is in milliseconds; E2E and Prefill are speedups relative to Dense.

Method	Budget	Lat. (ms)	E2E	Prefill
Dense	23329	3653.79	1.00×	1.00×
ForestPrune	512	2962.96	1.23×	2.78×
FastVID	512	1051.65	3.47×	33.57×
ONCE	512	1019.36	3.58×	36.46×

4.3 Ablation Study

Table 4 separates the roles of temporal partitioning, reassignment, grouping, and output representation. All inference-side variants reuse the codebook constructed for Full.

Table 4: LLaVA-OneVision-7B ablations at $B = 512$. Full is the complete ONCE configuration; Random grouping preserves its group sizes and randomizes membership.

Variant	Changed component	MV	LVB
Full (Table 1)	–	54.50	52.95
Single-bin	Partition	39.03	41.96
w/o complete reassignment	Reassignment	53.83	53.48
Random grouping	Grouping	45.33	46.00
Codeword output	Representation	38.38	41.74

Several apparently destructive variants remain partially effective because source-feature mean pooling preserves continuous current-video content at the LLM interface. Single-bin training still pools source features but loses temporal coverage in the offline sketch; Codeword output keeps learned grouping geometry but replaces these means with codewords, explaining its larger drop. Random grouping preserves group sizes and source means, so its decrease isolates membership geometry. Removing complete reassignment retains support selection and pooling and mainly changes source-token coverage, hence its smaller change. Source-count weighting and long-tail correction only shape offline codebook capacity and add no online module.

4.4 Codebook Analysis

Beyond downstream accuracy, we examine whether the offline-learned codebook captures reusable visual grouping structure in the feature space rather than merely storing training examples. Table 5 summarizes utilization on the training corpus using occupancy, entropy-effective capacity, and mean cosine error. These measures distinguish a codebook that is merely nonempty from one whose capacity is used across the corpus.

Table 5: Full-codebook utilization on the Stage-A cache ($K = 8192$). The 95% confidence intervals (CIs) use video-level bootstrap resampling.

Training-corpus diagnostic	Estimate	95% CI
Active codes (%)	99.93	–
Entropy-effective capacity (%)	53.39	[53.25, 53.42]
Mean cosine error ($\times 10^{-3}$)	4.763	[4.750, 4.775]

The table indicates broad occupancy but nonuniform effective use. Figure 3 localizes this imbalance by ordering codewords by assigned source-token mass and aggregating them into equal-count deciles. The emphasized bars isolate the highest-ranked fifth, while the dashed line provides the uniform reference.

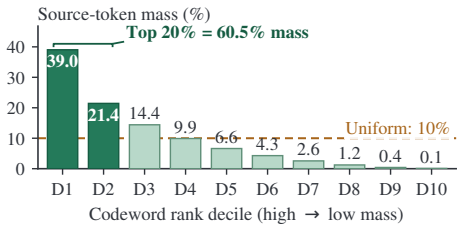
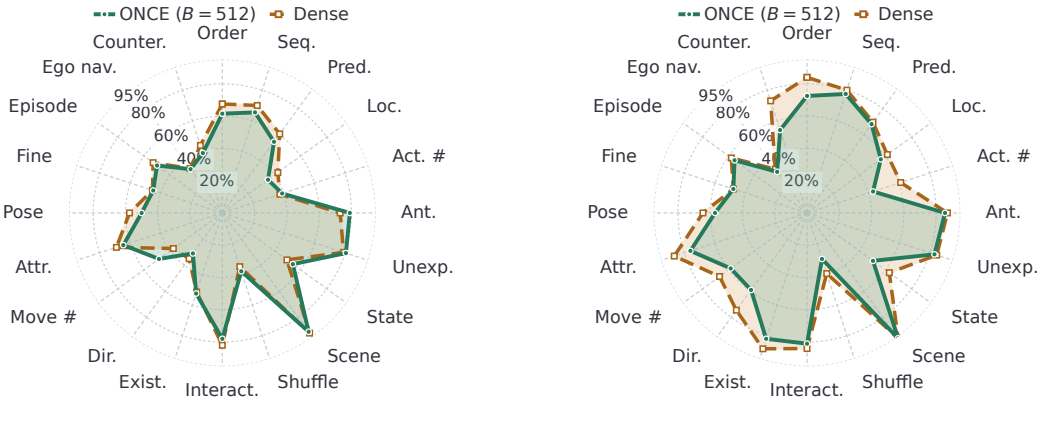


Figure 3: Training-corpus utilization of the selected Full codebook. Codewords are ranked by source-token mass; the bars show decile mass and the line shows cumulative mass.

Task profile. Figure 4 compares the matched Dense and ONCE profiles for LLaVA and Qwen. The profiles show partial consistency: Scene Transition is high in both dense runs, whereas Egocentric Navigation and Object Shuffle are among the lower-scoring tasks in both panels. The profile correlations are 0.762 for Dense and 0.855 for ONCE, suggesting shared task demands across the two feature spaces. Coarse scene cues are comparatively robust, while object shuffles and egocentric navigation require finer alignment; source-mean pooling can blur localization, ordering, and pose cues.

For LLaVA, gains concentrate on Moving Count, Action Antonym, State Change, and Object Shuffle, while Action Localization and Fine-grained Pose decline. Qwen shares weaknesses on Character Order, Action Localization, and Fine-grained Pose but changes in the counting and state-change tasks differ. The common profile therefore reflects a task-level trade-off whose magnitude depends on the feature space, visual layout, and fitted codebook.



(a) LLaVA-OneVision-7B

(b) Qwen3.5-9B

Figure 4: Per-task MVBench accuracy for matched Dense and ONCE runs across the two backbone-specific profiles.

Overall experimental interpretation. Across budgets and backbones, the results support reusable corpus-level geometry with source-derived current-video outputs and task-dependent retention.

5 Conclusion

ONCE separates reusable grouping geometry from current-video representation construction at a frozen Video-LLM interface. Temporal stratification, source-sequence reassignment, and backbone-specific layout preserve the temporal and spatial support of pooled features; active-group ordering organizes the compressed support without replacing the video timeline. Larger or hierarchical codebooks and learned geometric models remain to be further explored for broader feature spaces and finer-grained spatial or temporal tasks.

AI USE STATEMENT

In this work, generative AI tools were used to support parts of the research planning and execution workflow, to draft portions of the manuscript, and to edit the manuscript for clarity and organization. They were not used to generate synthetic data, formulate or prove mathematical claims, or supply experimental results. The authors independently verified each AI-assisted contribution against the source code, experiment records, numerical results, citations, and final manuscript, and take responsibility for the final content of this submission.

REPRODUCIBILITY STATEMENT

Reproducibility details are provided in the method and experimental protocol and in the supplementary material, including codebook construction, inference contracts, evaluation settings, and ablation results. The anonymous code package provides interface-level scaffolds and contract checks for the LLaVA-OneVision-7B and Qwen3.5-9B codebook workflows. Model checkpoints, benchmark media, and model-specific fitting and Video-LLM adapter implementations are not redistributed.

REFERENCES

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, 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, 2025. URL https://arxiv.org/abs/2511.21631.

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

Yukang Chen, Fuzhao Xue, Dacheng Li, Qinghao Hu, Ligeng Zhu, Xiuyu Li, Yunhao Fang, Haotian Tang, Shang Yang, Zhijian Liu, et al. LongVILA: Scaling long-context visual language models for long videos. In *International Conference on Learning Representations*, volume 2025, pp. 18227–18246, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/2e163450c1ae3167832971e6da29f38d-Abstract-Conference.html.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. InternVL: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24185–24198, 2024b.

Inderjit S Dhillon and Dharmendra S Modha. Concept decompositions for large sparse text data using clustering. *Machine learning*, 42(1):143–175, 2001.

Junhao Du, Jialong Xue, Anqi Li, Jincheng Dai, and Guo Lu. Unified Spatiotemporal Token Compression for Video-LLMs at Ultra-Low Retention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17661–17671, 2026.

Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming Transformers for High-Resolution Image Synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12873–12883, 2021.

Chaoyou Fu, Yuhan Dai, Yongdong 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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24108–24118, 2025a.

Tianyu Fu, Tengxuan Liu, Qinghao Han, Guohao Dai, Shengen Yan, Huazhong Yang, Xuefei Ning, and Yu Wang. FrameFusion: Combining similarity and importance for video token reduction

on large vision language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22654–22663, 2025b.

Yansong Guo, Chaoyang Zhu, Jiayi Ji, Jianghang Lin, and Liujuan Cao. HieraVid: Hierarchical token pruning for fast video large language models, 2026.

Xiaohu Huang, Hao Zhou, and Kai Han. PruneVid: Visual token pruning for efficient video large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 19959–19973, Vienna, Austria, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.findings-acl.1024.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *International conference on learning representations*, 2017.

Shaobo Ju, Baiyang Song, Tao Chen, Jiapeng Zhang, Qiong Wu, Chao Chang, HuaiXi Wang, Yiyi Zhou, and Rongrong Ji. ForestPrune: High-ratio visual token compression for video multimodal large language models via spatial-temporal forest modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Findings*, pp. 8326–8336, 2026.

Minchul Kim, Shangqian Gao, Yen-Chang Hsu, Yilin Shen, and Hongxia Jin. Token fusion: Bridging the gap between token pruning and token merging. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1383–1392, 2024.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. LLaVA-OneVision: Easy visual task transfer. *Transactions on Machine Learning Research*, 2025a. URL https://openreview.net/forum?id=zKv8qULV6n.

Jinlong Li, Liyuan Jiang, Haonan Zhang, and Nicu Sebe. Token Reduction via Local and Global Contexts Optimization for Efficient Video Large Language Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10451–10461, 2026.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. MVBench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22195–22206, 2024a.

KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. VideoChat: Chat-centric video understanding. *Science China Information Sciences*, 68(10):200102, 2025b.

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

Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. Video-LLaVA: Learning united visual representation by alignment before projection. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 5971–5984, 2024.

Xinxin Liu, Shiwei Gan, Xiao Liu, Yafeng Yin, Lei Xie, and Sanglu Lu. Infomerge: Information-aware token compression for efficient video large language models. *arXiv preprint arXiv:2606.02161*, 2026.

Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Khan. Video-ChatGPT: Towards detailed video understanding via large vision and language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12585–12602, Bangkok, Thailand, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.679. URL https://aclanthology.org/2024.acl-long.679/.

Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. EgoSchema: A diagnostic benchmark for very long-form video language understanding. *Advances in Neural Information Processing Systems*, 36:46212–46244, 2023.

Qwen Team. Qwen3.5-9B. Hugging Face model card, 2026. Available at https://huggingface.co/Qwen/Qwen3.5-9B (accessed July 29, 2026).

Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. LLaVA-PruMerge: Adaptive token reduction for efficient large multimodal models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22857–22867, 2025.

Kele Shao, Keda Tao, Can Qin, Haoxuan You, Yang Sui, and Huan Wang. HoliTom: Holistic token merging for fast video large language models, 2025.

Leqi Shen, Guoqiang Gong, Tao He, Yifeng Zhang, Pengzhang Liu, Sicheng Zhao, and Guiguang Ding. FastVID: Dynamic density pruning for fast video large language models. In *Advances in Neural Information Processing Systems*, volume 38, pp. 123553–123581. Curran Associates, Inc., 2025.

Aditya Kumar Singh, Hitesh Kandala, Pratik Prabhanjan Brahma, Zicheng Liu, and Emad Barsoum. DUET-VLM: Dual stage unified efficient token reduction for vlm training and inference. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17651–17660, 2026.

Keda Tao, Can Qin, Haoxuan You, Yang Sui, and Huan Wang. DyCoke: Dynamic compression of tokens for fast video large language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 18992–19001, 2025.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, volume 30, pp. 6306–6315. Curran Associates, Inc., 2017.

Hesong Wang, Xin Jin, Lu Lu, Chenhaowen Li, Jian Chen, Qiang Liu, and Huan Wang. EarlyTom: Early token compression completes fast video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 40559–40568, 2026.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. InternVL3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency, 2025a. URL https://arxiv.org/abs/2508.18265.

Xidong Wang, Dingjie Song, Shunian Chen, Junying Chen, Zhenyang Cai, Chen Zhang, Lichao Sun, and Benyou Wang. LongLLaVA: Scaling multi-modal LLMs to 1000 images efficiently via a hybrid architecture. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 21419–21436, Suzhou, China, 2025b. Association for Computational Linguistics. doi: 10.18653/v1/2025.findings-emnlp.1168. URL https://aclanthology.org/2025.findings-emnlp.1168/.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. LongVideoBench: A benchmark for long-context interleaved video-language understanding. *Advances in Neural Information Processing Systems*, 37:28828–28857, 2024.

Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. VisionZip: Longer is better but not necessary in vision language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19792–19802, 2025.

Chao Yuan, Shimin Chen, Minliang Lin, Limeng Qiao, Guanglu Wan, and Lin Ma. UniComp: Rethinking video compression through informational uniqueness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18609–18618, 2026.

Haichao Zhang and Yun Fu. VQToken: Neural discrete token representation learning for extreme token reduction in video large language models. In *Advances in Neural Information Processing Systems*, volume 38, pp. 32851–32869, 2025.

Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. In *Proceedings of the 2023 conference on empirical methods in natural language processing: system demonstrations*, pp. 543–553, 2023.

Wanpeng Zhang, Yicheng Feng, Hao Luo, Yijiang Li, Zihao Yue, Sipeng Zheng, and Zongqing Lu. Unified Multimodal Understanding via Byte-Pair Visual Encoding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12976–12986, 2025a.

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. LLaVA-Video: Video instruction tuning with synthetic data. *Transactions on Machine Learning Research*, 2025b. URL https://openreview.net/forum?id=EElFGvt39K.