A. Model Details

Our model consists of three main components: the GOP encoder, the MLP projector, and a large language model (LLM). In the GOP encoder, we leverage a pre-trained SigLIP-so400m [13] as the RGB frame encoder. Simultaneously, we utilize a custom-designed transformer as the motion encoder to extract motion information.

Given that the motion vectors extracted from compressed video streams are represented as a discrete list L:

$$L = \text{List}\left[(x^i_{\text{src}}, y^i_{\text{src}}, x^i_{\text{dst}}, y^i_{\text{dst}})\right], \quad i = 1, \ldots, n. \tag{1}$$

we first transform it into a 2D spatial format P as follows:

$$P[x^i_{\text{src}}, y^i_{\text{src}}] = \left(x^i_{\text{dst}} - x^i_{\text{src}}, y^i_{\text{dst}} - y^i_{\text{src}}\right). \tag{2}$$

Since the motion vector represents displacements for macroblocks of size 4×4, and the original frame dimensions are $h \times w \times 3$, we derive a motion matrix of shape $(h/2, w/2, 2)$. Subsequently, the motion matrix is resized to a fixed resolution of 96×96, which is corresponding to the frame resolution 384×384.

We employ a two-layer transformer as the motion encoder, with a hidden size of 256 and 2 channels. The input motion matrix is processed using patches of size 7×7. After encoding, the motion feature has the same dimensionality as the frame feature, ensuring consistency across modalities.

For the extracted motion features, we apply temporal pooling along the time dimension to summarize the temporal dynamics. Additionally, to reduce the number of input tokens, we perform adaptive pooling on both the frame features and the motion features. This operation leverages the `torch.nn.AdaptiveAvgPool2d` module to efficiently compress spatial dimensions while preserving important information.

Subsequently, we employ a fusion layer to integrate the frame and motion information. This fusion process is implemented using a cross-attention layer followed by a feed-forward layer to facilitate modality interaction. Additionally, we incorporate a residual module to retain the input information, ensuring that critical details from both modalities are preserved during the fusion.

B. Training Hyperparameters

During the first-stage training, we freeze the frame encoder and the large language model (LLM) while training the motion encoder, the GOP fusion layer, and the modality projector. A global batch size of 128 is used, and the model is trained for 1000 steps. The motion encoder and projector are optimized with a learning rate of 1×10^{-4}, while the remaining components are trained with a learning rate of 2×10^{-5}.

In the second-stage training, we unfreeze all model parameters for joint optimization. The learning rates for different components are set as follows: the frame encoder uses a learning rate of 2×10^{-6}, the motion encoder uses 1×10^{-5}, the projector uses 1×10^{-4}, and the remaining components use 2×10^{-5}. The training is conducted for 2400 steps with a global batch size of 128.

We employ DeepSpeed ZeRO-2 for distributed training to efficiently handle large-scale models and data. During training, different samples are packed into a single sequence with a maximum length of 4096 for joint optimization, significantly improving training efficiency. The training was conducted on 16 NVIDIA A100 GPUs, with a total training time of approximately 16 hours.

Additionally, the extra motion warmup was conducted on 8 NVIDIA A100 GPUs. During this phase, we utilized a batch size of 1024 for supervised training with a learning rate of 1×10^{-3}. The training was performed on the motion vectors of SSV2 [2] training videos for a total of 30 epochs.

C. MotionBench Details

We used the label set from SSV2 [2] as the initial pool of options. Subsequently, we employed GPT-4o as a teacher model to filter these 174 options, extracting 114 classes that could be mapped to our predefined four categories: **Linear**, **Curved**, **Rotation**, and **Contact**.

To facilitate evaluation, we designed MotionBench as a multi-choice QA task. To increase the task complexity, we identified three hard negative labels for each category, which were included as confusing options in the QA design. GPT-4o assisted in selecting these hard negatives. For example, the following represents a set of confusing options:

Confusing Label Sets

"Pouring something out of something"

"Pouring something into something"

"Pouring something onto something"

"Pouring something into something until it overflows"

In addition, considering the limitations of video types in SSV2, we introduced [3, 9] as a supplementary data source. Since these videos lack initial labels, we utilized GPT-4o to generate dense captions for the videos. These dense captions were then matched to candidate categories, with the matching process also conducted by GPT-4o.

After the labeling process was completed, we performed a manual screening of the test videos. During this step, we filtered out incorrectly labeled examples and those that were overly simplistic, such as cases where the answer could be inferred directly from static images.

Finally, MotionBench comprises 4 distinct classes: Linear, Curved, Rotation, and Contact, containing 800, 500, 300, and 700 samples, respectively. We present the accuracy for each individual class as well as the average accuracy across all four classes.

D. External Ablations

D.1. Use of Temporal Prompt

When feeding GOPs into the LLM, we added a textual temporal prompt for each GOP, which included its time coordinates. We found that this simple approach led to a significant performance improvement on long benchmarks, such as VideoMME [1]. However, it had a smaller impact on shorter VideoQA benchmarks, such as MSVD-QA [11] and MSRVTT-QA [11].

Table 1. Impact of temporal prompt. A simple textual temporal prompt proves beneficial for long video tasks, such as VideoMME, but has a smaller impact on shorter VideoQA tasks, such as MSVD-QA and MSRVTT-QA.

Model	**MSVD-QA** Acc. / Score	**MSRVTT-QA** Acc. / Score	**MotionBench** Avg.	**VideoMME** w/o sub / w sub
EMA	75.8 / 4.1	58.5 / 3.5	50.0	53.4 / 58.4
EMA w/o Temporal Prompt	75.6 / 4.1	58.5 / 3.5	49.8	51.2 / 56.7

D.2. GOP Token Number

In **EMA**, we employ a 3×3 pooling kernel to reduce the length of GOP tokens by a factor of 9. In this section, we evaluate the impact of this compression strategy across several VideoQA benchmarks. We experiment with different pooling kernel sizes while keeping the rest of the training setup consistent. Our results show that the 3×3 pooling kernel achieves performance comparable to both the 2×2 pooling and no pooling configurations, while benefiting from a significant reduction in token length (1/9 of the original), thereby accelerating inference.

Table 2. Influence of pooling strategy.

Pooling Strategy	**GOP Token Number**	**MSVD-QA** Acc. / Score	**MSRVTT-QA** Acc. / Score	**MotionBench** Avg.	**VideoMME** w/o sub / w sub
w/o pooling	729	76.0 / 4.1	58.9 / 3.5	50.2	53.6 / 58.9
2×2 pooling	196	75.8 / 4.1	58.4 / 3.5	49.7	53.3 / 58.4
3×3 pooling	81	75.8 / 4.1	58.5 / 3.5	50.0	53.4 / 58.4
4×4 pooling	49	73.6 / 3.9	56.8 / 3.3	49.0	52.0 / 56.2

E. Evaluation Results on More Long Video Benchmark

We evaluate our model's performance on additional long-video benchmarks MLVU [18], LongVideoBench [10], and VN-Bench [17]. We compare **EMA** with existing video understanding models. Our model demonstrated outstanding performance across these benchmarks as well.

Table 3. Evaluation result on long video benchmarks, MLVU [18], LongVideoBench [10] and VNBench [17]

Model	MLVU Dev	LongVideoBench Val	VNBench Overall
VideoChat [4]	29.2	-	-
VideoChatGPT [8]	31.3	-	4.1
Video-LLaVA [6]	47.3	39.1	12.4
Video-LLaMA2 [14]	35.5	-	4.5
LLaMA-VID [5]	33.2	-	7.1
LLaVA-NeXT-Video [16]	-	43.5	20.1
ST-LLM [7]	-	-	22.7
LongVA [15]	56.3	-	-
Qwen2-VL-7B [12]	55.6	-	33.9
EMA	**57.2**	**47.0**	**32.6**

F. Acknowledgement

This research is supported by Artificial Intelligence-National Science and Technology Major Project (2023ZD0121200) and the National Natural Science Foundation of China (6243000159, 62102416), and the Key Research and Development Program of Jiangsu Province under Grant BE2023016-3.

References

[1] Chaoyou Fu, Yuhan Dai, Yondong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. *arXiv preprint arXiv:2405.21075*, 2024. 2

[2] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The" something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE international conference on computer vision*, pages 5842–5850, 2017. 1

[3] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 2

[4] KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. Videochat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355*, 2023. 3

[5] Yanwei Li, Chengyao Wang, and Jiaya Jia. Llama-vid: An image is worth 2 tokens in large language models. *arXiv preprint arXiv:2311.17043*, 2023. 3

[6] Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. *arXiv preprint arXiv:2311.10122*, 2023. 3

[7] Ruyang Liu, Chen Li, Haoran Tang, Yixiao Ge, Ying Shan, and Ge Li. St-llm: Large language models are effective temporal learners. In *European Conference on Computer Vision*, pages 1–18. Springer, 2025. 3

[8] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. *arXiv preprint arXiv:2306.05424*, 2023. 3

[9] K Soomro. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 2

[10] Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. Longvideobench: A benchmark for long-context interleaved video-language understanding. *arXiv preprint arXiv:2407.15754*, 2024. 3

[11] 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 *ACM Multimedia*, 2017. 2

[12] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024. 3

[13] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11975–11986, 2023. 1

[14] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*, 2023. 3

[15] Peiyuan Zhang, Kaichen Zhang, Bo Li, Guangtao Zeng, Jingkang Yang, Yuanhan Zhang, Ziyue Wang, Haoran Tan, Chunyuan Li, and Ziwei Liu. Long context transfer from language to vision. *arXiv preprint arXiv:2406.16852*, 2024. 3

[16] Yuanhan Zhang, Bo Li, haotian Liu, Yong jae Lee, Liangke Gui, Di Fu, Jiashi Feng, Ziwei Liu, and Chunyuan Li. Llava-next: A strong zero-shot video understanding model, 2024. 3

[17] Zijia Zhao, Haoyu Lu, Yuqi Huo, Yifan Du, Tongtian Yue, Longteng Guo, Bingning Wang, Weipeng Chen, and Jing Liu. Needle in a video haystack: A scalable synthetic framework for benchmarking video mllms. *arXiv preprint arXiv:2406.09367*, 2024. 3

[18] Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Shitao Xiao, Xi Yang, Yongping Xiong, Bo Zhang, Tiejun Huang, and Zheng Liu. Mlvu: A comprehensive benchmark for multi-task long video understanding. *arXiv preprint arXiv:2406.04264*, 2024. 3