

One Token per Highly Selective Frame: Towards Extreme Compression for Long Video Understanding

Supplementary Material

A Implementation Details

A.1 XComp Architecture

XComp is based on VideoChat-Flash [15] and uniquely integrates our learnable progressive compression (**LP-Comp**) (Sec. 3.2) and question-conditioned compression (**QC-Comp**) (Sec. 3.3). XComp comes from fine-tuning the VideoChat-Flash-2B model [15] with a small amount of 2.5% data, the fine-tuning details are shown in Appendix A.2.

Algorithm A shows the neural network architecture of XComp, which is largely consistent with VideoChat-Flash. It begins with the UMT-L visual encoder [17], which encodes short video clips consisting of 8 frames into visual tokens. Token merging [3] is subsequently applied to reduce the token count to 128 tokens per clip and 16 tokens per frame. A two-layer MLP connector then maps these visual tokens from the visual encoder space into the representation space of the large language model. Finally, Qwen2-1.5B [2] serves as the large language model (LLM) in our framework.

During inference with QC-Comp, multiple forward passes through the model are performed. First, XComp conducts a forward pass to obtain scores for the frames. Note that for videos longer than 512 frames, the video is divided into shorter segments, each processed separately to compute frame scores. These individual scores are then aggregated. Based on the aggregated scores, XComp selects the top-ranked frames and passes them to the model to generate responses.

Details of LP-Comp Algorithm B shows the details of LP-Comp (Sec. 3.2). It compresses video tokens layer-by-layer in a suffix-preserving manner. Given input tokens $V^{(\ell)} \in \mathbb{R}^{K \times N^{(\ell)} \times d}$ at layer ℓ , the algorithm computes the target token count N_{next} for the next layer using a cosine-based schedule. If no reduction is needed, tokens are returned unchanged. Otherwise, for each video clip and each frame within it, only the last N_{next} tokens are kept from the current N_{prev} , preserving the temporal suffix. The retained tokens across all frames and clips are concatenated to form the output $V^{(\ell+1)}$. This design ensures progressive compression while retaining semantically rich information.

Details of QC-Comp Algorithm C shows the details of Segmented Local Attention, which is the main part of QC-Comp (Sec. 3.3) and computes scores from attention maps in LLM layers. For a

Algorithm A: MODEL_FORWARD: forward pass with LP-Comp and the score for QC-Comp

Input: \mathcal{M} (model), V (video), Q (text), $\text{returnScore} \in \{0, 1\}$

Output: logits ; score (per-clip) if $\text{returnScore} = 1$

/ Step 1: video encoding */*

$\{V_1^{\text{clip}}, \dots, V_K^{\text{clip}}\} \leftarrow \text{Partition}(V, 8)$ // separate each 8 frames \rightarrow 1 clip

foreach V_k^{clip} **do**

$T_k \leftarrow \mathcal{M}.\text{umt_visual_enc}(V_k^{\text{clip}})$

$T_k \leftarrow \text{TokenMerge}(T_k, 128)$ // to 128 tokens/clip, 16 tokens/frame

$T_k \leftarrow \mathcal{M}.\text{mlp}(T_k)$ // project to LLM dim

$V^{(0)} \leftarrow \text{concat}(T_1, \dots, T_K)$

$Q^{(0)} \leftarrow \mathcal{M}.\text{text_embed}(Q)$

/ Step 2: reasoning in LLM layers */*

for $\ell = 0$ **to** $\mathcal{M}.L - 1$ **do**

$[V^{(\ell+1)}, Q^{(\ell+1)}] \leftarrow \mathcal{M}.\text{llm_layer}_\ell(V^{(\ell)}, Q^{(\ell)})$

$V^{(\ell+1)} \leftarrow \text{LP_Comp}(\ell, V^{(\ell+1)})$ // token-level compression

$\text{score}[\ell] \leftarrow \text{SegmentedLocalAttention}(\ell, V^{(\ell)}, Q^{(\ell)})$ // score for QC-Comp

$\text{logits} \leftarrow \text{ComputeLogits}(Q^{(\mathcal{M}.L)})$

if returnScore **then return** (logits , score)

else return logits

given video V and text query Q , the algorithm slides a local window (64 frames with a stride of 32) over the video sequence. Within each segment, it computes the full attention map from the LLM’s ℓ -th layer. For each frame in the segment, the attention weights over the query tokens are averaged and accumulated into corresponding 8-frame clip buckets. Although Segmented Local Attention inherently produces clip-level scores, QC-Comp assigns the same score to all frames within a clip, thereby converting clip-level scores to frame-level scores. For long videos exceeding 512 frames, the method processes multiple overlapping 512-frame chunks. The final frame-level scores are obtained by aggregating the results from these overlapping chunks. To ensure diverse frame-level scores, each frame would be included in n_{repeat} overlapping and shifted chunks that encourage score variation across neighboring frames. With these frame-level scores, $n_{\text{selected_frames}}$ among total frames are selected. Table B shows the hyperparameters.

A.2 Supervised Fine-tuning Details

Hyperparameters Table A shows the hyperparameters used in fine-tuning. We follow the same training configuration as VideoChat-Flash [15], including the learning rate, weight decay, warmup ratio, and learning rate scheduler. The only difference lies in the frame sampling parameters: frames_upbound, frames_lowbound, and the default frames per second (FPS). This change is due to our use of LP-Comp, which enables more efficient frame representation. As a result, we double the values of frames_lowbound and frames_upbound compared to VideoChat-Flash.

Datasets We used the 2.5% supervised fine-tuning data collected or released by VideoChat-Flash [15]. During data curation, we disregarded a few datasets that were exceptionally

Algorithm B: LP_COMP: suffix-preserving layer-wise video-token compression

Input: ℓ (layer index), $V^{(\ell)}$ (video tokens, shape $K \times N^{(\ell)} \times d$)
Output: $V^{(\ell+1)}$ (compressed video tokens)
 $N_{\text{prev}} \leftarrow \left\lfloor \frac{N^{(1)}-1}{2} \cos\left(\frac{\ell}{L}\pi\right) + \frac{N^{(1)}+1}{2} \right\rfloor$
 $N_{\text{next}} \leftarrow \left\lfloor \frac{N^{(1)}-1}{2} \cos\left(\frac{\ell+1}{L}\pi\right) + \frac{N^{(1)}+1}{2} \right\rfloor$ // Eq. (1)
if $N_{\text{prev}} = N_{\text{next}}$ **then return** $V^{(\ell)}$ // nothing to compress
foreach $T \in V^{(\ell)}$ **do** // iterate over K clips
 foreach $f \leftarrow 1$ **to** F **do** // iterate over F frames in the clip
 $\text{idx_keep} \leftarrow \left[(f-1)N_{\text{prev}} + N_{\text{prev}} - N_{\text{next}}, \dots, fN_{\text{prev}} - 1 \right]$
 $T'_f \leftarrow T[\text{idx_keep}]$ // suffix-preservation
 $T' \leftarrow \text{concat}(T'_1, \dots, T'_F)$
 $V^{(\ell+1)} \leftarrow \text{concat}(T'_{\text{clip}1}, \dots, T'_{\text{clip}K})$
return $V^{(\ell+1)}$

Algorithm C: SEGMENTEDLOCALATTENTION: compute clip-level saliency scores

Input: ℓ (layer index), V (video tokens), Q (text tokens)
Output: score (list of length K , one mean score per clip)
 $L_{\text{seg}} \leftarrow 64$ // frames per segment (8 clips)
 $\text{stride} \leftarrow 32$ // stride in frames (4 clips)
 $\text{bucket} \leftarrow []_K = 1^K$
for $\text{start} = 0$ **to** $F - L_{\text{seg}}$ **step** stride **do** // slide local window over the video
 $V_{\text{seg}} \leftarrow V[\text{start} : \text{start} + L_{\text{seg}}]$
 $A \leftarrow \mathcal{M}.\text{llm_layer}_\ell.\text{attn}([V_{\text{seg}}, Q])$ // full attention map
 for $i = 0$ **to** $L_{\text{seg}} - 1$ **do**
 $c \leftarrow \left\lfloor \frac{\text{start}+i}{8} \right\rfloor$ // clip index
 $w \leftarrow \text{mean}(A[i, Q_{\text{start}} : Q_{\text{end}}])$
 $\text{bucket}[c].\text{append}(w)$
for $k = 1$ **to** K **do**
 $\text{score}[k] \leftarrow \text{mean}(\text{bucket}[k])$ // final clip score
return score

Table A: The hyperparameters used in fine-tuning.

Hyperparameter	Value / Description
tunable_parts	Large Language Model
learning_rate	1×10^{-5}
weight_decay	0.0
warmup_ratio	0.03
lr_scheduler_type	Cosine
dataloader_num_workers	1
frames_upbound	1024
frames_lowbound	128
local_num_frames	8
sample_type	Dynamic FPS (8 fps by default)

Table B: The hyperparameters used in evaluation.

Hyperparameter	Value / Description
temperature	0.0
do_sample	False
num_beams	1
n_repeat	2 (each frame in 2 chunks)
n_selected_frames	256 (Long), 512 (MME), 1,024 (MLVU), 2,048 (LVB)

Table C: Datasets used for supervised fine-tuning (71,927 instances).

Image datasets	Video datasets
LLaVA-OneVision [13], LLaVA-NeXT [16], M4-Instruct [14]	Kinetics-400 [11], Something-Something [8], TGIF-QA [10], TVQA [12], CLEVRER [25], NExT-QA [23], FAVD [19], MovieChat-1K [20], TextVR [22], ShareGPT-Video [5], ShareGPT-4o [6], Oops [7], OVIS [18], UVO [21], GUI-World [4], Vript [24], HT-Step [1], Ego4D [9], LLaVA-Video-178K [26], VideoChat-Flash [15]

large on disk or whose licenses made automatic download impractical. After this filtering, the final training set contains 71,927 instances are drawn from publicly available datasets, listed in Table C.

B Additional Experiments

B.1 Ablation Study on Additional Fine-tuning

Table D presents an ablation study to isolate the effect of our proposed design from that of fine-tuning. Specifically, we compare XComp with a baseline that applies the same fine-tuning procedure (on 2.5% of the data as mentioned in Appendix A.2) to the original VideoChat-Flash-2B model, without introducing our LP-Comp and QC-Comp. VideoChat-Flash-2B+FT achieves comparable performance to the original model, suggesting a limited benefit from fine-tuning. This indicates that the performance gains of XComp stem from our method’s enhancements, rather than from fine-tuning alone.

B.2 Multi-hop NIAH

Figure A shows the results of the Multi-Hop Needle-in-a-Haystack QA task [15] which is designed to evaluate extreme long-context reasoning abilities. This benchmark embeds a reasoning path of images within long video sequences, where each image contains clues guiding the model to the next. Given a starting point, the model must trace the correct path, identify the target image (needle), and

Table D: Ablation study of fine-tuning.

Average Duration	Size	LongVideoBench 473s	MLVU 651s	VideoMME (Long) 2386s	LVBench 4101s
VideoChat-Flash-2B [15]	2B	58.3	65.7	44.9	42.9
VideoChat-Flash-2B+FT	2B	57.4	65.6	44.7	43.2
XComp	2B	59.7	66.7	45.6	46.2

66 answer a related question. In this experiment, we use QC-Comp with `n_selected_frames` = 1
 67 and `n_repeat` = 8. It accurately selects the keyframe with 65% accuracy at a sequence length
 68 of 6,144, leading to QA average accuracies of 72.6 and 72.2 over 2,000 to 10,000 total frames for
 69 VideoChat-Flash+QC-Comp and XComp, respectively.

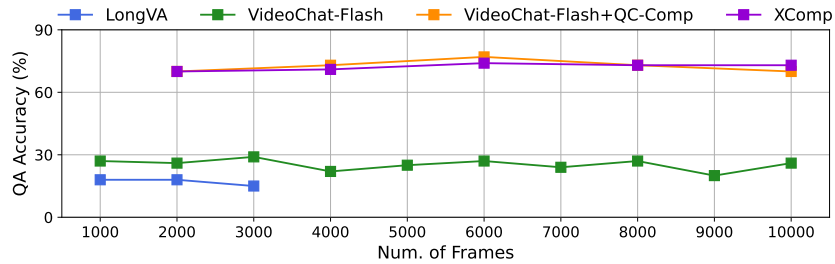


Figure A: Multi-Hop Needle-in-a-Haystack QA Performance.

70 C Broader Impacts

71 This work presents advancements in vision-language modeling, with a focus on improving long video
 72 understanding. While our contributions are primarily foundational, we acknowledge potential societal
 73 risks associated with vision-language models, including misuse of disinformation, privacy concerns,
 74 and the amplification of biases present in training data. We encourage future work to include fairness
 75 assessments and responsible release strategies. Where applicable, safeguards should be considered to
 76 mitigate unintended harms.

77 References

- 78 [1] Triantafyllos Afouras, Effrosyni Mavroudi, Tushar Nagarajan, Huiyu Wang, and Lorenzo
79 Torresani. Ht-step: Aligning instructional articles with how-to videos. *Advances in Neural*
80 *Information Processing Systems*, 36:50310–50326, 2023. 3
- 81 [2] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang,
82 Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*,
83 2025. 1
- 84 [3] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and
85 Judy Hoffman. Token merging: Your vit but faster. *arXiv preprint arXiv:2210.09461*, 2022. 1
- 86 [4] Dongping Chen, Yue Huang, Siyuan Wu, Jingyu Tang, Huichi Zhou, Qihui Zhang, Zhigang
87 He, Yilin Bai, Chujie Gao, Liuyi Chen, et al. Gui-world: A video benchmark and dataset
88 for multimodal gui-oriented understanding. In *The Thirteenth International Conference on*
89 *Learning Representations*, 2024. 3
- 90 [5] Lin Chen, Xilin Wei, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong
91 Duan, Zhenyu Tang, Li Yuan, et al. Sharegpt4video: Improving video understanding and
92 generation with better captions. *Advances in Neural Information Processing Systems*, 37:19472–
93 19495, 2024. 3
- 94 [6] E Cui, Y He, Z Ma, Z Chen, H Tian, W Wang, K Li, Y Wang, W Wang, X Zhu, et al.
95 Sharegpt-4o: Comprehensive multimodal annotations with gpt-4o, 2024. 3
- 96 [7] Dave Epstein, Boyuan Chen, and Carl Vondrick. Oops! predicting unintentional action in video.
97 *arXiv preprint arXiv:1911.11206*, 2019. 3
- 98 [8] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne
99 Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag,
100 et al. The "something something" video database for learning and evaluating visual common
101 sense. In *Proceedings of the IEEE international conference on computer vision*, pages 5842–
102 5850, 2017. 3
- 103 [9] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit
104 Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world
105 in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF conference on computer*
106 *vision and pattern recognition*, pages 18995–19012, 2022. 3
- 107 [10] Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. Tgif-qa: Toward
108 spatio-temporal reasoning in visual question answering. In *Proceedings of the IEEE conference*
109 *on computer vision and pattern recognition*, pages 2758–2766, 2017. 3
- 110 [11] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijaya-
111 narasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human
112 action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 3
- 113 [12] Jie Lei, Licheng Yu, Mohit Bansal, and Tamara L Berg. Tvqa: Localized, compositional video
114 question answering. *arXiv preprint arXiv:1809.01696*, 2018. 3
- 115 [13] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan
116 Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint*
117 *arXiv:2408.03326*, 2024. 3
- 118 [14] Feng Li, Renrui Zhang, Hao Zhang, Yuanhan Zhang, Bo Li, Wei Li, Zejun Ma, and Chunyuan
119 Li. Llava-next-interleave: Tackling multi-image, video, and 3d in large multimodal models.
120 *arXiv preprint arXiv:2407.07895*, 2024. 3
- 121 [15] Xinhao Li, Yi Wang, Jiashuo Yu, Xiangyu Zeng, Yuhang Zhu, Haian Huang, Jianfei Gao,
122 Kunchang Li, Yinan He, Chenting Wang, et al. Videochat-flash: Hierarchical compression for
123 long-context video modeling. *arXiv preprint arXiv:2501.00574*, 2024. 1, 2, 3, 4

- 124 [16] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee.
125 Llava-next: Improved reasoning, ocr, and world knowledge, January 2024. [3](#)
- 126 [17] Ye Liu, Siyuan Li, Yang Wu, Chang-Wen Chen, Ying Shan, and Xiaohu Qie. Umt: Unified multi-
127 modal transformers for joint video moment retrieval and highlight detection. In *Proceedings*
128 *of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3042–3051,
129 2022. [1](#)
- 130 [18] Jiyang Qi, Yan Gao, Yao Hu, Xinggang Wang, Xiaoyu Liu, Xiang Bai, Serge Belongie, Alan
131 Yuille, Philip HS Torr, and Song Bai. Occluded video instance segmentation: A benchmark.
132 *International Journal of Computer Vision*, 130(8):2022–2039, 2022. [3](#)
- 133 [19] Xuyang Shen, Dong Li, Jinxing Zhou, Zhen Qin, Bowen He, Xiaodong Han, Aixuan Li, Yuchao
134 Dai, Lingpeng Kong, Meng Wang, et al. Fine-grained audible video description. In *Proceedings*
135 *of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10585–10596,
136 2023. [3](#)
- 137 [20] Enxin Song, Wenhao Chai, Guan hong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu,
138 Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, et al. Moviechat: From dense token to
139 sparse memory for long video understanding. In *Proceedings of the IEEE/CVF Conference on*
140 *Computer Vision and Pattern Recognition*, pages 18221–18232, 2024. [3](#)
- 141 [21] Weiyao Wang, Matt Feiszli, Heng Wang, and Du Tran. Unidentified video objects: A benchmark
142 for dense, open-world segmentation. In *Proceedings of the IEEE/CVF international conference*
143 *on computer vision*, pages 10776–10785, 2021. [3](#)
- 144 [22] Weijia Wu, Yuzhong Zhao, Zhuang Li, Jiahong Li, Hong Zhou, Mike Zheng Shou, and
145 Xiang Bai. A large cross-modal video retrieval dataset with reading comprehension. *Pattern*
146 *Recognition*, 157:110818, 2025. [3](#)
- 147 [23] Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. Next-qa: Next phase of question-
148 answering to explaining temporal actions. In *Proceedings of the IEEE/CVF conference on*
149 *computer vision and pattern recognition*, pages 9777–9786, 2021. [3](#)
- 150 [24] Dongjie Yang, Suyuan Huang, Chengqiang Lu, Xiaodong Han, Haoxin Zhang, Yan Gao, Yao
151 Hu, and Hai Zhao. Vript: A video is worth thousands of words. *Advances in Neural Information*
152 *Processing Systems*, 37:57240–57261, 2024. [3](#)
- 153 [25] Kexin Yi, Chuang Gan, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua B
154 Tenenbaum. Clevrer: Collision events for video representation and reasoning. *arXiv preprint*
155 *arXiv:1910.01442*, 2019. [3](#)
- 156 [26] Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Video
157 instruction tuning with synthetic data. *arXiv preprint arXiv:2410.02713*, 2024. [3](#)