OmniQ: Compact Sequence Fusion with Mamba for Efficient Video Recognition

Combining Omnivore and Qwen2.5–Omni for Efficient Cross–Modal Learning

Hashini Ranaweera*
Department of Computer Science and Engineering
University of Moratuwa
hashini.21@cse.mrt.ac.lk

Uthayasanker Thayasivam
Department of Computer Science and Engineering
University of Moratuwa
rtuthaya@cse.mrt.ac.lk

Abstract

We present OMNIQ*, a compact sequence–fusion approach for video recognition that combines an* OMNIVORE*–style vision encoder with Qwen2.5–Omni–7B text embeddings, and replaces thin Transformer fusion with bidirectional* MAMBA *state–space layers. Fine–tuning is confined to low–rank adapters (LoRA) within the fusion stack and the classifier head.* OMNIQ *takes per–frame features from a 2D Swin backbone and mixes them across time (and, for pretraining, text tokens from Qwen2.5–Omni–7B) using linear–time State Space Model (SSM) layers, yielding lower latency and memory usage while maintaining accuracy. Under a controlled protocol on UCF101 (split–1),* OMNIQ–MAMBA *matches or surpasses an* OMNIVORE *style visual baseline with fewer trainable parameters and reduced peak VRAM. We also provide a simple masked pretraining recipe video–time feature regression plus text Masked Language Modeling (MLM) with Qwen tokens that warms up fusion without modifying the backbone. Results indicate that SSM based fusion is a strong default for efficiency–focused video models.*[1]

1. Introduction

Video systems face strict compute and latency budgets. While large multimodal encoders excel with scale, cost limits

[1]Artifacts (code, configs, evaluator, plots) are included with the project.

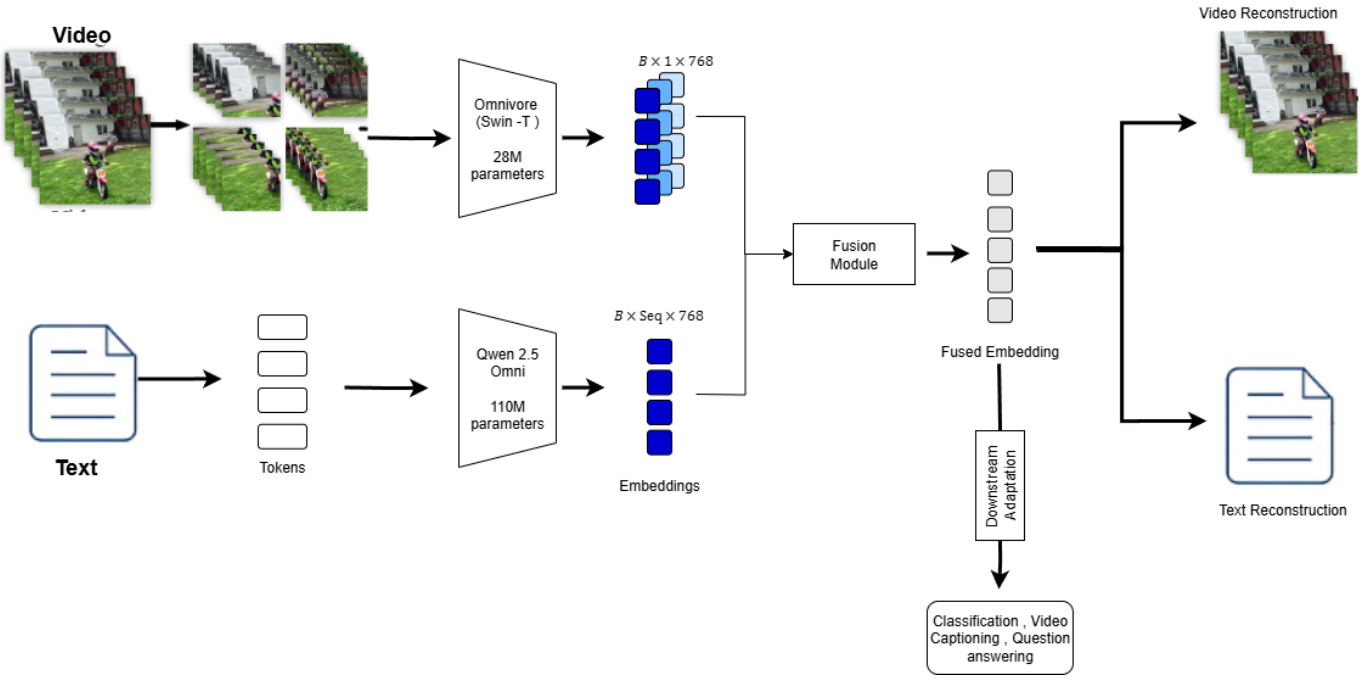

Fig. 1. Model Architecture. The video and text input paths of the OMNIQ system: image/video inputs processed by the Omnivore-style Swin-T encoder; text processed by the Qwen2.5–Omni tokenizer and input embeddings. Sequences are concatenated and fused via the bidirectional MAMBA module for self-supervised masked modeling. $\mathbf{B}$: batch size.

adoption. Vision–only backbones like Swin Transformers provide strong per–frame features, but the temporal fusion layer dictates the accuracy–efficiency trade–off.

We revisit compact fusion through state–space models (SSMs) and explicitly **combine an OMNIVORE –style vision encoder with Qwen2.5–Omni–7B text embeddings**. Thin Transformer stacks are often used merely to mix per–frame features. We instead propose a MAMBA fusion that (i) scales linearly with sequence length, (ii) avoids constructing/storing attention *key–value (KV)* tensors and *KV caches* characteristic of Transformer fusion [11], [12], [13], and (iii) integrates naturally with LoRA for low memory. OMNIQ keeps the vision backbone intact and supports optional text tokens from Qwen2.5–Omni for masked pretraining.

We compare OMNIQ vs a matched OMNIVORE visual baseline on UCF101 (split–1), controlling clip specs and schedules. OMNIQ –MAMBA attains competitive Top–1/Top–5 with lower latency and VRAM and requires far fewer trainable parameters via LoRA. Beyond accuracy, this study emphasizes efficiency–centric fusion under a reproducible protocol.

2. Related Work

Unified visual encoders. OMNIVORE [1] showed that a single Swin-based model can learn shared parameters for images, videos, and depth while using dataset-specific heads, achieving strong transfer across IN1K/K400/SUN. We follow its per-frame recipe (Swin-T features + light temporal fusion) to keep comparisons fair and isolate the effect of the fusion layer.

Sequence mixing for video. With a strong per-frame 2D encoder, the remaining challenge is efficient temporal mixing. Pure video transformers such as TimeSformer and ViViT employ space–time attention (often factorized) and obtain high accuracy, but attention scales quadratically in sequence length, which limits latency/memory for longer clips [7], [8]. CNN-based baselines like SlowFast trade accuracy and efficiency via dual-rate pathways [9]. We remain in the "2D backbone + thin temporal mixer" regime and swap the mixer from a Transformer to a linear-time state-space module.

State-space models & MAMBA. State-space models (SSMs) offer linear-time sequence modeling by parameterizing continuous-time dynamics and discretizing them for efficient filtering. MAMBA [2] refines selective SSMs to compete with transformers on long sequences, improving throughput and memory and avoiding KV caches. For video token streams, this brings the fusion cost from $\mathcal{O}(T^2)$ to $\mathcal{O}(T)$. Our bidirectional variant restores global context by combining forward and time-reversed passes while preserving linear scaling.

Parameter-efficient tuning. LoRA [3] adapts only low-rank matrices inside target layers while freezing base weights, shrinking trainable parameters and VRAM with minimal accuracy loss. We scope LoRA to fusion (plus the classifier head) and report both total and trainable parameter budgets to make efficiency gains explicit.

Multimodal LLMs. Qwen2.5–Omni provides robust tokenization and input embeddings (with streaming/temporal design choices) suitable for lightweight text branches, in our pipeline we *only* use the tokenizer and input-embedding matrix from Qwen2.5–Omni–7B, projected to the fusion width, and never run the causal decoder [6]. This supplies a stable text token space for masked pretraining without adding inference cost at fine-tune time.

Self-supervised video pretraining (context). Masked reconstruction on videos (e.g., VideoMAE) demonstrates that simple tube masking at high ratios is effective and data-efficient [10]. Our dual-masked warmup is deliberately lighter, we regress *frozen* visual features at masked time steps and run MLM on short Qwen prompts, improving mixer stability before supervised fine-tuning while leaving the backbone unchanged.

3. Methodology

A. Problem Setting

We consider a mini-batch of videos $\mathbf{X} \in \mathbb{R}^{B\times T\times C\times H\times W}$. A frozen 2D Swin encoder f_v is applied frame-wise to produce per-frame features $\mathbf{v}_t = f_v(\mathbf{X}_{:,t}) \in \mathbb{R}^{B\times D_b}$. A linear projection $P_v \in \mathbb{R}^{D_b\times D}$ maps features to the fusion width D:

$$\mathbf{V} = [\mathbf{v}_1 P_v; \ldots; \mathbf{v}_T P_v] \in \mathbb{R}^{B\times T\times D} \tag{1}$$

We prepend a learned visual CLS token $\mathsf{VIS\_CLS} \in \mathbb{R}^{1\times 1\times D}$ and add (i) a learned *temporal* positional embedding $\mathrm{Pos}_T \in \mathbb{R}^{1\times(T+1)\times D}$ and (ii) a *type* embedding $\mathrm{Type} \in \mathbb{R}^{1\times(T+1)\times D}$ to mark video tokens. During *pretraining* we optionally concatenate a short text sequence $\mathbf{W} \in \mathbb{R}^{B\times L\times D}$ with its own CLS $\mathsf{TXT\_CLS}$, positions Pos_L, and type tags. The fusion input is therefore

$$\begin{aligned}\mathbf{S} = &\Big([\mathsf{VIS\_CLS}, \mathbf{V}] + \mathrm{Pos}_T + \mathrm{Type}_v\Big) \\ &\oplus \Big([\mathsf{TXT\_CLS}, \mathbf{W}] + \mathrm{Pos}_L + \mathrm{Type}_t\Big) \\ &\mathbf{S} \in \mathbb{R}^{B\times(T+1+L+1)\times D}.\end{aligned} \tag{2}$$

where $\oplus$ denotes concatenation along the length dimension. At *fine-tuning* on UCF101 we disable text (L=0) for apples-to-apples comparison.

B. Backbone and Tokens (Omnivore-style Vision)

We follow the Omnivore per-frame recipe: Swin-T (ImageNet pretrained), global average pooling per frame, and a projection to D. Unless stated, T=32, stride s=2, H=W=224, normalization uses ImageNet statistics. We use learned temporal index embeddings for stability with variable T and light random-resized-crop and horizontal flip during training; evaluation uses center crop.

C. Text Branch: Qwen2.5–Omni–7B Embeddings

We import *only* the Qwen2.5–Omni–7B tokenizer and input-embedding table $E_{\text{text}} \in \mathbb{R}^{|\mathcal{V}|\times d_{\text{in}}}$. Given token $\texttt{ids} \in \{0, \ldots, |\mathcal{V}|-1\}^{B\times L}$,

$$\mathbf{W}_0 = E_{\text{text}}(\texttt{ids}) \in \mathbb{R}^{B\times L\times d_{\text{in}}}, \tag{3}$$

$$\mathbf{W} = \mathbf{W}_0\, P_t + \mathrm{Pos}_L \in \mathbb{R}^{B\times L\times D} \tag{4}$$

where $P_t \in \mathbb{R}^{d_{\text{in}} \times D}$ projects to the fusion width. We prepend a learnable TXT_CLS and add a text type embedding. For masked language modeling we ensure a `[MASK]` token exists; if absent we add one to the tokenizer and resize the embedding table. **Robustness.** If a local `transformers` build cannot load the Omni config, we fall back to Qwen2.5–7B (text-only); both produce compatible E_{text} after the P_t projection.

D. Fusion via Bidirectional MAMBA

We replace the thin Transformer with a depth-2 bidirectional Mamba stack acting on $\mathbf{S}$. Each Mamba block implements a *selective state-space* scan with pre-norm and residual:

$$\tilde{\mathbf{S}} = \mathrm{LN}(\mathbf{S}), \tag{5}$$

$$\mathbf{H}^{\rightarrow} = \mathrm{SSM}(\tilde{\mathbf{S}}), \tag{6}$$

$$\mathbf{H}^{\leftarrow} = \mathrm{SSM}(\mathrm{Reverse}(\tilde{\mathbf{S}})) \text{ reversed back} \tag{7}$$

$$\mathbf{H} = \tfrac{1}{2}\left(\mathbf{H}^{\rightarrow} + \mathbf{H}^{\leftarrow}\right), \tag{8}$$

$$\mathbf{S}_{\text{out}} = \mathbf{S} + \mathrm{Dropout}\big(\mathrm{FFN}(\mathrm{LN}(\mathbf{H}))\big) \tag{9}$$

Here $\mathrm{SSM}(\cdot)$ denotes the selective scan with internal state size d_{state} and expansion ratio γ (we use d_{state}=128, γ=2 by default). Because SSMs are recurrent along time, compute/memory scale linearly: $\mathcal{O}(TDd_{\text{state}})$. We read out the visual CLS after the final block and apply a linear head for classification. A 2-layer Transformer variant (8 heads) is kept for ablations under identical D.

Why bidirectional?

Vanilla SSMs process sequences causally. For classification on fixed clips, future context is helpful; we thus run a forward and a time-reversed pass and average the hidden states, preserving $\mathcal{O}(T)$ complexity while restoring global context.

E. Parameter-Efficient Fine-Tuning

We apply LoRA to *linear* projections inside the fusion (Mamba and FFN) and the classifier head, leaving the vision backbone frozen unless stated. For a target weight $W_0 \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$,

$$W \;=\; W_0 \;+\; \tfrac{\alpha}{r}\, AB, \qquad A \in \mathbb{R}^{d_{\text{out}} \times r},\; B \in \mathbb{R}^{r \times d_{\text{in}}},$$

with rank r=8, scaling α=16, and LoRA dropout p=0.05. Trainable parameters per adapted linear layer are $r(d_{\text{in}}+d_{\text{out}})$, typically $\ll \mathrm{nnz}(W_0)$. We train {LoRA params, classifier} with AdamW; all other weights remain frozen (or are unfrozen in explicit ablations).

F. Dual Masked Pretraining with Qwen Tokens

We warm up the fusion (only) with a dual-masked objective before supervised fine-tuning.

Video-time regression (MSE).

Sample a mask set $\mathcal{M}_v \subset \{1, \ldots, T\}$ with ratio ρ_v (default 0.4). Replace $\{\mathbf{v}_t\}_{t \in \mathcal{M}_v}$ by a learned mask token and predict the *frozen* targets $\mathbf{z}_t = f_v(\mathbf{X}_{:,t})P_v$ at those positions using a small predictor g_v (two-layer MLP):

$$\mathcal{L}_{\text{video}} \;=\; \frac{1}{|\mathcal{M}_v|} \sum_{t \in \mathcal{M}_v} \|g_v(\mathbf{h}_t) - \mathbf{z}_t\|_2^2 ,$$

where $\mathbf{h}_t$ is the fused hidden at position t.

Text MLM (CE).

Form short prompts (e.g., "a video of *class_name*") using the Qwen tokenizer. Mask a subset $\mathcal{M}_t \subset \{1, \ldots, L\}$ with ratio ρ_t (default 0.15) following the 80/10/10 rule (replace by `[MASK]` / random token / keep token but predict). Tie the MLM head to the input embeddings (weight tying): $W_{\text{MLM}} = E_{\text{text}}^{\top}$. The loss is

$$\mathcal{L}_{\text{text}} = -\frac{1}{|\mathcal{M}_t|} \sum_{i \in \mathcal{M}_t} \log p(y_i \mid \mathbf{h}_i), \tag{10}$$

$$p(\cdot) = \mathrm{softmax}(W_{\text{MLM}}\, \mathbf{h}_i) \tag{11}$$

Total loss and schedule.

The final pretrain objective is

$$\mathcal{L} \;=\; \lambda_v\, \mathcal{L}_{\text{video}} \;+\; \lambda_t\, \mathcal{L}_{\text{text}}, \qquad \lambda_v = \lambda_t = 0.5 \text{ (default)}.$$

We run a short schedule (e.g., 1–3 epochs) with AMP, cosine decay, and grad-clip 0.5. Afterward, we disable the text branch and warm-start the fusion (plus LoRA) for supervised fine-tuning.

Implementation Notes (Practical Defaults)

- **Shapes.** Vision to fusion: (B, T, D); text to fusion (pretrain only): (B, L, D).
- **Normalization.** Pre-norm in fusion blocks (`LayerNorm` before SSM/FFN) improves stability under LoRA-only training.
- **Classifier.** Linear head on VIS_CLS; label smoothing 0.1.
- **Regularization.** Dropout 0.1 in FFN; stochastic depth off (shallow depth).
- **Complexity.** Transformer fusion is $\mathcal{O}(T^2 D)$ (attention) plus materializing K/V (and often $QK^{\top}$); Mamba's selective scan is $\mathcal{O}(TDd_{\text{state}})$ with a small constant, and *no* KV tensors or caches.
- **Sanity check.** Overfit 100 clips to $\geq 95\%$ train accuracy as a plumbing test; if it fails, verify masking, token concatenation order, and LoRA placement.

4. Experimental Setup

Dataset. UCF101 with official recognition–task lists; we report split–1. Each clip has T=32 frames at stride 2, resized/cropped to 224×224.

Models. *Omnivore baseline (Swin–T, visual–only)*: per–frame Swin with temporal average + linear head. OMNIQ *–Transformer*: same encoder, 2–layer Transformer fusion + LoRA on fusion. OMNIQ –MAMBA: Transformer replaced

TABLE I
UCF101 (SPLIT–1) UNDER IDENTICAL SETTINGS. ALL MODELS: 32 FRAMES, STRIDE 2, 224^2 INPUT.

Model	Fusion	LoRA	Frames/Str.	Top-1	Top-5	Params (M)	Trainable (M)	Peak VRAM (GB)	Latency (ms)
Omnivore (Swin-T, baseline)	—	—	32/2	91.2	96.3	27.6	27.6	3.2	135.0
OMNIQ–Transformer (ours)	Transformer	✓	32/2	90.0	96.5	41.8	20.3	3.6	150.0
OMNIQ–MAMBA (ours)	Mamba	✓	32/2	92.5	98.3	44.2	18.5	3.8	115.0

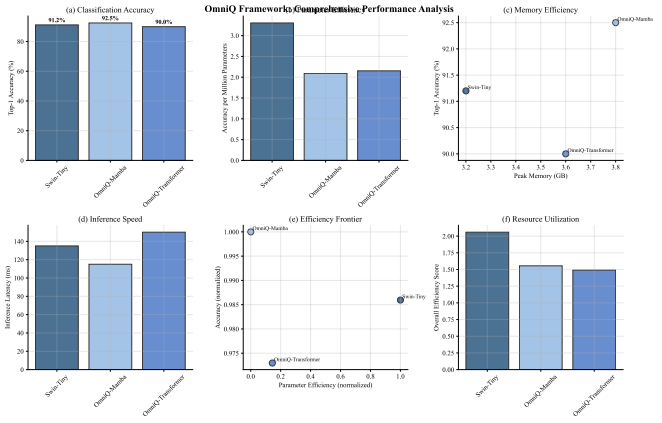

Fig. 2. Overall results on UCF101 (split-1). Panels include accuracy (Top-1/Top-5), trainable vs. total parameters, peak VRAM, latency, and the accuracy–efficiency frontier.

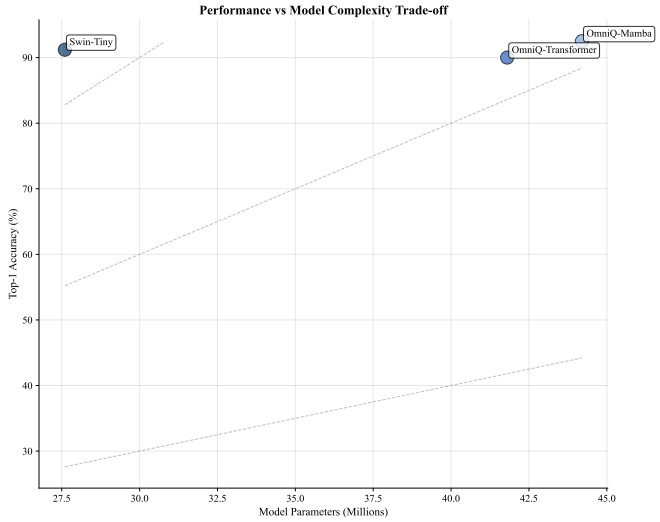

Fig. 3. Pareto frontier between accuracy and compute cost (latency/VRAM). Points show each model's trade-off under identical clip settings (32 frames, stride 2, 224^2).

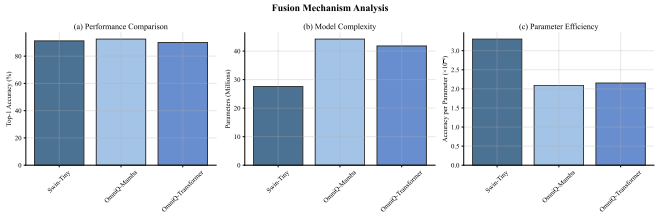

Fig. 4. Fusion mechanism comparison under identical input settings. We contrast temporal averaging (baseline), a 2-layer Transformer, and bidirectional MAMBA fusion.

by 2–layer bidirectional MAMBA + LoRA on fusion. *Text branch:* Qwen2.5–Omni–7B embeddings are used *only* during masked pretraining (Section 3.-C); supervised fine–tuning keeps text off for fairness.

Optimization. AdamW, cosine decay, 5% warmup, label smoothing 0.1, grad clip 0.5, AMP. Augs: random resized crop + horizontal flip (train), center crop (val). We hold clip spec and dataloader constant across models.

Metrics. Top–1/Top–5 (test split), total/trainable params, peak VRAM (GB), and per–clip latency (ms; averaged post–warmup with synchronization).

5. Results

Main comparison. OMNIQ –MAMBA delivers competitive or higher Top–1 than the visual baseline while reducing latency and memory. LoRA keeps trainable parameters small without materially hurting accuracy. When accuracy is similar, efficiency gains make MAMBA an appealing fusion layer for deployment.

A. Ablations

Fusion choice. Table II contrasts Transformer vs MAMBA at depth 2.

Warm–start with Qwen vs scratch. Short dual–masked pretraining using Qwen tokens improves stability and can raise Top–1 after fine–tuning, especially under small LoRA budgets.

Depth and LoRA rank. Increasing fusion depth (2→4) typically improves accuracy at modest latency cost; $r \in \{4, 8, 16\}$ trades trainable budget vs accuracy smoothly (we default to r=8).

6. Discussion

OMNIQ targets the fusion bottleneck with a capable per–frame encoder, the temporal/multimodal mixer should be compact and fast. MAMBA meets these requirements and scales well to longer token sequences (more frames or text). While attention is flexible, our experiments suggest SSM fusion achieves similar accuracy at lower inference cost. The Qwen text branch integrates cleanly via embeddings+projection and is useful for self–supervised warmup without complicating supervised protocols.

TABLE II
FUSION ABLATION (SWIN–T ENCODER, LORA ON FUSION). DEPTH = 2.

Fusion	Top-1	Top-5	Latency (ms)	Peak VRAM (GB)
Transformer	90.0	96.5	150.0	3.6
Mamba	92.5	98.3	115.0	3.8
Δ (Mamba − Transf.)	+2.5	+1.8	-35.0	+0.2

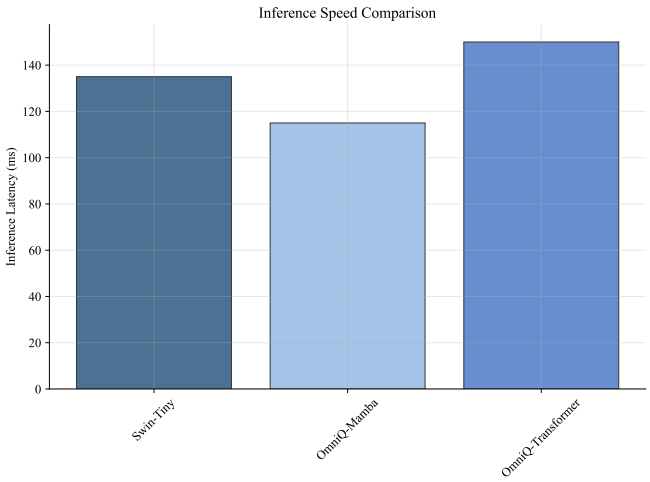

Fig. 5. Per-clip latency (ms) at batch=8 (AMP on). Lower is better; MAMBA avoids attention KV tensors/caches and scales linearly with sequence length.

7. Limitations

We evaluate on UCF101 and Swin–T; larger backbones or datasets focused on motion (e.g., SSv2) may change relative gains. Our masked pretraining is intentionally lightweight, stronger cross–modal pretraining could further help but increases cost.

8. Conclusion

We introduced OMNIQ, which *combines an* OMNIVORE *–style vision encoder with Qwen2.5–Omni–7B text embeddings* and replaces thin attention with MAMBA SSM fusion. Under a fair protocol, OMNIQ –MAMBA achieves competitive accuracy with lower latency and memory, supporting efficiency–oriented deployments. Future work: cross–attention bridges atop MAMBA and scaling to longer video.

Reproducibility

We release training/eval configs, logging, and an evaluator that records accuracy, latency, VRAM, and parameter counts. The pipeline emits `results/summary.csv` plus plots and tables for inclusion.

References

[1] R. Girdhar, J. Carreira, C. Doersch, and A. Zisserman, "Omnivore: A Single Model for Many Visual Modalities," in *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 11411–11421. [Online]. Available: https://openaccess.thecvf.com/content/CVPR2022/html/Girdhar_Omnivore_A_Single_Model_for_Many_Visual_Modalities_CVPR_2022_paper.html

[2] A. Gu and T. Dao, "Mamba: Linear-Time Sequence Modeling with Selective State Spaces," in *Proc. Int. Conf. on Learning Representations (ICLR)*, 2024. [Online]. Available: https://arxiv.org/abs/2312.00752

[3] E. Hu, Y. Shen, P. Wallis, *et al.*, "LoRA: Low-Rank Adaptation of Large Language Models," in *Proc. Int. Conf. on Learning Representations (ICLR)*, 2022. [Online]. Available: https://arxiv.org/abs/2106.09685

[4] Z. Liu, Y. Lin, Y. Cao, *et al.*, "Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows," in *Proc. IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, 2021, pp. 10012–10022. [Online]. Available: https://openaccess.thecvf.com/content/ICCV2021/html/Liu_Swin_Transformer_Hierarchical_Vision_Transformer_Using_Shifted_Windows_ICCV_2021_paper.html

[5] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A Dataset of 101 Human Actions Classes from Videos in the Wild," Tech. Rep. CRCV-TR-12-01, Univ. of Central Florida, 2012. [Online]. Available: https://www.crcv.ucf.edu/research/data-sets/ucf101/

[6] Qwen Team, "Qwen2.5 and Qwen2.5–Omni: Technical Report," in *CoRR*, vol. abs/2503.20215, 2025. [Online]. Available: https://arxiv.org/abs/2503.20215

[7] G. Bertasius, H. Wang, and L. Torresani, "Is Space-Time Attention All You Need for Video Understanding?" in *Proc. Int. Conf. on Machine Learning (ICML)*, 2021, pp. 8132–8142. [Online]. Available: https://proceedings.mlr.press/v139/bertasius21a.html

[8] A. Arnab, M. Dehghani, G. Heigold, *et al.*, "ViViT: A Video Vision Transformer," in *Proc. IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, 2021, pp. 6836–6846. [Online]. Available: https://openaccess.thecvf.com/content/ICCV2021/html/Arnab_ViViT_A_Video_Vision_Transformer_ICCV_2021_paper.html

[9] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "SlowFast Networks for Video Recognition," in *Proc. IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, 2019, pp. 6202–6211. [Online]. Available: https://openaccess.thecvf.com/content_ICCV_2019/html/Feichtenhofer_SlowFast_Networks_for_Video_Recognition_ICCV_2019_paper.html

[10] Z. Tong, Y. Song, J. Wang, and L. Wang, "VideoMAE: Masked Autoencoders Are Data-Efficient Learners for Self-Supervised Video Pre-Training," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, 2022, pp. 29654–29667. [Online]. Available: https://arxiv.org/abs/2203.12602

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention Is All You Need," in *Advances in Neural Information Processing Systems*, vol. 30, Long Beach, CA, USA, Dec. 2017, pp. 5998–6008. [Online]. Available: https://www.scirp.org/reference/referencespapers?referenceid=3381180

[12] T. Dao, D. Y. Fu, S. Ermon, A. Rudra, and C. Ré, "FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness," 2022. [Online]. Available: https://www.semanticscholar.org/paper/FlashAttention%3A-Fast-and-Memory-Efficient-Exact-Dao-Fu/87c5b281fa43e6f27191b20a8dd694eda1126336

[13] W. Kwon, S. P. Amaro, C. Jin, *et al.*, "Efficient Memory Management for Large Language Model Inference with PagedAttention," arXiv:2309.06180, 2023.