Supplementary Material

REEF: Relevance-Aware and Efficient LLM Adapter for Video Understanding

Sakib Reza[1*] Xiyun Song[2] Heather Yu[2] Zongfang Lin[2] Mohsen Moghaddam[3] Octavia Camps[1]
[1]Northeastern University [2]Futurewei Technologies Inc [3]Georgia Institute of Technology
{reza.s, o.camps}@northeastern.edu, {xsong, hyu, zlin1}@futurewei.com, mohsen.moghaddam@gatech.edu

1. Differential Top-K Operation

In the main paper, we explain how the perturbed maximum method [1, 2, 5] is used to make the Top-K operator differentiable, enabling the training of the token scorer networks. Building on this, we solve Equation 3 (in the main paper), and as outlined in [5], we carry out the following forward and backward operations to enable end-to-end training of the scorer networks with Top-K operations.

Forward Pass A smooth approximation of the Top-K operation from the main paper's Equation 3 can be implemented by introducing random perturbations and taking the expectation over these perturbations:

$$\mathbf{Y}_\sigma = \mathbb{E}_{\mathbf{Z}}\left[\arg\max_{\mathbf{Y}\in\mathcal{C}} \left\langle \mathbf{Y}, \mathbf{r}'\mathbf{1}^\top + \sigma\mathbf{Z}\right\rangle\right] \tag{1}$$

where $\mathbf{Z}$ denote a random noise vector drawn from a standard Gaussian distribution, with σ as the hyperparameter controlling the noise variance. The symbol $\langle\cdot,\cdot\rangle$ represents the inner product operation throughout the main paper and supplementary material. In practice, we execute the Top-K operation n times (where $n = 500$, as determined empirically to perform effectively across all experiments) and compute the average across these iterations.

Backward Pass As described in [2, 5], the Jacobian for the forward pass can be computed as:

$$J_s\mathbf{Y} = \mathbb{E}_{\mathbf{Z}}\left[\arg\max_{\mathbf{Y}\in\mathcal{C}} \left\langle \mathbf{Y}, \mathbf{r}'\mathbf{1}^\top + \sigma\mathbf{Z}\right\rangle \mathbf{Z}^\top/\sigma\right] \tag{2}$$

This expression simplifies in the case where $\mathbf{Z}$ is normally distributed, allowing for efficient backpropagation through the Top-K operation.

We train the backbone models along with the token selection networks using a cross-entropy loss (described in equation 2 in the main paper). During training, the learned Top-K operation (implemented in PyTorch [3]) is applied in the forward pass to enhance performance. By continuously applying random noise, the network becomes aware of the Top-K operation, allowing it to improve over time.

2. Query Memory Bank

Query Memory Bank (QMB) differs from the fixed visual memory bank, which stores static visual features. Instead, it accumulates input queries over time, denoted as $\mathbf{\Theta}_t = \text{Concat}[\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_t], \mathbf{\Theta}_t \in \mathbb{R}^{tL\times D}$, creating a dynamic memory that captures the model's evolving understanding of each frame up to the current timestep through the Q-Former. This Query Memory Bank also serves as the key and value: $\mathbf{Q} = \boldsymbol{\theta}_t\mathbf{W}_Q, \quad \mathbf{K} = \mathbf{\Theta}_t\mathbf{W}_K, \quad \mathbf{V} = \mathbf{\Theta}_t\mathbf{W}_V$ as previously described. At each timestep, the learned query $\boldsymbol{\theta}_t$ encapsulates critical information specific to the video up to that point. Unlike the static visual memory bank, these input queries $\boldsymbol{\theta}_t$ are progressively refined through cascaded Q-Former blocks during training, enabling the capture of distinct video concepts and patterns at increasingly abstract levels. As a result, each self-attention layer is associated with a unique Query Memory Bank, where the stored queries are continuously updated during training.

3. GFLOPs Calculation

To evaluate our method's computational efficiency, we used giga floating-point operations (GFLOPs), a common metric for assessing efficiency in machine learning models. We employed the *ptflop* python package [4] to calculate the GFLOPs for different components of our method. As noted in the main paper, the GFLOPs presented in the results tables are specific to the LLM adapter for a single pass. They exclude the LLM itself and are estimates, as certain components were manually calculated. Since we used the same LLM model with the same input token size as the baseline and focused exclusively on improving the adapter, we reported only the measurements related to the adapter for a

*Work done during an internship at Futurewei Technologies Inc.

Table 1. Hyperparameters used in experiments across different datasets for the untrimmed video classification task.

Datasets	**Breakfast**	**ActivityNet**
LLM	Vicuna-7B	
Initial Training Epoch	5	
Main Training Epoch	20	
Learning Rate	1e-4	
Batch Size	64	
AdamW β	(0.9, 0.999)	
Weight Decay	0.05	
Image Resolution	224	
Beam Size	5	
Frame Length	20	
Memory Bank Length	10	
Score Balance Weight α	0.7	0.9
Selected Spatial Tokens	100	144
Prompt	"What type of breakfast is shown in the video?"	"what is the person doing in the video?"

Table 2. Hyperparameters used in experiments across different datasets for the video question answering task.

Datasets	**MSVD**	**ActivityNet**
LLM	Vicuna-7B	
Initial Training Epoch	2	
Main Training Epoch	5	
Learning Rate	1e-4	
Batch Size	128	
AdamW β	(0.9, 0.999)	
Weight Decay	0.05	
Image Resolution	224	
Beam Size	5	
Frame Length	20	
Memory Bank Length	10	
Score Balance Weight α	0.7	0.9
Selected Spatial Tokens	100	121
Prompt	"Question: {} Short Answer:"	

Table 3. Hyperparameters used in experiments across different datasets for the video captioning task.

Datasets	**MSVD**	**YouCook2**
LLM	Vicuna-7B	
Initial Training Epoch	3	
Main Training Epoch	10	
Learning Rate	1e-5	1e-4
Batch Size	96	
AdamW β	(0.9, 0.999)	
Weight Decay	0.05	
Image Resolution	224	
Beam Size	5	
Frame Length	80	
Memory Bank Length	40	
Score Balance Weight α	0.7	
Selected Spatial Tokens	100	144
Prompt	"what does the video describe?"	

fair comparison and to enhance the reader's convenience.

It is important to note that the estimated GFLOPs measurement for Video-Llama [6] is not directly comparable to MA-LMM or our approach, as Video-Llama processes the entire video in one pass without requiring an online iterative process. However, we included it for reference. Additionally, **'N/A'** is indicated for non-LLM methods in the results tables since they do not involve an LLM adapter, and we only reported GFLOPs for the LLM adapter.

4. Experiment Configurations

The hyperparameters used in our experiments across different tasks and datasets are outlined in Tables 1, 2, and 3, detailing the configurations for untrimmed video classification, video question answering, and video captioning, respectively. These hyperparameters were determined empirically through systematic experimentation. For further details, the configuration files are available on GitHub alongside the codebase.

References

[1] Quentin Berthet, Mathieu Blondel, Olivier Teboul, Marco Cuturi, Jean-Philippe Vert, and Francis Bach. Learning with differentiable pertubed optimizers. *Advances in neural information processing systems*, 33:9508–9519, 2020. 1

[2] Jean-Baptiste Cordonnier, Aravindh Mahendran, Alexey Dosovitskiy, Dirk Weissenborn, Jakob Uszkoreit, and Thomas Unterthiner. Differentiable patch selection for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2351–2360, 2021. 1

[3] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 1

[4] Vladislav Sovrasov. ptflops: a flops counting tool for neural networks in pytorch framework, 2018-2024. 1

[5] Junke Wang, Xitong Yang, Hengduo Li, Li Liu, Zuxuan Wu, and Yu-Gang Jiang. Efficient video transformers with spatial-temporal token selection. In *European Conference on Computer Vision*, pages 69–86. Springer, 2022. 1

[6] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video un-

derstanding. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 543–553, 2023. 2