MeToM: Metadata-Guided Token Merging for Efficient Video LLMs

Supplementary Material

A. Ablation Study

Ablation study on merging modules. To verify the effectiveness of each component in MeToM, we conduct an ablation study on LLaVA-OneVision-7B [10], as summarized in Table 5. **RPM** merges tokens with low information density during the tokenization stage, reducing computational costs while ensuring performance. **BTM** leverages spatiotemporal information density to compress tokens, greatly reducing the computational burden. Finally, the synergy of **RPM**, **BTM**, and **MATM** achieves the optimal trade-off between efficiency and accuracy.

Table 5. Ablation study results of the main components of MeToM on VideoMME and NExT-QA benchmarks.

RPM	BTM	MATM	VideoMME			NExT-QA		
			R.Acc. ↑	R.TTFT ↓	R.N_V ↓	R.Acc. ↑	R.TTFT ↓	R.N_V ↓
✓	✗	✗	101.4	79.1	75.0	**100.2**	84.5	75.0
✗	✓	✗	98.2	53.9	50.4	98.9	50.4	49.7
✓	✓	✗	100.8	38.2	28.0	99.3	40.6	29.2
	✓		**102.7**	**27.8**	**28.0**	99.1	**32.5**	**29.2**

Ablation study on RPM. We investigate the impact of the merging threshold τ defined in Eqn.(8) of the main paper, which governs the sensitivity of RPM in identifying low-information regions. A higher τ encourages more aggressive merging of spatial tokens, whereas a lower τ preserves more fine-grained details. To determine the optimal setting, we conduct an ablation study by varying τ from 0.15 to 0.35 on the VideoMME and NExT-QA benchmarks.

As illustrated in Fig. 5, the model maintains robust performance when $\tau \leq 0.25$. However, setting $\tau > 0.25$ leads to a sharp decline in accuracy across both benchmarks, indicating that overly aggressive merging begins to discard semantically meaningful visual cues. Conversely, while reducing τ below 0.25 preserves more tokens, it yields negligible performance gains while increasing the computational overhead. Therefore, we empirically set $\tau = 0.25$ as the default configuration to achieve the optimal trade-off between inference efficiency and model performance.

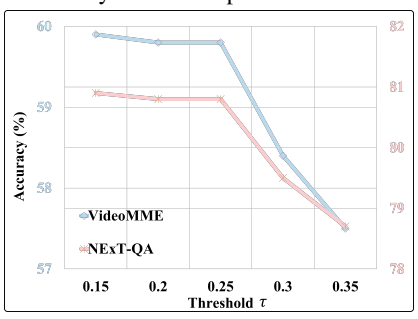

Figure 5. Ablation study on the merging threshold τ of RPM.

Ablation study on MATM. We conduct comprehensive ablation studies to determine the optimal configuration for the Multi-Layer Attention-Guided Token Merging (MATM) module, specifically investigating the **insertion layer** and the **pruning ratio**.

Table 6 compares the performance of inserting MATM at different layers L of the LLM. Inserting MATM at an early stage (Layer 1) yields sub-optimal accuracy (99.1%). This is primarily attributed to the unstable attention score in the initial layers, where visual tokens have not yet sufficiently interacted with the textual context. Conversely, deploying MATM at a deeper layer (Layer 7) fails to deliver significant efficiency gains (43.7% TTFT), as the preceding layers still process the full token sequence. Consequently, **Layer 3** emerges as the optimal choice, achieving the highest accuracy (102.7% on VideoMME) while maintaining a low latency (27.8%). We adopt $L = 3$ as the default setting to balance inference efficiency and model performance.

Table 6. Ablation study on the MATM insertion layers L. The **best** result is in bold, second best underlined.

LLM Layer	VideoMME		NExT-QA	
	R.Acc. ↑	R.TTFT ↓	R.Acc. ↑	R.TTFT ↓
1	99.1	**27.2**	98.8	**31.8**
3	**102.7**	27.8	99.1	32.5
7	100.5	43.7	**99.2**	48.1

We further explore the impact of the visual token pruning ratio $R\%$ in MATM, as shown in Fig. 6. The results indicate that the model maintains acceptable performance as the pruning ratio increases from 20% to 60%. However, increasing the ratio beyond 50% (*e.g.*, to 60%) triggers a sharp decline in accuracy, suggesting that critical visual information is being discarded. To maximize inference efficiency without suffering a catastrophic performance loss, we identify 50% as the optimal trade-off point. Therefore, we set the pruning ratio $R\%$ to 50%.

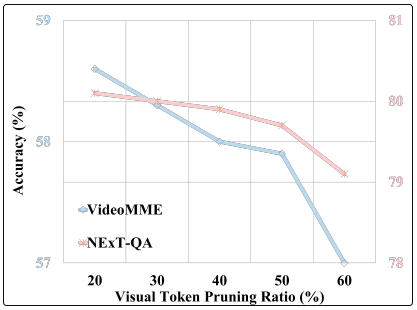

Figure 6. Ablation study on the visual token pruning ratio $R\%$ of MATM.

B. Supplementary Experimental Data

In the main paper, we reported relative performance metrics (normalized to the 100% token budget baseline) in Tables 1, 2, 3, and 4 in the main paper to highlight efficiency trade-offs. To provide a comprehensive view of the actual computational costs, we present the corresponding absolute values for Accuracy, Time-To-First-Token (TTFT), and visual token counts (N_V) in this section. Specifically, the absolute results for LLaVA-OneVision-7B [10], LLaVA-Video-7B [37], Qwen2VL-7B [1], and LLaVA-Video-72B [37] are provided in Tables 7, 8, 9, and 10, respectively.

C. Supplementary Visualizations

To provide a comprehensive qualitative analysis, we present additional visualization examples in this section. As illustrated in the following Figs. 7 and 8, we visualize the effect of our token merging strategy. The middle row displays the spatial information density cues derived from codec metadata. Guided by these cues, MeToM aggressively merges redundant tokens in background areas (visualized as blank blocks in the bottom row) while strictly preserving critical details in the foreground.

D. Limitations and Future Work

MeToM effectively reduces computational costs by leveraging codec metadata. However, relying on such heuristics may occasionally misalign with semantic importance. Furthermore, the substantial size of the LLM backbone poses challenges for deployment on resource-constrained devices. To address this, future work will explore combining MeToM with model compression techniques, such as quantization.

E. Broader Impacts

MeToM significantly enhances the efficiency of Video LLMs by leveraging intrinsic codec metadata for training-free token reduction. By mitigating the heavy computational burden, our framework improves the accessibility of advanced video understanding models and contributes to more sustainable, energy-efficient AI systems.

Table 7. Comparison of training-free token reduction methods using LLaVA-OneVision-7B. The **best** result is in bold, second best underlined.

Token Budget	Method	VideoMME			LongVideoBench			MLVU			EgoSchema			NExT-QA			***Avg.***		
		Acc ↑	TTFT ↓	N_V ↓	Acc ↑	TTFT ↓	N_V ↓	Acc ↑	TTFT ↓	N_V ↓	Acc ↑	TTFT ↓	N_V ↓	Acc ↑	TTFT ↓	N_V ↓	Acc ↑	TTFT ↓	N_V ↓
100%	*LLaVA-OV 7B*	59.0	1.904	22086	56.3	1.712	19624	67.5	2.224	25088	61.3	2.216	25069	80.6	0.630	8116	64.9	1.737	19997
50%	+ FastV	58.4	0.934	11043	56.2	0.842	9812	67.4	1.071	12544	61.0	1.080	12535	80.2	0.330	4058	64.6	0.851	9998
	+ ToMe	59.8	0.963	11043	57.7	0.865	9812	68.8	1.101	12544	62.3	1.108	12535	80.1	0.343	4058	65.7	0.876	9998
	+ DyCoke	60.1	0.831	10555	**57.9**	0.740	9393	68.7	0.941	11978	61.8	0.948	11969	80.4	0.301	3957	65.8	0.752	9570
	+ STTM	**60.7**	0.773	**8579**	57.7	0.804	**9011**	69.7	1.020	**11692**	61.7	0.972	**10944**	80.5	0.312	**3628**	66.1	0.776	**8771**
	+ HoliTom	59.8	0.889	9642	56.8	0.796	9304	68.4	1.007	11821	61.2	1.012	11812	79.9	0.316	3824	65.2	0.804	9281
	+ Ours	60.6	**0.718**	9520	57.8	**0.706**	9129	**69.9**	**0.936**	11760	**62.4**	**0.943**	11751	**80.6**	**0.298**	3794	**66.3**	**0.720**	9191
30%	+ FastV	58.1	0.597	6626	55.3	0.543	5887	65.6	0.681	7526	60.9	0.694	7520	79.6	0.218	2435	63.9	0.547	5999
	+ ToMe	59.8	0.649	6626	56.5	0.587	5888	**69.1**	0.740	7527	62.1	0.750	7521	79.5	0.240	2436	65.4	0.593	6000
	+ DyCoke	59.8	0.537	6878	55.9	0.488	6131	68.3	0.598	7798	62.1	0.607	7792	79.7	0.208	2631	65.2	0.488	6246
	+ STTM	**60.6**	0.601	6264	56.6	0.570	6022	68.4	0.735	7989	61.8	0.570	**5621**	79.7	0.240	2577	65.4	0.543	**5695**
	+ HoliTom	59.6	0.569	6511	55.6	0.519	6076	68.0	0.642	7566	60.7	0.653	7560	79.4	0.213	2387	64.7	0.519	6020
	+ Ours	**60.6**	**0.529**	**6190**	**56.8**	**0.482**	**5854**	69.0	**0.592**	**7488**	**62.3**	**0.566**	7482	**79.9**	**0.205**	**2368**	**65.7**	**0.475**	5876

Table 8. Comparison of training-free token reduction methods using LLaVA-Video-7B under 50% and 30% token budgets. The **best** result is in bold, second best underlined.

Token Budget	Method	VideoMME			LongVideoBench			MLVU			EgoSchema			NExT-QA			***Avg.***		
		Acc ↑	TTFT ↓	N_V ↓	Acc ↑	TTFT ↓	N_V ↓	Acc ↑	TTFT ↓	N_V ↓	Acc ↑	TTFT ↓	N_V ↓	Acc ↑	TTFT ↓	N_V ↓	Acc ↑	TTFT ↓	N_V ↓
100%	*LLaVA-Video 7B*	63.1	2.039	22086	59.6	1.805	19624	70.9	2.343	25088	58.7	2.312	25069	82.9	0.659	8116	67.0	1.832	19997
50%	+ FastV	61.0	1.034	11043	57.4	0.918	9812	68.3	1.164	12544	57.6	1.166	12535	82.4	0.353	4058	65.3	0.927	9998
	+ ToMe	61.4	1.043	11043	58.0	0.949	9812	69.7	1.192	12544	**58.7**	1.199	12535	82.6	0.370	4058	66.1	0.951	9998
	+ DyCoke	61.5	0.912	10555	58.1	0.820	9393	69.5	1.046	11978	58.6	1.049	11969	82.1	0.330	3957	66.0	0.831	9570
	+ STTM	62.6	1.021	10771	59.6	0.895	9183	69.9	1.152	12187	58.6	1.045	**10737**	82.5	0.322	**3452**	66.6	0.887	9266
	+ HoliTom	62.1	0.973	9642	59.2	0.872	9304	70.2	1.104	11821	57.7	1.105	11812	82.2	0.342	3824	66.3	0.879	9281
	+ Ours	**62.9**	**0.905**	**9520**	**59.8**	**0.815**	**9129**	**70.4**	**1.038**	**11760**	**58.7**	**1.041**	11751	**82.7**	**0.318**	3794	**66.9**	**0.823**	**9191**
30%	+ FastV	59.2	0.683	6626	54.7	0.610	5887	69.3	1.620	7562	56.9	0.771	7520	81.6	0.240	2435	64.3	0.785	6006
	+ ToMe	59.2	0.720	6626	56.3	0.658	5888	67.0	0.821	7527	57.4	0.834	7521	81.6	0.265	2436	64.3	0.660	6000
	+ DyCoke	60.7	0.598	6878	57.1	0.544	6131	67.3	**0.684**	7798	**58.3**	0.693	7792	81.5	0.229	2631	65.0	0.550	6246
	+ STTM	62.3	0.640	**5929**	57.0	0.616	**5702**	68.5	0.769	**7337**	58.0	0.773	**7285**	**82.0**	0.235	**2168**	65.6	0.607	**5684**
	+ HoliTom	61.3	0.641	6511	57.9	0.579	6076	68.1	0.734	7566	57.6	0.726	7560	81.5	0.235	2387	65.3	0.583	6020
	+ Ours	**62.5**	**0.592**	6190	**58.1**	**0.539**	5854	**69.4**	0.688	7488	58.1	**0.676**	7482	81.9	**0.226**	2368	**66.0**	**0.544**	5876

Table 9. Comparison using Qwen2VL-7B. The **best** result is in bold, second best underlined. **DyCoke*** denotes the DyCoke-stage1 setting.

Token Budget	Method	VideoMME			LongVideoBench			***Avg.***		
		Acc ↑	TTFT ↓	N_V ↓	Acc ↑	TTFT ↓	N_V ↓	Acc ↑	TTFT ↓	N_V ↓
100%	*Qwen2VL 7B*	61.8	10.745	74982	56.8	10.597	72109	59.3	10.671	73546
50%	+ ToMe	61.9	4.509	37491	56.7	4.348	36054	59.3	4.429	36773
	+ DyCoke*	62.6	4.166	**36057**	57.5	4.148	34735	60.1	4.157	35396
	+ STTM	**62.9**	4.761	39217	57.4	4.610	37315	60.2	4.686	38266
	+ Ours	62.7	**4.096**	36540	**57.8**	**4.092**	**34220**	**60.3**	**4.094**	**35380**
30%	+ ToMe	61.4	2.835	22496	54.7	2.600	21633	58.1	2.718	22065
	+ DyCoke*	61.8	2.710	23479	57.5	2.694	22649	59.7	2.702	23064
	+ STTM	62.4	2.766	23143	56.9	2.472	20022	59.7	2.619	21583
	+ Ours	**62.6**	**2.624**	**21980**	**57.6**	**2.430**	**19750**	**60.1**	**2.527**	**20865**

Table 10. Comparison using LLaVA-Video-72B. The **best** result is in bold, second best underlined. **DyCoke*** denotes the DyCoke-stage1 setting.

Token Budget	Method	VideoMME		
		Acc ↑	TTFT ↓	N_V
100%	*LLaVA-Video 72B*	70.5	17.698	22086
50%	+ ToMe	70.6	8.424	11043
	+ DyCoke*	70.4	8.052	10555
	+ STTM	71.4	7.821	10082
	+ Ours	**71.5**	**7.420**	**9520**
30%	+ ToMe	68.5	5.186	6626
	+ DyCoke*	69.3	5.353	6878
	+ STTM	**69.9**	5.405	6897
	+ Ours	**69.9**	**5.022**	**6190**

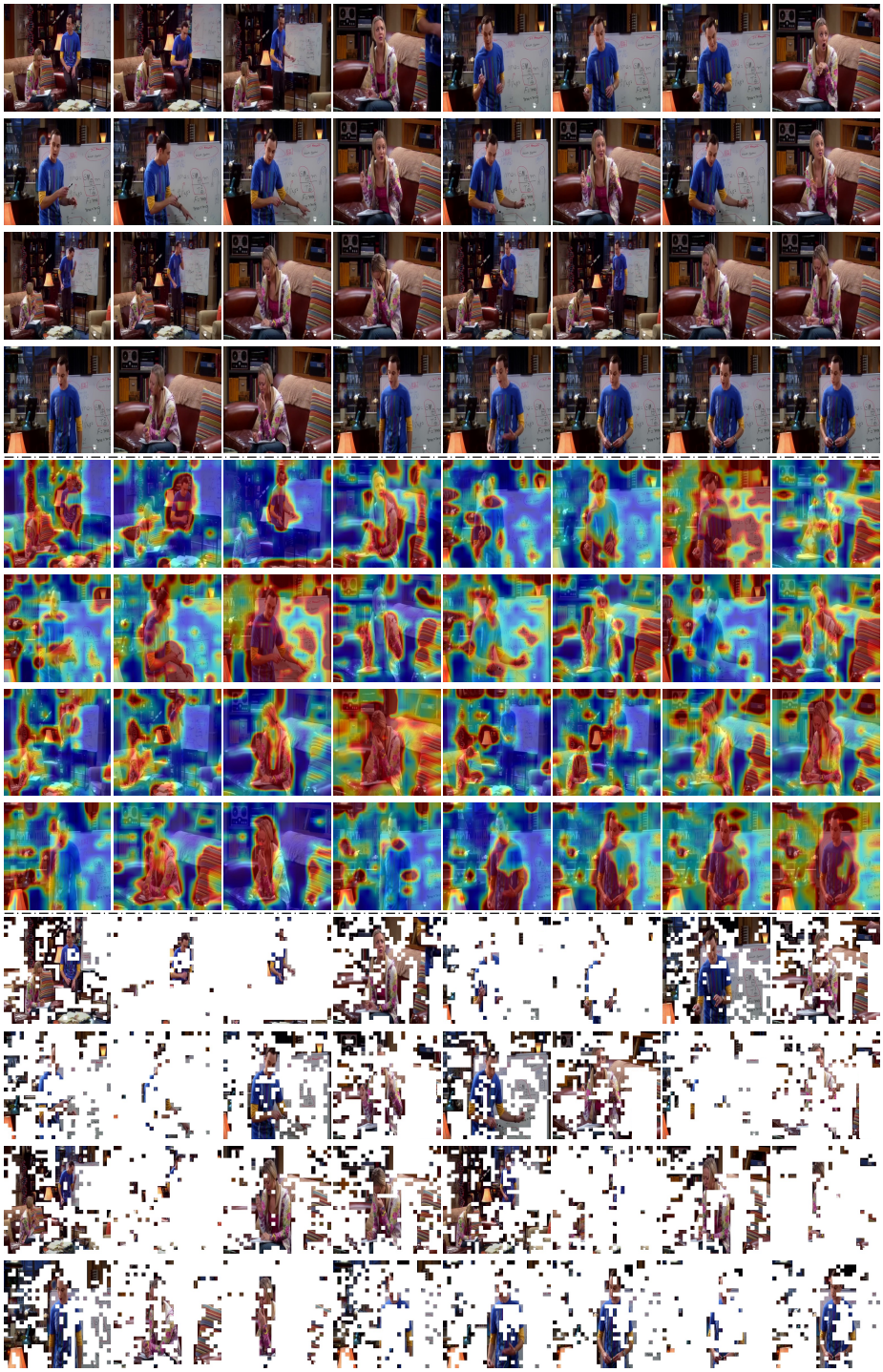

Figure 7. Visualization of token merging. Top: Frames of the input video. Middle: Residual energy maps derived from metadata. Bottom: Video frames after token merging, where merged tokens are visualized as blank blocks.

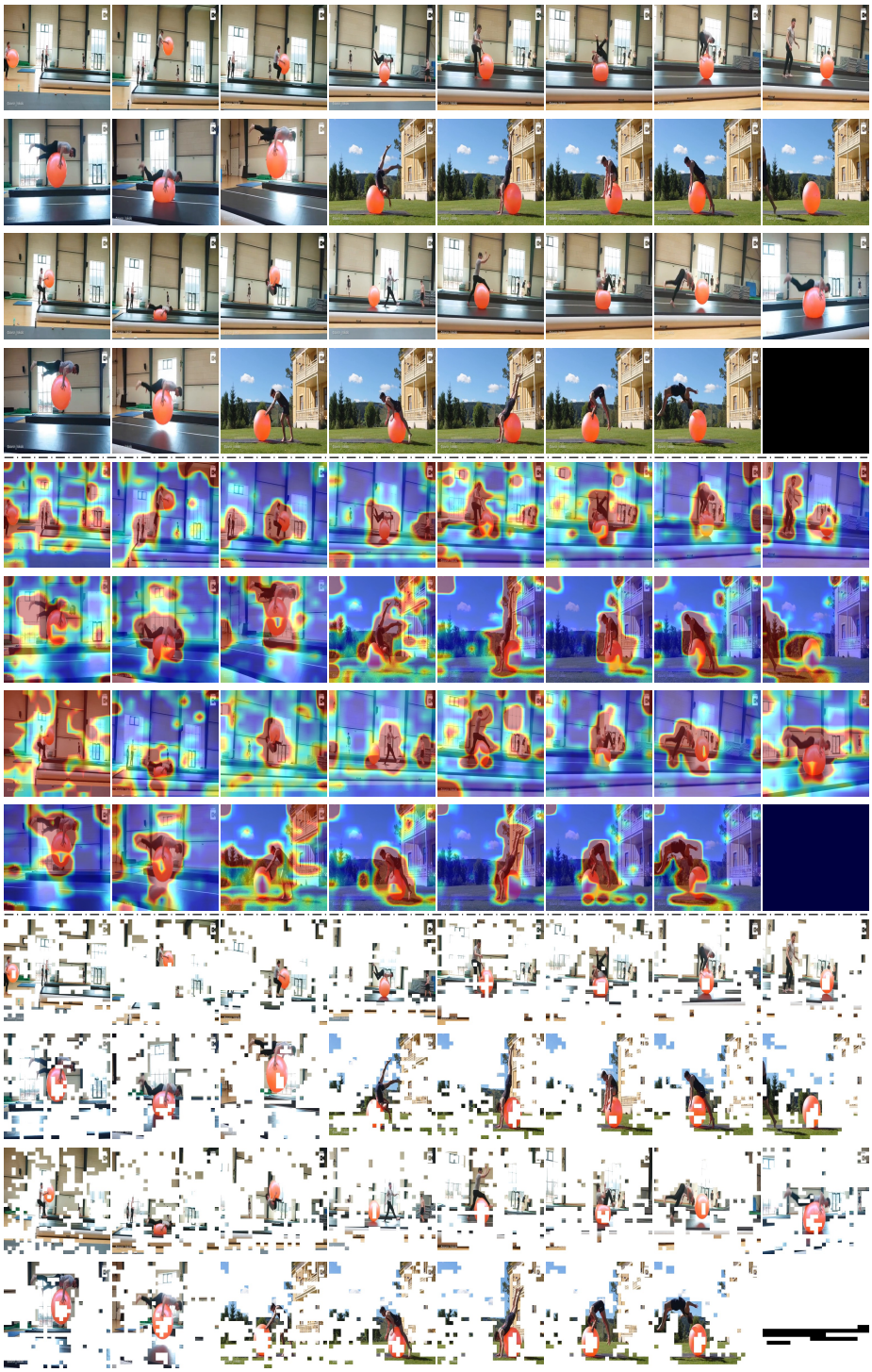

Figure 8. Visualization of token merging. Top: Frames of the input video. Middle: Residual energy maps derived from metadata. Bottom: Video frames after token merging, where merged tokens are visualized as blank blocks.