FrameFusion: Combining Similarity and Importance for Video Token Reduction on Large Vision Language Models

Supplementary Material

7. Detailed Experiment Setup

7.1. Method Setup

7.1.1. Baseline Setup

For the baselines StreamingLLM [31] and FastV [3], we follow the official implementations and set the attention sink size of StreamingLLM to 8 and K in FastV to 2.

7.1.2. FrameFusion Setup

Workflow details. For FrameFusion, token merging is only applied to visual tokens because they dominate input length and show higher similarity between adjacent frames, enabling $O(N)$ complexity merging. The detailed workflow of FrameFusion is shown in Figure 10.

Hyperparameters. The merging ratios across layers are controlled by two hyperparameters: $S_{\text{threshold}}$ and $N_{\text{threshold}}$, as discussed in Section 4.1.

$S_{\text{threshold}}$ defines the minimum cosine similarity required for two tokens to be considered similar and merged. Since similarity distributions vary across models, we set $S_{\text{threshold}}$ to match the median of similarity at the first model layer under typical input cases, such as 128 samples from the VideoMME dataset. For the Llava-Video series, we set $S_{\text{threshold}} = 0.6$; for MiniCPM-V, we set $S_{\text{threshold}} = 0.7$; for NVILA-2B,8B,15B, We set $S_{\text{threshold}} = 0.6, 0.75, 0.8$, respectively.

$N_{\text{threshold}}$ determines the transition from merging to pruning. If the number of similar tokens (tokens with cosine similarity above $S_{\text{threshold}}$) falls below $N_{\text{threshold}}$, the model switches to pruning. We set $N_{\text{threshold}} = 0.1$ to avoid extensive similarity computations across the entire model.

To ensure the merging process does not excessively reduce the token count below the predefined token budget C, we precompute the maximum number of token pairs (N_{max}) that can be merged per layer. If the actual number of pairs exceeds N_{max}, only the top N_{max} pairs with the highest cosine similarity are merged. Any remaining merging or pruning steps are skipped, and the model proceeds with a standard forward pass.

7.2. Model Setup

We follow the default frame count settings for all models, except for NVILA-Lite-2B. Since NVILA-Lite-2B is not specifically trained for video tasks, we set its frame count to 64. For the Llava-Video series and Minicpm-V, the frame count is set to 64, while for NVILA-Video-8B and NVILA-Video-15B, it is set to 256.

8. Additional Experiment Results

8.1. Performance

8.1.1. Computation-Accuracy Trade-off

We further investigate the trade-off between computational cost and accuracy. We evaluate the Llava-Video-7B and NVILA-8B models on the VideoMME and VideoNIAH benchmark, respectively. The results are shown in Figure 11 and Figure 12. As the number of FLOPs decreases, other baseline methods exhibit a noticeable decline in accuracy, whereas FrameFusion maintains superior performance.

8.1.2. Performance Across Different Input Length

Figure 13 presents the performance of Llava-Video-7B on the VideoMME benchmark as the number of input frames varies from 8 to 128. Across all configurations, FrameFusion consistently outperforms the baseline methods, demonstrating its robustness to different input length.

8.1.3. Performance Across Different Token Budgets

Table 6 presents the benchmark performance of the Llava-Video-7B model at token budgets ranging from 0.3 to 0.7. At a 30% token budget, FrameFusion achieves strong performance, with a maximum relative drop of less than 3.0% compared to the dense model. As the budget increases to 0.5 and 0.7, the maximum drops further decrease to $\leq$1.2%.

8.1.4. Performance Across Different Models

We present the detailed numeric results of the scalability experiments in Section 5.3.

As shown in Figure 14, FrameFusion consistently outperforms FastV baseline across all model sizes and VideoMME categories, demonstrating comparative performance with the original model at a 30% relative token budget. Note that the model Llava-Video-32B has been removed by its author team. However, in order to demonstrate the generalization capability of our FrameFusion method across variable model sizes, we still include this model in the performance and efficiency tests here.

Table 7 provides the VideoMME scores for various model sizes across different video lengths and categories, offering a numerical breakdown of Figure 14.

Table 8 illustrates how retrieval accuracy scales with the number of input frames, complementing the insights from Figure 8. As shown, FrameFusion maintains consistent accuracy improvements across increasing frame numbers, matching the performance of the original model. In

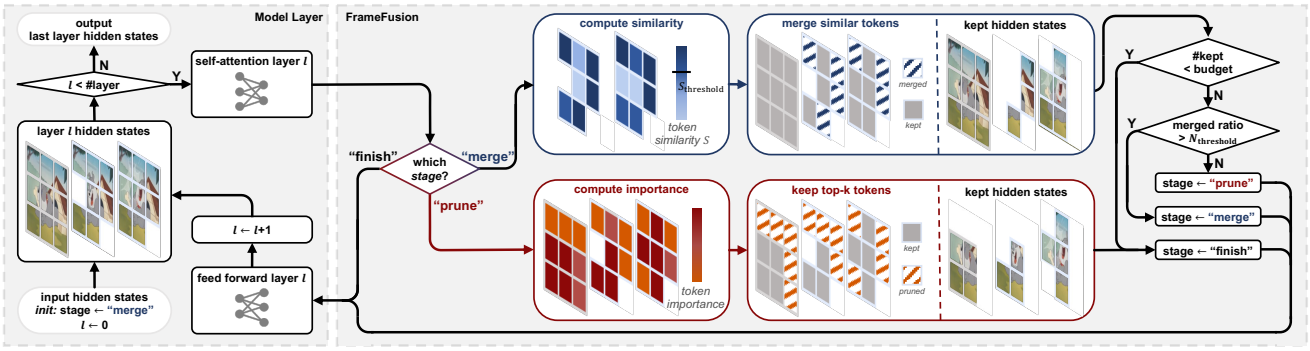


Figure 10. The workflow of FrameFusion when applied to LVLMs. At each layer, FrameFusion performs merging, pruning, or no action between the self-attention and feed-forward layers, depending on the current stage. The stage initially starts as "merge" and updates according to transition conditions.

Model	Method	Budget	VideoMME Score ↑	VideoMME Drop ↓	NExt-QA-MC Score ↑	NExt-QA-MC Drop ↓	NExt-QA-OE Score ↑	NExt-QA-OE Drop ↓	Max. Drop ↓
Llava-Video-7B	Original	1.0	63.2	-	83.2	-	32.1	-	-
	Ours	0.3	61.3	3.0%	81.8	1.7%	31.7	1.2%	3.0%
		0.5	62.6	0.9%	82.7	0.6%	32.1	0.0%	0.9%
		0.7	63.0	0.3%	82.8	0.5%	32.1	0.0%	0.5%
MiniCPM-V-8B	Original	1.0	58.5	-	78.9	-	13.8	-	-
	Ours	0.3	57.4	1.9%	78.2	0.9%	16.3	-18.1%	1.9%
		0.5	58.5	0.0%	78.6	0.4%	17.4	-26.1%	0.4%
		0.7	57.8	1.2%	78.6	0.4%	16.1	-16.7%	1.2%

Table 6. Performance comparison between the original and proposed methods on VideoMME, NExt-QA-MC, and NExt-QA-OE benchmarks with different relative token budgets on Llava-Video-7B model. Drop indicates the relative performance decrease compared to the original method.

contrast, both StreamingLLM and FastV exhibit noticeable drops in accuracy.

We further test the performance of two extra models: Qwen2-VL-7B and InternVL-2.5-8B. As shown in Table 9, our method performs well compared to original models on VideoMME.

8.1.5. Retrieval Benchmark Details

We further investigate the retrieval accuracy details with the VideoNIAH benchmark, as shown in Figure 15. FrameFusion demonstrates similar retrieval performance as the original dense model, with consistent performance across lengths and positions. In contrast, StreamingLLM hardly retrieves the initial frames of the video. FastV does not show particular failure patterns but undergoes uniform performance degradation across grids.

8.1.6. Performance on Image Benchmark

We further investigate our method's performance on an image benchmark: MMMU-Pro-standard. As shown in Table 10, although our method is not designed for image inputs, it still demonstrates comparative performance.

8.2. Efficiency

8.2.1. Efficiency Across Different Model Sizes

We evaluate the scalability of FrameFusion 's efficiency across different model sizes, as shown in Figure 17 and 18. To accommodate the increased KV-Cache and memory overhead, we distribute models across multiple GPUs. With larger models, FrameFusion achieves greater end-to-end speedups, delivering 2.8× for Llava-Video-32B on two GPUs and 3.2× for Llava-Video-72B on four GPUs at a 30% token budget. Besides, FrameFusion reduces memory consumption for KV-Cache to 37% for Llava-Video-32B and 51% for Llava-Video-72B with a 30% token budget.

Model	Method	Short	Medium	Long	KL	FT	SC	AP	LR	ML
Llava-Video-7B	Original	75.8	61.7	52.2	63.1	67.2	61.8	61.7	63.7	58.9
	StreamingLLM	63.4	54.1	46.4	55.1	57.2	56.0	54.2	52.9	48.9
	FastV	68.4	58.0	49.6	59.1	60.0	58.9	57.8	58.1	55.6
	PruMerge	69.7	**60.1**	**50.2**	59.1	63.6	**59.1**	58.9	60.6	**57.8**
	Ours	**74.0**	59.8	50.0	**62.7**	**63.6**	58.0	**61.7**	**60.8**	56.7
Llava-Video-72B	Original	80.9	69.7	62.1	73.2	74.4	68.0	71.4	68.9	62.2
	StreamingLLM	68.2	59.9	59.8	65.7	66.7	59.3	65.6	58.7	58.9
	FastV	73.0	64.9	60.2	66.8	72.8	61.1	69.2	63.2	61.1
	PruMerge	74.0	65.8	60.3	70.4	**73.6**	62.9	68.3	61.6	54.4
	Ours	**78.3**	**67.9**	**60.9**	**72.2**	73.1	**65.6**	**69.7**	**65.9**	**61.1**
NVILA-2B	Original	61.4	48.9	42.4	47.2	56.4	49.8	55.0	51.0	52.2
	StreamingLLM	52.9	43.9	40.3	43.0	49.2	46.2	50.6	44.4	43.3
	FastV	53.7	45.6	40.8	43.8	49.7	46.2	51.4	46.7	43.3
	PruMerge	53.9	45.0	**43.1**	43.5	52.8	45.8	51.7	48.1	45.6
	Ours	**61.3**	**47.0**	43.0	**48.3**	**55.6**	**48.2**	**55.3**	**49.8**	**45.6**
NVILA-8B	Original	74.9	62.1	54.7	64.8	66.4	62.2	61.9	63.7	63.3
	StreamingLLM	61.2	53.8	48.0	54.9	57.8	54.7	52.5	52.2	55.6
	FastV	72.0	56.7	50.0	60.7	62.8	57.6	57.8	58.9	57.8
	PruMerge	67.6	54.9	48.3	57.3	61.1	56.0	54.7	56.2	55.6
	Ours	**74.2**	**57.7**	**51.3**	**60.7**	**65.3**	**59.3**	**58.6**	**62.1**	**58.9**
NVILA-15B	Original	77.3	64.7	55.3	67.2	68.1	62.7	63.3	66.2	66.7
	StreamingLLM	63.8	57.4	54.3	60.6	60.6	55.6	58.6	56.5	60.0
	FastV	69.2	58.7	53.9	62.8	63.3	57.1	60.8	58.1	63.3
	PruMerge	66.0	59.3	52.6	61.0	61.1	55.1	57.5	59.8	61.1
	Ours	**73.2**	**62.3**	**55.0**	**64.6**	**68.1**	**60.9**	**61.1**	**62.5**	**65.6**
MiniCPM-V-8B	Original	69.1	56.6	49.8	59.0	63.6	54.2	63.3	54.9	60.0
	StreamingLLM	61.1	51.8	48.4	54.6	58.1	52.2	56.4	49.7	55.6
	FastV	67.1	53.9	**49.2**	57.2	59.2	53.8	**60.8**	**54.6**	56.7
	Ours	**69.7**	**54.1**	48.3	**57.9**	**63.1**	**53.8**	60.3	54.4	**56.7**

Table 7. Numeric VideoMME scores of different methods and model sizes across various video categories. "KL", "FT", "SC", "AP", "LR", "ML" are short for "Knowledge", "Film & Television", "Sports Competition", "Artistic Performance", "Life Record", and "Multilingual".

Method	Number of frames 64	85	107	128	Max. Relative Drop
Original	76.4	78.4	80.7	82.9	-
StreamingLLM	23.3	25.8	27.6	27.6	70%
FastV	58.2	63.6	65.8	69.3	24%
Ours	**75.3**	**78.2**	**80.0**	**83.6**	**1%**

Table 8. Numeric VideoNIAH retrieval accuracy of different methods across various frame counts.

Setting	Qwen2-VL-7B Original	Qwen2-VL-7B Ours	InternVL-2.5-8B Original	InternVL-2.5-8B Ours
w/o sub	55.9	58.4	63.1	62.3
w/sub	60.6	61.1	66.3	64.2

Table 9. Performance comparison between original and Framefusion of Qwen2-VL-7B and InternVL-2.5-8B on VideoMME.

8.2.2. Token Reduction Details

FrameFusion reduces computational cost through both token merging and pruning. Using 128 samples from the VideoMME dataset with the Llava-Video-7B model, we calculate the token count per layer. As shown in Figure 19, FrameFusion progressively reduces tokens per layer, achieving the desired relative token budget (represented by the area under the line).

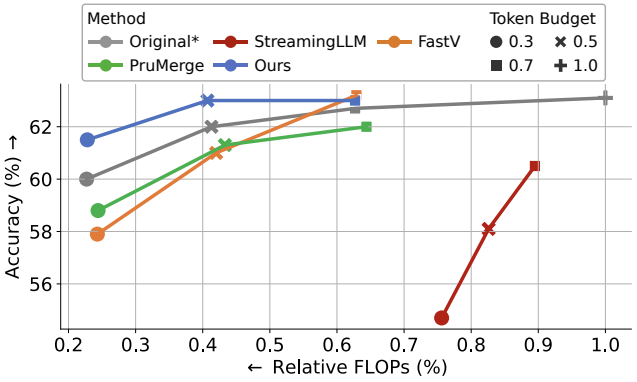


Figure 11. The accuracy-computation trade-offs of various token compression methods, tested on Llava-Video-7B with VideoMME benchmark. Original* represents the original model with reduced frame rates.

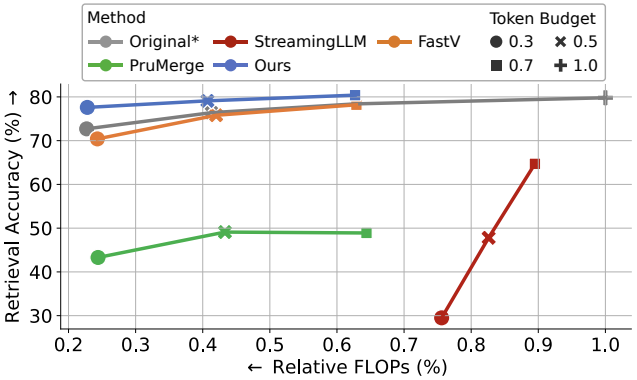


Figure 12. The accuracy-computation trade-offs of various token compression methods, tested on NVILA-8B with VideoNIAH benchmark. Original* represents the original model with reduced frame rates.

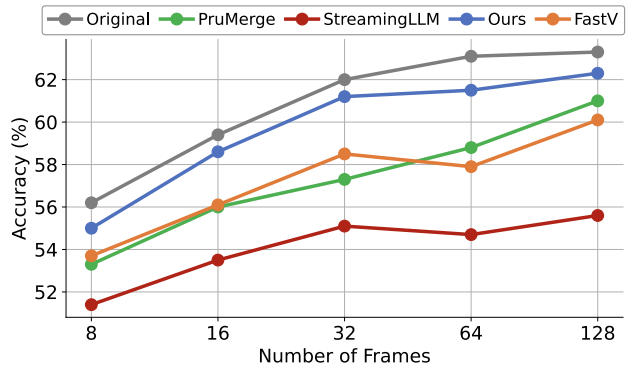


Figure 13. The VideoMME performances for the Llava-Video-7B across various numbers of input frames.

Model	Original	Fastv	Ours
NVILA-2B	23.6	**23.8**	23.1
NVILA-8B	30.3	28.7	**29.0**
NVILA-15B	36.1	30.6	**32.8**

Table 10. The MMMU-Pro-standard performance across NVILA-2B, 8B, and 15B for different methods.

Choice	VideoNIAH	VideoMME	NExt-QA	Avg.
inner product	71.3	58.9	55.0	61.7
minkowski-2	71.3	60.9	**57.1**	63.1
minkowski-1	71.3	61.0	57.0	63.1
cosine similarity	**75.1**	**61.4**	56.9	**64.5**

Table 11. Performance of different distance calculation strategies with the same relative token budget of 30% on VideoNIAH, VideoMME, and NExt-QA.

8.3. Ablation Study

8.3.1. Similarity Computation Strategy

We empirically study whether our approach successfully finds the most similar token pairs. All three $O(N)$ complexity strategies are compared against the posterior optimal upper bound, which merges the most similar tokens using the full $N \times N$ similarity computation. As shown in Figure 20, given different merging rate, the token pairs found by our method constantly shows the highest average similarity. We successfully reach 90% average similarity with only $1/10^4$ computing overhead. Further ablations are detailed in Section 5.5

8.3.2. Distance Metrics

FrameFusion adopts cosine similarity as the distance metric between tokens. To evaluate the impact of different distance metrics, we replace cosine similarity with the inner product, Minkowski-2, and Minkowski-1 distance. We test the performance of FrameFusion at a 30% token budget. As shown in Table 11, the average accuracy using cosine similarity is 2.8%, 1.4%, and 1.4% higher than the baseline metrics, respectively.

8.3.3. Choice of Similarity Threshold

We conduct ablation studies on the sensitivity of the similarity ($S_{\text{threshold}}$) and merging-pruning transition ($N_{\text{threshold}}$) thresholds on NVILA-8B. As shown in Table 12, our method shows robust performance to threshold variations.

8.3.4. Effect of Positional Embedding

We investigate the impact of positional embeddings on token similarity. Specifically, we compare models with and without positional embedding at the first layer and analyze the resulting changes in the similarity of the input hid-

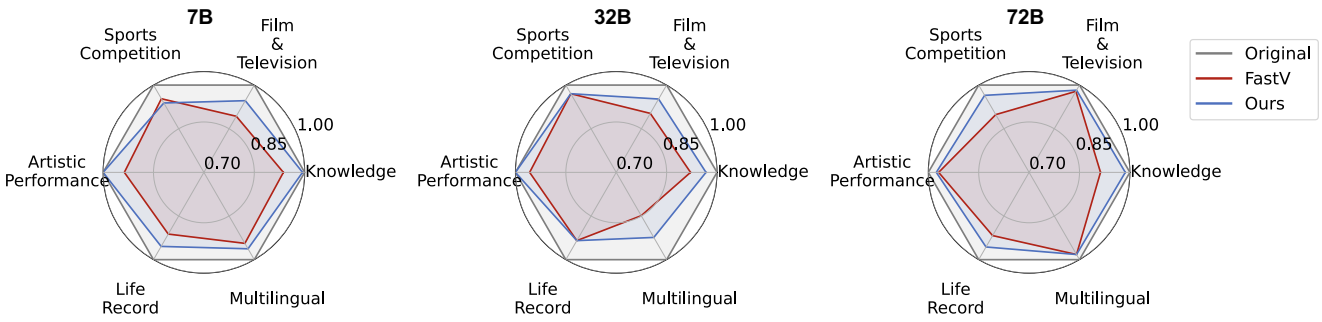


Figure 14. The VideoMME performance for each category across Llava-Video-7B, 32B, and 72B for different methods. All scores are normalized by the original model.

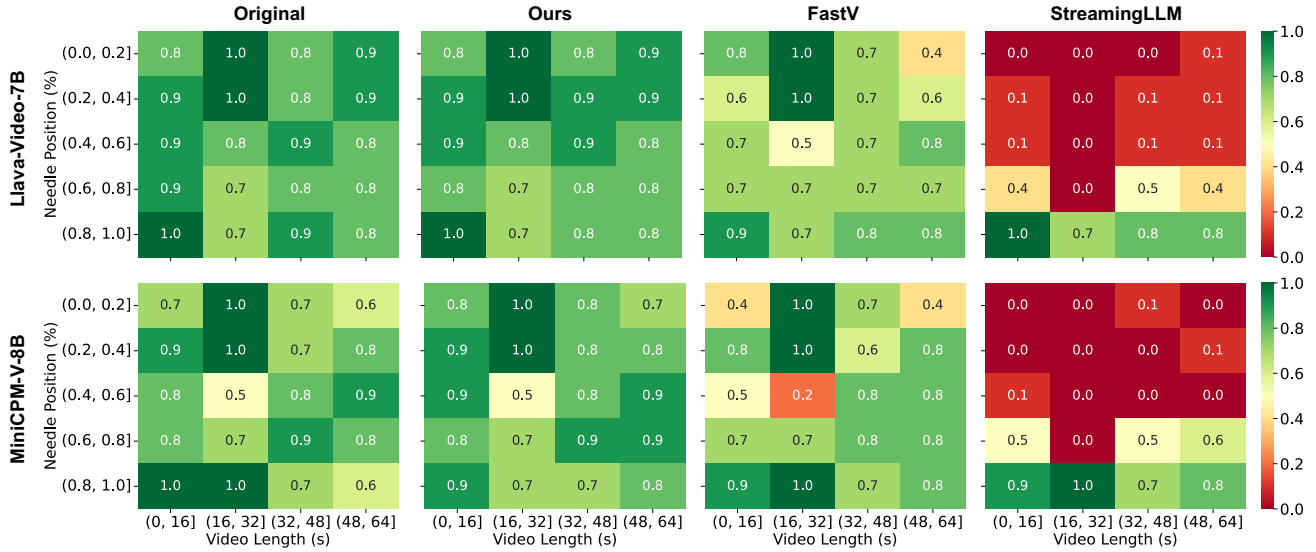


Figure 15. VideoNIAH retrieval accuracy of the Llava-Video-7B and MiniCPM-V-8B models using different token compression methods across varying video lengths and retrieval positions. All token compression methods employ 30% relative token budget.

Threshold	Value	VideoNIAH	VideoMME	NeXT-QA-mc
$S_{\text{threshold}}$	0.6	73.6	56.9	56.5
	0.7 (default)	73.3	57.4	56.3
	0.8	72.9	57.6	56.5
	0.9	72.2	57.7	56.3
$N_{\text{threshold}}$	0.1 (default)	73.3	57.4	56.3
	0.2	74.0	57.6	56.5
	0.3	74.2	57.5	56.5

Table 12. Performance of different similarity and merging-pruning transition thresholds on VideoNIAH, VideoMME, and NExt-QA.

den states to the second layer. The results show that the L1-norm of the similarity matrix changes by an absolute amount of 0.0087 ± 0.0010, corresponding to a relative change of $2.73\% \pm 0.66\%$. It shows that the token contents, rather than the positional embeddings, dominate token similarity.

9. Asymptotic Complexity Analysis

We estimate the computing cost of FrameFusion following the approach of FastV [3]. Given a model with L layers and a specified relative token budget C, FrameFusion operates in the merging stage from layer 0 to layer $K-1$, then transitions to the pruning stage at layer K. Let N_l denote the number of tokens in layer l before token reduction at this layer. Note that N_{l+1} represents the number of tokens of layer l after token reduction, and we let N_{-1} equal the original input token length N. FrameFusion reduces N_l with merging and pruning at the initial $K+1$ layers. After the token reduction, the remaining tokens for the successive layers are calculated as follows:

$$N_l = \frac{L \times C \times N - (N_0 + \ldots + N_K)}{L - K - 1}, l \in [K+1, L) \tag{4}$$

The model inference computation FLOPs $F(N_l, N_{l+1})$

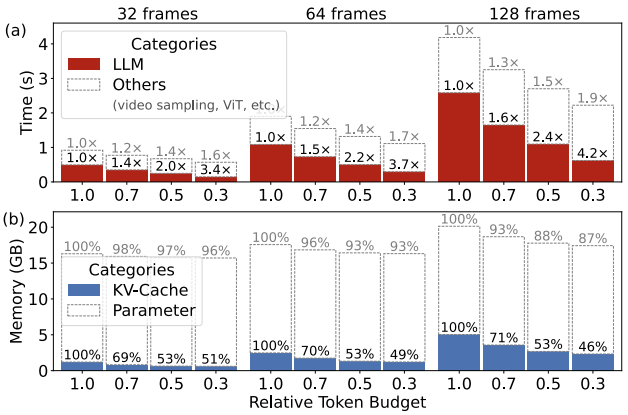


Figure 16. Runtime and memory breakdown of Llava-Video-7B on a single A100-80GB GPU using FrameFusion. A relative token budget of 1.0 represents the original dense model. Numbers on bars show (a) LLM and end-to-end speedups and (b) LLM's KV-Cache and total relative memory.

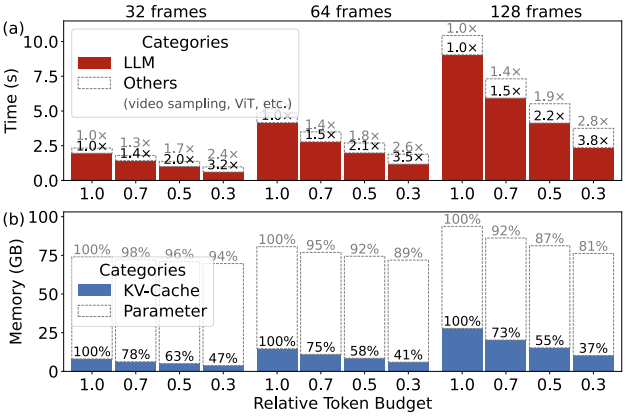


Figure 17. Runtime and memory breakdown of Llava-Video-32B on two A100-80GB GPUs using FrameFusion. A relative token budget of 1.0 represents the original dense model. Numbers on bars show (a) LLM and end-to-end speedups and (b) LLM's KV-Cache and total relative memory.

of layer l is calculated as follows:

$$F(N_l, N_{l+1}) = 4N_l D^2 + 2N_l^2 D + 3N_{l+1} DM \quad (5)$$

where D denotes the hidden state size, and M denotes the intermediate FFN size. The additional computation $F'(N_l)$ introduced by FrameFusion during similarity computation is:

$$F'(N_l) = 3N_l D \quad (6)$$

Note that the additional computation F' introduced by FrameFusion shows negligible asymptotic complexity with respect to input length and model size, compared with the $O(N^2 D)$ and $O(ND^2)$ complexities of the original model.

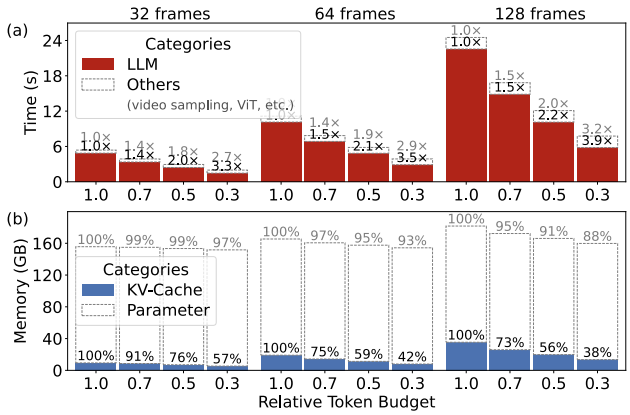


Figure 18. Runtime and memory breakdown of Llava-Video-72B on four A100-80GB GPUs using FrameFusion. A relative token budget of 1.0 represents the original dense model. Numbers on bars show (a) LLM and end-to-end speedups and (b) LLM's KV-Cache and total relative memory.

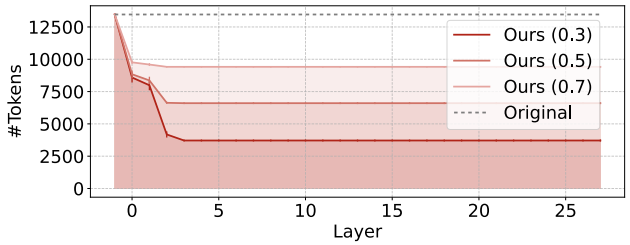


Figure 19. Average number of tokens per layer in the Llava-Video-7B model with FrameFusion at different relative token budgets. Error bars represent variance across data items.

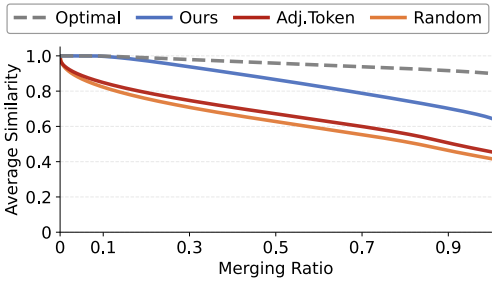


Figure 20. The average token similarity of the merged tokens for the first layer of Llava-Video-7B model across various merging rates.

10. Additional Observation Details

10.1. Similarity Distribution Details

We take 128 videos from the VideoMME dataset and calculate the variance in token similarity across different layers. As shown in Figure 21, the similarity variance decreases in the deeper layers of the model, validating Observation 2.

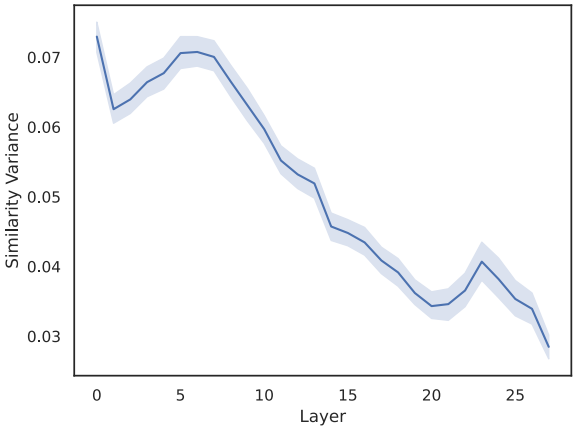


Figure 21. Average token similarity variance per LLM layer in the Llava-Video-7B model, tested on 128 samples from the VideoMME dataset. Shading represents the variance across data items.

No significant outliers are observed in token similarity, in contrast to the common outliers seen with respect to the magnitude of hidden features [36, 38].

10.2. Observations on Additional Models

In addition to the analysis of the Llava-Video model in Section 3, we conduct a similar study on the MiniCPM architecture. Results are presented in Figures 22, 23, 24, and 25.

Overall, the conclusions align with those of the Llava-Video model, with a few notable differences: Firstly, as shown in Figure 22, MiniCPM, which incorporates Q-Former [17, 34], exhibits additional high similarity among visual tokens within the same frame. However, the prominent 210th sub-diagonal persists, supporting our token similarity calculation strategy. Secondly, as shown in Figure 23, high similarity decreases less steeply in deeper layers for MiniCPM compared to Llava-Video. Despite this, the superior efficiency of cascaded merging at shallower layers ensures that Design Choice 2 remains valid.

10.3. Video Pruning Visualization

We select a video example to visualize the effect of our token merging strategy. Figure 26 shows the frames of the original video sampled at a frame rate of 1 fps. In Figure 27, we present the video input to the model after token merging in Layer 0, where blank patches indicate tokens that have been merged. Furthermore, we replace the blank regions with the average of the merged patches, and the resulting visualization is shown in Figure 28. As shown in the examples, FrameFusion token merging strategy successfully merges similar visual tokens, reducing the computational costs, while maintaining high validity of the video.

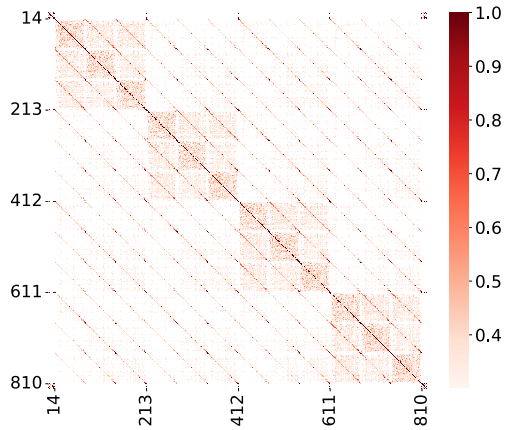


Figure 22. Token similarities between all input tokens at the first LVLM layer in MiniCPM-V-8B.

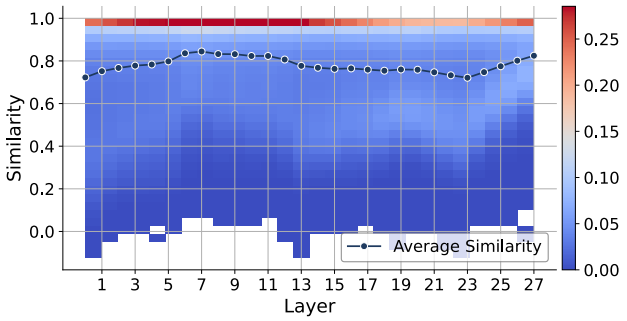


Figure 23. Heatmap of token similarity across different model layers for the MiniCPM-V-8B model. Each cell represents the similarity at a specific layer, with color intensity denoting distribution frequency. The line overlay shows the average token similarity across layers.

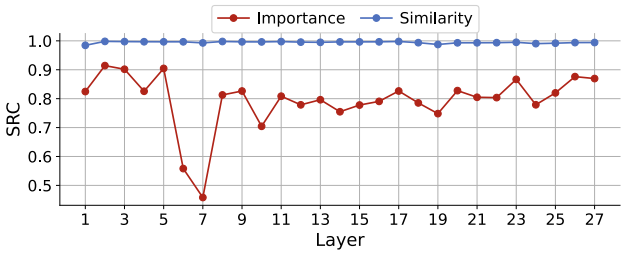


Figure 24. Spearman Rank Correlation (SRC) between adjacent layers for the MiniCPM-V-8B model.

10.4. Importance-Similarity Joint-Distribution

We visualize the joint distribution of token importance and similarity across different layers of Llava-Video-7B. As shown in Figure 29, it can be observed that in the shallow layers of the model, a significant number of tokens exhibit both high similarity and high importance values. Frame-

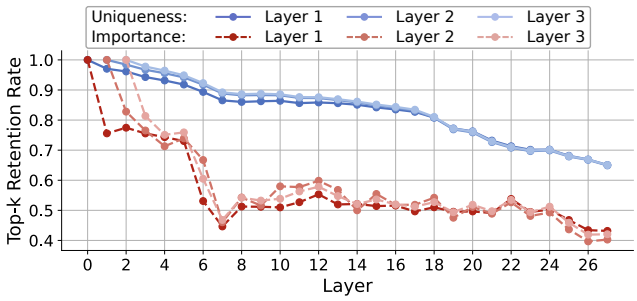


Figure 25. The Top-30% retention rate across model layers for the MiniCPM-V-8B model, using different retention metrics and reference layers.

Fusion can effectively compress these tokens. This phenomenon becomes less apparent in the deeper layers of the model, supporting our design choice of performing token merging in the shallow layers of the model.

11. Additional Discussion on Related Works

Prior works have also explored token merging in image-based tasks [13, 25, 45]. For instance, while FrameFusion adopts an $O(N)$ *temporal* merging strategy, EVL-Gen [13] performs $O(N^2)$ *spatial* merging via bipartite matching among tokens. AIM [45] similarly adopts bipartite-matching-based merging prior to the first layer of the LLM, followed by a token pruning process in subsequent LLM layers, ultimately reducing the number of visual tokens to zero. LLaVA-Prumerge [25] first prunes tokens at the output of the visual encoder and then merges the pruned tokens into the top-k most similar remaining tokens. In all these methods, the similarity computation incurs a complexity of $O(N^2)$. Although the computational efficiency of is comparable at the image scale ($N \approx 256$), our method scales more effectively to video scenarios where N can reach 10K to 1M tokens.

12. Limitation and Future Works

While FrameFusion demonstrates significant improvements in token reduction and efficiency for video LVLMs, certain challenges remain for future work. First, the similarity-based merging process can be further refined to better handle highly diverse or complex video content, minimizing potential information loss. Second, the reliance on predefined similarity and importance metrics calls for the development of adaptive and task-specific strategies to improve generalization across diverse scenarios. Future work will focus on designing more robust similarity measures and integrating FrameFusion with advanced token-efficient architectures.

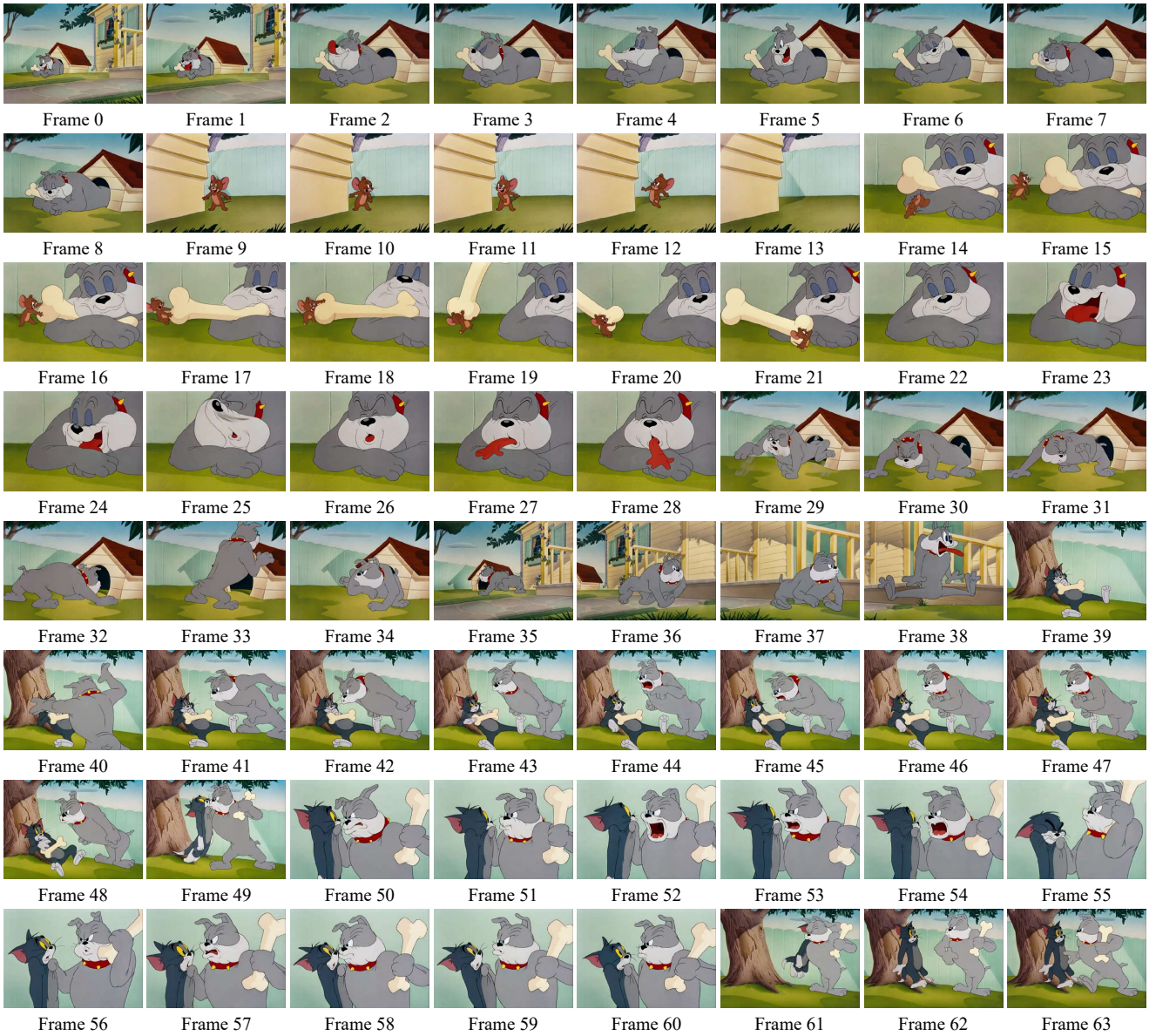


Figure 26. An example input video with 1 fps frame rate.

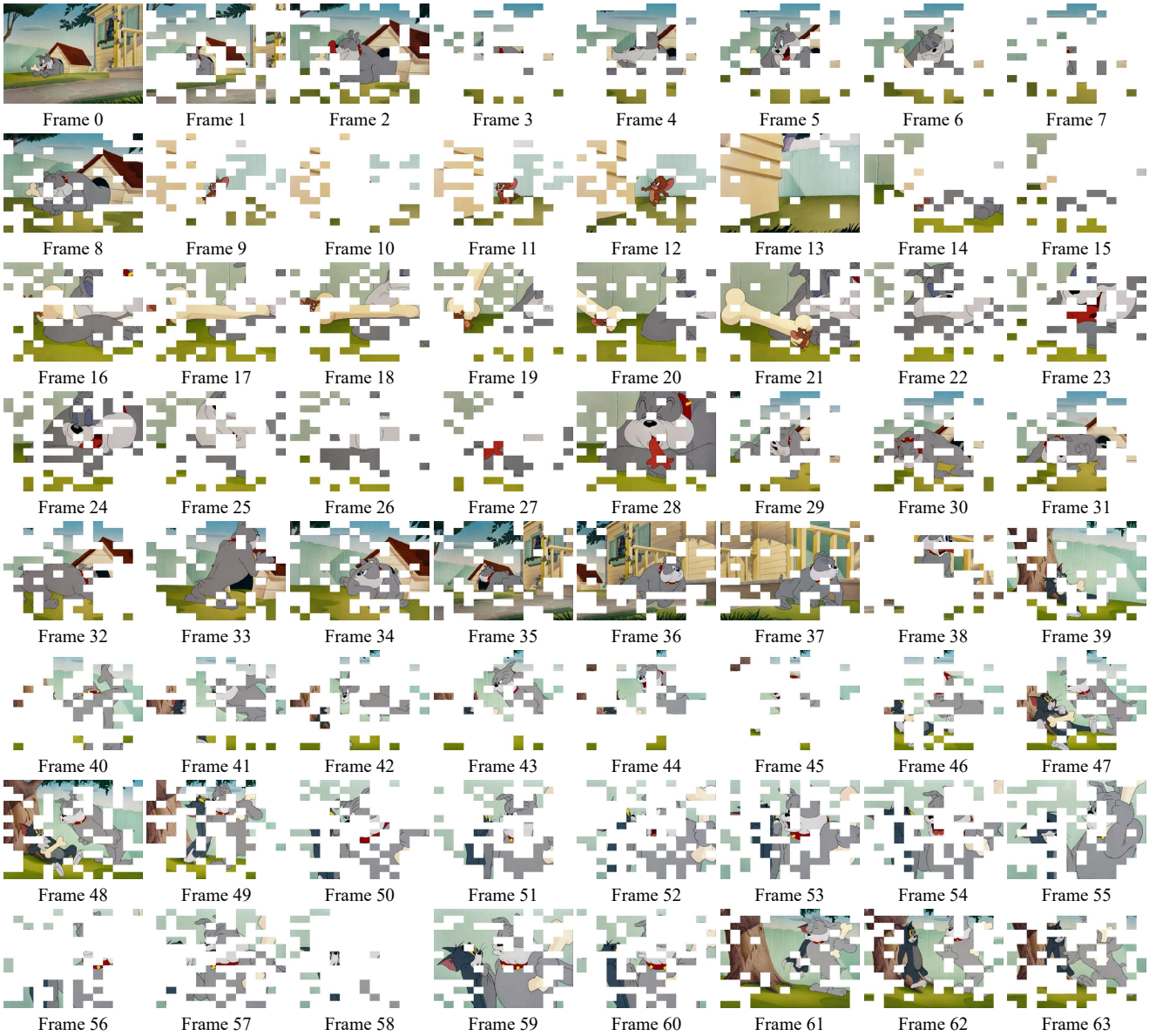


Figure 27. The example of the video after token merging. Merged tokens are visualized with the blank blocks.

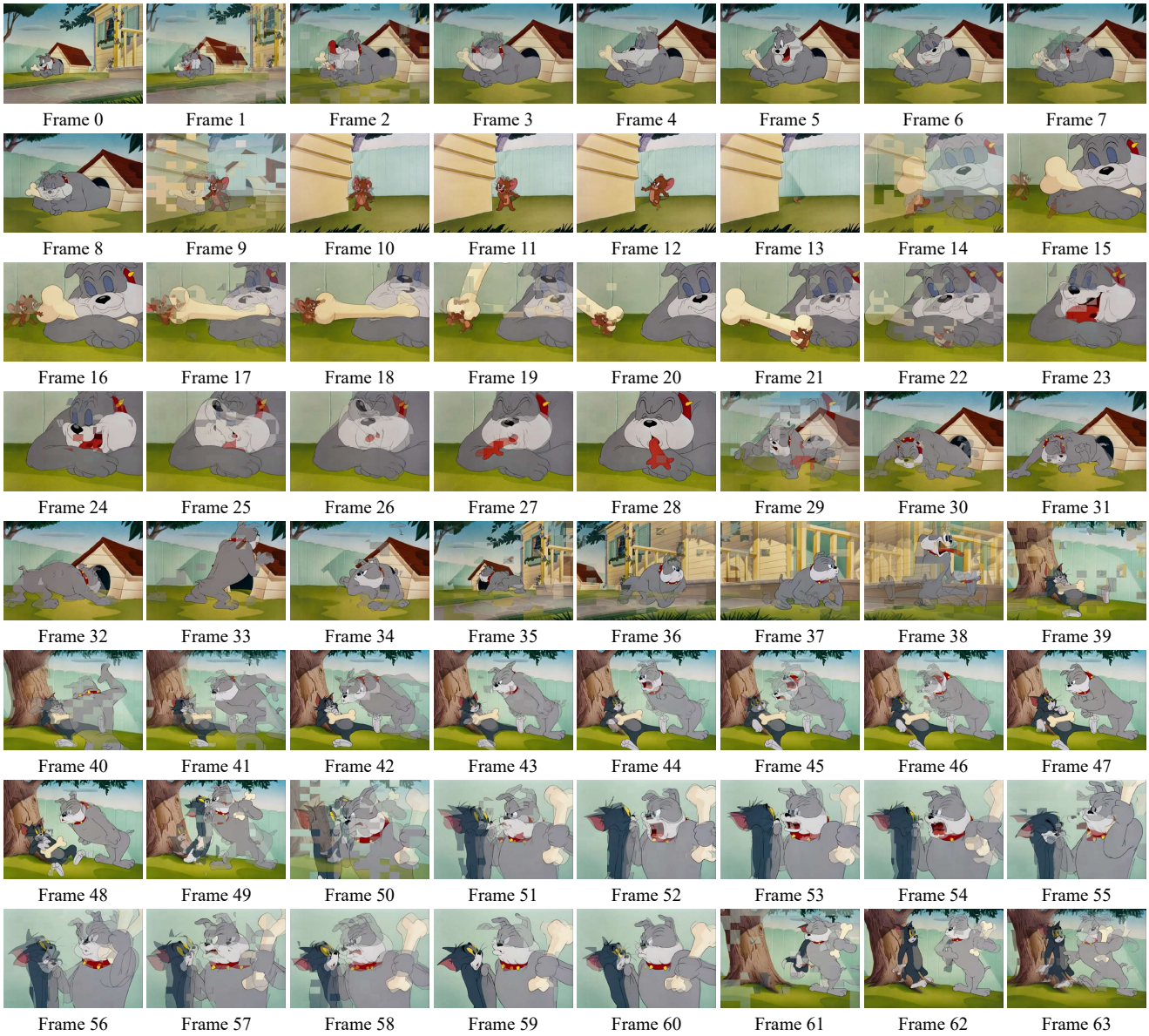
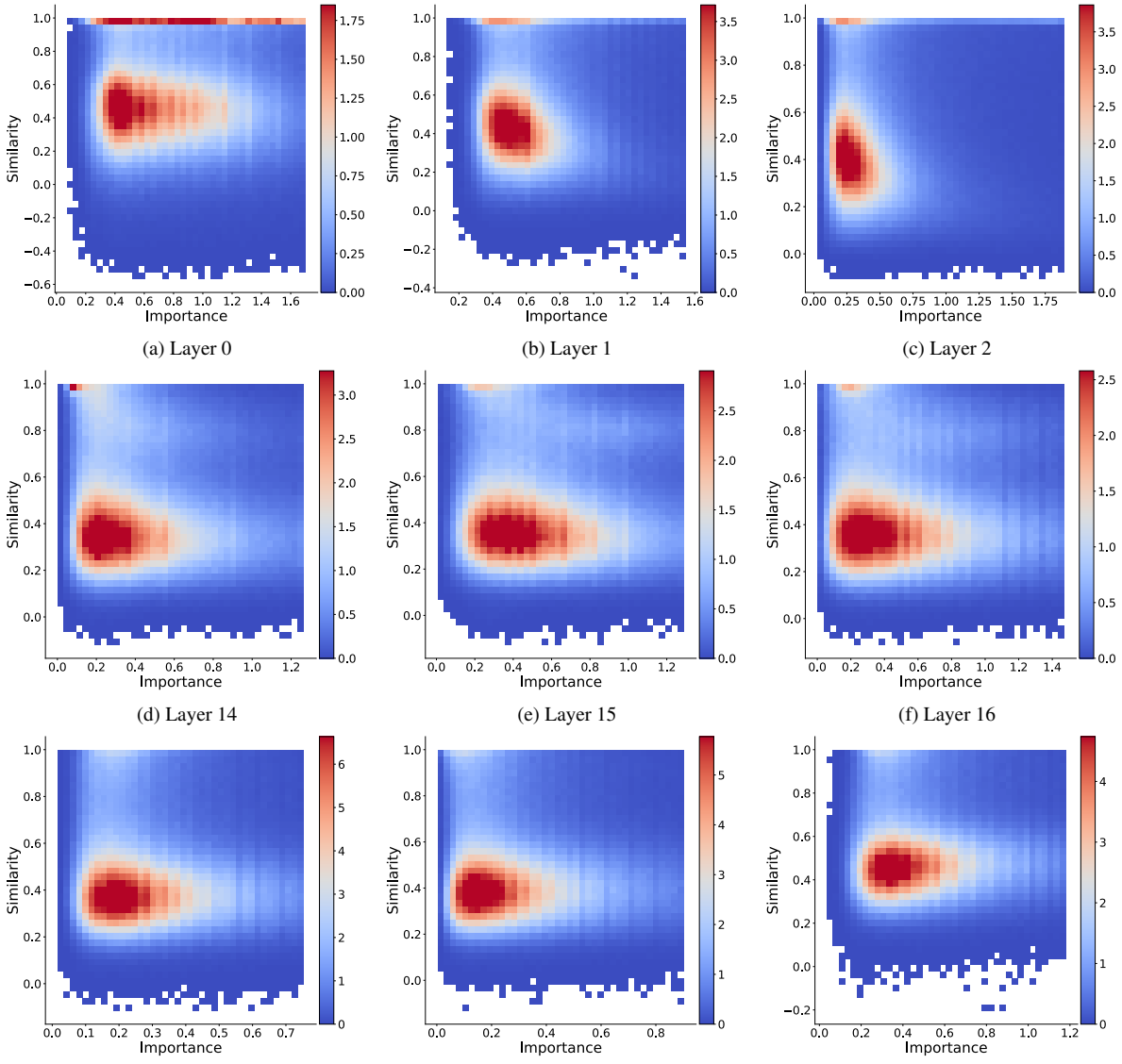


Figure 28. The example of the video after token merging. Merged tokens are visualized with the average image patches.



(a) Layer 0 (b) Layer 1 (c) Layer 2

(d) Layer 14 (e) Layer 15 (f) Layer 16

(g) Layer 25 (h) Layer 26 (i) Layer 27

Figure 29. Importance-similarity joint-distribution of different layers, with color intensity denoting distribution frequency.