Slow-Fast Weights for Long-Term Video Understanding

David Gardner[1] Rajat Modi[2] Yogesh Singh[2]

[1]Boston University [2]University of Central Florida

https://github.com/davidg335/slow_fast_long_video

July 31, 2025

Abstract

Understanding long videos remains a significant challenge due to their length, complexity, and high computational cost. Current state-of-the-art approaches rely on fixed-length memory banks that compress frame features but risk losing important temporal information. In this work, we introduce a learned fast weight memory bank inspired by biological synaptic plasticity and associative memory. Our approach augments the MA-LMM video-language model by replacing hand-crafted compression rules with a lightweight neural module that dynamically encodes temporal features in its weights. These learned weights are passed to a Q-Former module via cross-attention, enabling more flexible and expressive memory representations. While performance trails the original MA-LMM in preliminary experiments, our method demonstrates the feasibility of using fast weights for scalable video memory and opens promising directions for future research in long-term temporal modeling.

1 Introduction

Long video understanding has several applications, including detecting anomalies in hours of CCTV camera footage and summarizing long sports matches. However, processing long videos is a task that machine learning models struggle to do efficiently. Long videos typically contain complicated actions and dynamics. Additionally, long videos have a large quantity of information: a 60-second color video at 10 fps standard definition contains half a million bits. With transformers scaling quadratically to input size, taking a whole video as a transformer input is not feasible.

State-of-the-art long-term memory models are auto-regressive and take advantage of a fixed-length memory bank to store temporal information [4] [3] [1]. In MA-LMM [4], a nonparametric cosine similarity rule is applied to the most similar frames when the memory bank reaches capacity and a new frame enters. HierarQ [1] uses both first-in-first-out and cosine similarity as a compression rule. These compression techniques have the potential to lose frame information. Additionally, there is no little evidence that our brain stores neural activation states [8]. Instead, the brain uses several plasticity mechanisms that are synapse-specific, similar to the weights in a neural network [8].

To combat the limitations of current fixed memory banks, we propose a learned, fast weight memory bank. This memory bank consists of a lightweight neural network that fits sequentially to video frames. Instead of the current hand-designed compression techniques, the fast weight memory bank compresses frame information by training on future frames. The weights of the memory bank are passed to the temporal feature extraction section of the architecture instead of activation vectors.

The idea of fast weights has not been implemented in the context of long videos. This report acts as an initial exploration into this topic by addressing setbacks of current memory banks.

2 Related Work

The idea of slow-fast weights originates in Hinton's 1987 paper [5] where he introduces the idea of storing information in weights rather than activation layers of a neural network. Slow weights contain long term knowledge of a system and are the weights typically used in feed-forward neural networks [8]. Fast weights adjust quickly to new information and decay quickly as well. Hinton ties the fast-slow weight idea to the concept of associative memory in neuroscience, where probing the brain with one idea triggers the recall of similar events. For instance, mentioning a person's name, one could remember where they are from or their birthday. Associative memory is promising because

1. memory in weights has more storage capacity than in activation vectors. In a fully connected neural network with layers with n dimensional activation vectors, there are n^2 weight values. Also,

2. associative memory allows for more flexible memory compression than with fixed external memory banks.

Associative memory has been implemented several times. The Hopfield network [6] allows storing patterns in a complete graph using an energy function and supports pattern retrieval from noisy inputs. However, Hopfield networks have limited capacity: a network with N nodes can hold up to $0.15N$ binary vectors of length $\log(N)$ each. In 1987, Hinton and Plaut [5] show how in a simple neural network, fast weights can act as a de-blurring mechanism where a noisy input can be associated with stored input-output pairs. Fast Associative Memory [8] implements a fast weight matrix into an RNN. This fast weight matrix acts as a continuously updating memory bank that decays exponentially, much like how leftover $[Ca^{2+}]$ ions in the axon terminal cause facilitation, where a neuron is more likely to fire again if it was recently fired [14].

3 Method

We introduce a model that incorporates a learned memory bank into the MA-LMM long-term video understanding model. The key architecture change is the introduction of a fast weight memory bank that is fed as a cross-attention layer to the lightweight trainable Querying Transformer (Q-Former) [10].

3.1 Fast Weight Memory Bank

The fast weight memory bank fits to the output of the visual encoder. We use Asynchronous Perceptron Machine (APM) [11] as our memory bank model because of its lightweight architecture and ability to encode semantic information in its weights. As seen in 1, the model architecture is as follows. For each 14×14 pixel image patch location (x, y) for each frame timestep t, we produce a feature representation I from a single layer CNN filter, an average pixel representation I_{xyt} from a pooling filter, and a fixed 3-D positional embedding p_{xyt} of the patch and timestep. The concatenated three items $[I, I_{xyt}, p_{xyt}]$ are then passed to a seven-layer multilayer perceptron. The output of APM is a feature vector $\hat{f}_{xyt}$ that trains on the output of the visual encoder f_{xyt} with the mean square error loss

$$\mathcal{L}_{APM} = \frac{1}{n} \sum_{\{x,y,t\} \in S} \left(\hat{f}_{xyt} - f_{xyt} \right)^2,$$

where S is a subset of a batch of frames. We choose a batch size of 16 and S is of size 2000, where we sample randomly from the 4096 patches. Each frame is $224 \times 224 \times 3$ pixels, so each frame has 256 patches. [11]

3.2 Temporal Feature Extraction

We use the same architecture as the BLIP-2 Q-Former[10]. Let C be the hidden dimension of the query tokens, N be the number of tokens per query q, and T_t be the number of queries or visual features in the fixed memory banks. A trainable hidden query $q \in R^{N \times C}$ learns a semantic feature representation of the whole video auto-regressively by being passed

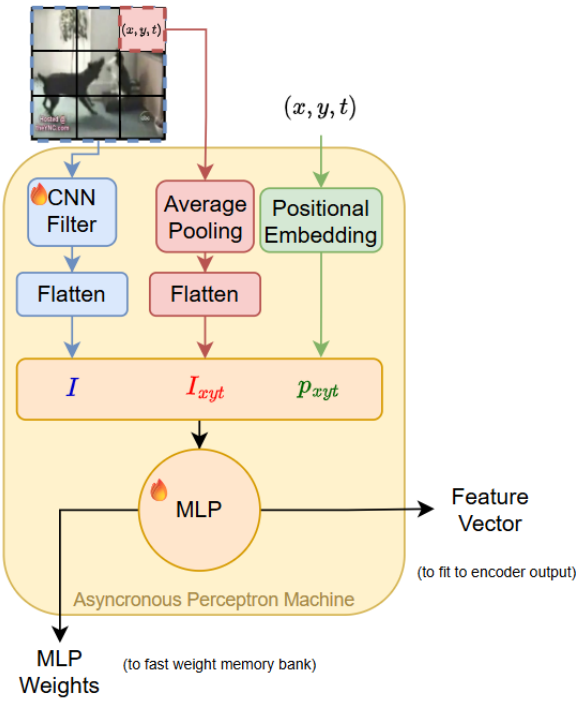


Figure 1: A simplified diagram of the APM model [11]. The model learns associations of image patches with their respective embedding outputs from the visual encoder. Weights from a multilayer perceptron are passed to the fast weight memory bank then to a cross attention layer in the Q-former.

through 12 attention layers. Every layer contains a cross-attention module with a fixed query memory bank, where the module incorporates long-term information from past frames. Every other Q-Former layer contains two cross-attention blocks. From the MA-LMM paper, the fixed visual memory bank incorporates episodic, short-term frame information. Our novel fast weight memory bank incorporates a similar, less lossy short-term memory. Note that the query memory bank is unique to each Q-Former layer, but the visual memory banks are the same across Q-Former layers.

After processing the whole video, the final hidden query q containing the semantic information from the whole video is passed to a fully connected layer, then to an LLM along with the task prompt. The LLM output is the model's response to the task prompt.

The loss for the whole model is the standard cross-entropy loss

$$\mathcal{L} = -\frac{1}{S}\sum_{i=1}^{S} \log P(w_i \mid w_{<i}, V)$$

where V represents the input video, and w_i is the i-th ground-truth text token.

3.3 Memory Compression

The fixed visual and query memory banks both use the compression rule from the MA-LMM [4]. The memory bank is of fixed size M. For each video, the memory bank is initially empty. Frames are auto-regressively concatenated to the memory bank until we hit frame f_{M+1}. The memory bank is now $[f_1, ..., f_{M+1}]$ and we find the two temporally adjacent patches with the highest similarity with

$$s_t^i = \cos(f_i^t, f_i^{t+1}), \quad t \in [1, M],\ i \in [1, P].$$

We then the frame $\hat{t}$ from the largest s_t^i and average frames $f_{\hat{t}}$ and $f_{\hat{t}+1}$. This process continues for frames past timestep M.

3.4 Training and Inference Pipeline

During the training stage, we first pass the whole batch of B videos through the visual encoder and

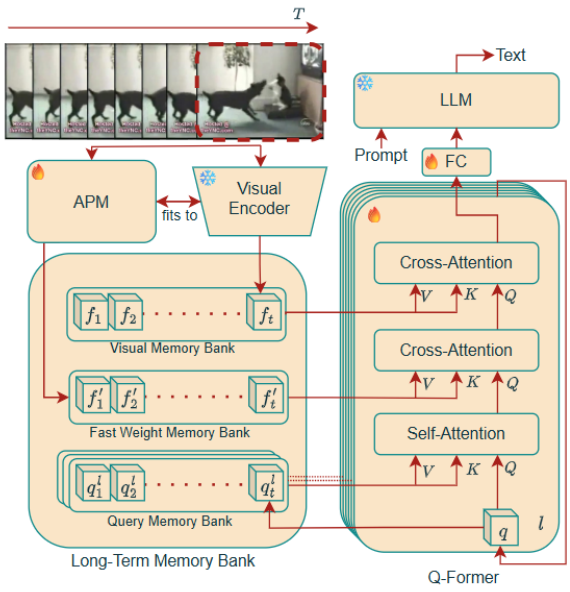


Figure 2: A diagram displaying the model architecture. Frames are auto-regressively passed to a vision encoder, where a fast weight memory bank model fits to the output of the encoder. The fast weight memory bank is passed via cross-attention to the Q-Former, along with learnable queries and the original fixed visual memory bank. The final hidden state q is passed to an LLM, where the task prompt is asked.This model is adapted from MA-LMM [4].

train the APM on the batch. We then freeze the APM weights and pass the APM MLP weights to our fast weight memory bank. The frame embeddings from the visual encoder are then passed autoregressively through the long-term memory bank and Q-Former. The loss for the Q-Former is computed independently of the gradient values of the APM weights.

During inference, the Q-Former is frozen and the APM is trained, similar to other fast weight architectures [8]. For each video, APM processes the whole video before passing its weights to the Q-Former This allows the fast weight memory bank to learn features unique to each video.

4 Dataset

For video understanding, there are three primary tasks that we focus on with reference from the MA-LMM paper [4].

Long Term Video Classification Task. This task consists of watching a video and then classifying the action of the video into predefined categories. We focus on the Breakfast [9] dataset for the task of classifying home videos of people preparing breakfast food into one of 10 classes. The dataset has 1989 total videos with a mean video length of 1 minute 22 seconds.

Video Captioning Task. The video captioning task is the model generating a short caption after watching a video. This is typically a short video classification task. We focus on the MSVD [2] and MSR-VTT [13] datasets, both of which are 10 to 15 second videos scraped from youtube.

Video Question Answering Task. This task has the model process the video, then the model is asked a question about the video, and the model gives a one word response. We use the MSVD-QA [12] and MSVETT-QA [12] datasets.

5 Results

In our experiment, we test our architecture with two and four training steps per batch of 16 frames per video in 3. There is no significant difference between the number of training steps for accuracy. The best epoch for APM performed worse than the best epoch for the baseline MA-LMM results, as shown in Table 1. The lower APM performance is likely due to not all of the APM MLP weights being passed to the fast weight memory bank. The moderate APM performance could also indicate that the architecture could be improved to better interpret the APM weights.

	No APM	APM 2 Steps	APM 4 Steps
Best Epoch Accuracy	60.6	51.22	51.64

Table 1: Best Epoch Accuracy across APM configurations.

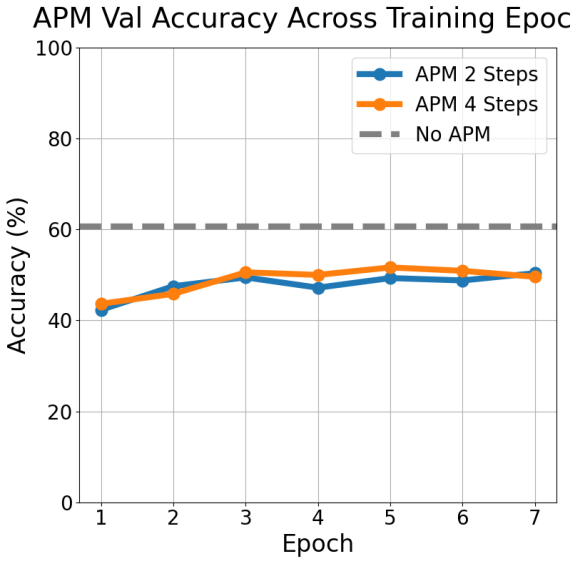


Figure 3: Our model's performance with two and four APM training

6 Future Work

Further experiments and architecture modifications are needed to determine how to best incorporate weight memory bank information into long video understanding models.

To determine how the fast weight memory bank is interacting with the transformer, an experiment where the fixed memory bank and learned memory bank Q-Formers are separated would be intriguing. Our APM is currently trained before passing the fast weight memory bank to the Q-Former, but we have not yet measured the model's performance when APM is trained in parallel to the Q-Former.

Another limitation of our current setup is the use of fixed-length video clips, inherited from the MA-LMM architecture [4]. The MA-LMM architecture samples a small number of frames from the input video for processing. For instance, for the question answering task, only 20 frames are sampled. We hypothesize that the fast weight memory bank will perform better when scaling to longer videos, like MovieNet [7]. Future experiments on longer video datasets would validate this hypothesis.

7 Conclusion

This report presents a novel approach to video understanding by introducing a learnable fast weight memory bank, inspired by biological plasticity and associative memory. While prior work has largely relied on fixed-length memory compression, our method explores a dynamic alternative that stores temporal information directly in adaptable weights. Initial results suggest that this direction is promising, though further experimentation is required to refine the architecture and fully integrate fast weight memory into temporal feature extraction pipelines.

References

[1] Shehreen Azad, Vibhav Vineet, and Yogesh Singh Rawat. Hierarq: Task-aware hierarchical q-former for enhanced video understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 8545–8556, 2025.

[2] David Chen and William B Dolan. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, pages 190–200, 2011.

[3] Gueter Josmy Faure, Jia-Fong Yeh, Min-Hung Chen, Hung-Ting Su, Shang-Hong Lai, and Winston H. Hsu. Hermes: temporal-coherent long-form understanding with episodes and semantics, 2025.

[4] Bo He, Hengduo Li, Young Kyun Jang, Menglin Jia, Xuefei Cao, Ashish Shah, Abhinav Shrivastava, and Ser-Nam Lim. Ma-lmm: Memory-augmented large multimodal model for long-term video understanding, 2024.

[5] Geoffrey E Hinton and David C Plaut. Using fast weights to deblur old memories. In *Proceedings*

of the ninth annual conference of the Cognitive Science Society, pages 177–186, 1987.

[6] John J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, 1982.

[7] Qingqiu Huang, Yu Xiong, Anyi Rao, Jiaze Wang, and Dahua Lin. Movienet: A holistic dataset for movie understanding, 2020.

[8] Geoffrey Hinton Jimmy Ba, Volodymyr Mnih, Joel Z. Leibo, and Catalin Ionescu. Using fast weights to attend to the recent past, 2016.

[9] H. Kuehne, A. B. Arslan, and T. Serre. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *Proceedings of Computer Vision and Pattern Recognition Conference (CVPR), year =*.

[10] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, 2023.

[11] Rajat Modi and Yogesh Rawat. Asynchronous perception machine for efficient test time training. *Advances in Neural Information Processing Systems*, 37:106020–106058, 2024.

[12] Dejing Xu, Zhou Zhao, Jun Xiao, Fei Wu, Hanwang Zhang, Xiangnan He, and Yueting Zhuang. Video question answering via gradually refined attention over appearance and motion. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1645–1653, 2017.

[13] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5288–5296, 2016.

[14] Robert S Zucker and Wade G Regehr. Short-term synaptic plasticity. *Annual review of physiology*, 64(1):355–405, 2002.