UTDL Submission to the 2023 Epic-Kitchens Challenge

Yue Zhao Philipp Krähenbühl
UT Austin
yzhao@cs.utexas.edu

Abstract

*This report describes the approach behind our solution to the 2023 Epic-Kitchens Action Recognition and Multi-Instance Retrieval Challenge. Our approach builds upon our recent work, Language-augmented Video-Language Pre-training (LaViLa). We propose several improvements such that we can successfully pre-train and fine-tune a medium- to large-sized video-text dual encoder model with **8** consumer grade GPUs. Our final submission consists of an ensemble of models varying backbone sizes. On the Action Recognition Challenge, our approach achieved 54.3% Action-level Top-1 accuracy on the test set, 1.5% higher than last year's winning entry while using a significantly smaller number of ensemble models and pre-training data. On the Multi-Instance Retrieval Challenge, our approach achieved 70.8% average nDCG on the test set, 9.3% higher than last year's winning entry.*

1. Introduction

Following the success of CLIP [12] in the the field of image understanding, video-language pre-training has proven an effective approach for video understanding. However, it requires a significantly increasing pre-training cost. Particularly, training a CLIP-style video-language model typically requires as many as 32 to 64 GPUs or TPUs [10, 11] to construct a batch of $\sim 1K$ video clips.

In this submission we propose a solution of pre-training a state-of-the-art video-language model on a single machine with 8 consumer grade GPUs by optimizing the training pipeline from three aspects: model, video loading, and video pre-processing. The pre-trained model can seamlessly adapt to down-stream tasks through end-to-end fine-tuning and achieve excellent performance on the chosen Action Recognition and Multi-Instance Retrieval Challenge.

2. Method

2.1. Optimizing Pre-training Pipeline

We follow the pre-training pipeline LaViLa [17] and introduce a series of techniques to optimize the GPU memories, CPU utilization, and IO bottleneck respectively.

A memory-efficient video ViT. We choose a plain video ViT [1, 6] architecture rather than a TimeSformer [2] since ViT is more memory and computationally efficient if optimized properly.

We observe that the attention operator accounts for over 60% of the overall memory consumption in a plain video ViT architecture. We remove this memory footprint using FlashAttention [5]. We can further trade computation for memory efficiency through gradient checkpointing [3].

Increasing CPU Utilization in Pre-processing. A typical video training pipeline consists of both video decoding and cropping, both of which are CPU intensive. With a larger batch size, CPU-intensive video pre-processing becomes a bottleneck. To address this, we propose to merge RandomResizedCrop, which is a standard cropping operation in contrastive visual-language pre-training, into the video decoding stage as a cropping filter.

Eliminating IO bottleneck. Both Ego4D and Epic-Kitchens videos are long-term videos with an average length of ~ 20 minutes. An increased throughput will pose challenges to the disk IO because of video loading. One solution is to split each input video into multiple fixed-length chunks [10, 17]. While the length of these chunks is often chosen heuristically, we propose a way to compute the optimal chunk length in the following.

Let B denote the batch size, ρ denote the average bitrate of a video, S_r denote the maximum read speed, and Δ denote the elapsed time of one training iteration. To hide the IO bottleneck from the training, we require the video model to consumer fewer bits $B \times \rho \times T$ than the disk can provide $S_r \times \Delta$:

$$B \times \rho \times T \leq S_r \times \Delta. \tag{1}$$

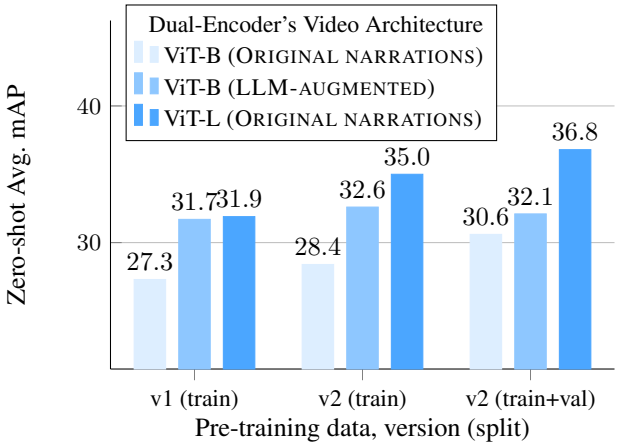
Note that we only control the length T of each chunk while the rest of the variables are determined by the hardware en-

Method	Hardware	Batch size	Mem. (GB)	GPU·hour	0-shot Avg. mAP
(ORIGINAL NARRATIONS)					
EgoVLP	32× A100	16	22	1,536	22.1
Ours	8× A5000	256	19	170(-89%)	27.4(+5.3)
(LLM-AUGMENTED)					
LaViLa	32× V100	32	25	1,824	29.5
Ours	8× A5000	256	19	408(-78%)	31.7(+2.2)

Table 1. **Comparison of Training cost.**

Backbone	Split	Ver.	LM-aug.	Corpus size	0-shot Avg. mAP
ViT-B	train	v1		4.0M	27.3
	train	v2		5.5M	28.4
ViT-B	train	v1	✓	35M	31.7
	train	v2	✓	36.5M	32.6
ViT-L	train	v1		4.0M	31.9
	train	v2		5.5M	35.0
	train+val	v2		6.6M	36.8

Table 2. **Scaling effect of video-language pre-training.**



(a) EK-100 MIR 0-Shot mAP.

Figure 1. **Effect of pre-training batch size.** The numbers are measured using $T = 4$ frames as input. Large-batch training, which was not possible without multi-node training, benefits the video-language contrastive models consistently especially in the presence of larger scale narrations.

vironment. In our experimental setup, typical values for a ViT-base model are $N = 1024$, $\rho = 1$ Mb/sec, $\Delta = 4$ sec and $S_r = 500$ MB/sec, which leads to $T \leq 16$ sec. Therefore, we use 15-second chunks in practice for both Ego4D and Epic-Kitchens videos.

2.2. Fine-tuning on the Down-stream Tasks

Fine-tuning Action Recognition Models. When fine-tuning on the action recognition task, we take the video encoder of the pre-trained dual-encoder model, drop the last projection layer, and attach a classification head. We then end-to-end train the model on the training split and evaluate it on the validation/test split. Note that due to resource limit, we did *not* train another model on the joint train+val split when submitting the testing result to the leader-board.

Fine-tuning Multi-Instance Retrieval Models. When fine-tuning on the multi-instance retrieval task, we take the dual-encoder model *as is*, and fine-tune it with the multi-instance max-margin loss [13], which proves more effective for this task than the InfoNCE loss.

3. Experiments

3.1. Video-Language Pre-training on Ego4D

Experimental Setup. The video-language model follows CLIP [12]. The vision encoder is a Vision Transformer Base (ViT-B) model [6], whose weights are initialized from CLIP except that we randomly initialize the temporal position embedding PE_t and then add it to the original patch-wise (spatial) position embedding. We represent each video clip by $T = 4$ frames when pre-training on Ego4D.

Pre-training efficiency. In Table 1, we showcase the pre-training efficiency after optimization with the ViT-B backbone. With the original 4M ground-truth narrations, our model can be trained in 5 full epochs using 8× RTX A5000 GPUs in 21 hours, accounting for only $\frac{1}{9}$ of the GPU·hours required by EgoVLP [10]. Our model also benefits from larger-scale pseudo-narrated video-text pairs generated by LaViLa [17].

Scaling effect. We study the effect of data scaling in Table 2. After a recent update[1], Ego4D v2 narrations increase by 30%. Besides, we include videos from the validation and testing splits of the Ego4D challenge benchmarks, resulting in 20% more video-narration pairs (denoted as "train+val"). We find that the zero-shot performance consistently increases with respect to the increased corpus size. We also observe an similar trend of improvement using LaViLa (denoted as "LM-aug.") with ViT-B. This indicates the importance of increasing data scale for video-language pre-training. We leave evaluating LaViLa w/. ViT-L for future work due to time limit.

3.2. Fine-tuning EK100 Action Recognition

Experimental Setup. The fine-tuning protocol mostly follows LaViLa [17]. When fine-tuning on EK-100 CLS, we increase input frame number T from 4 to 16 and linearly interpolate PE_t along the temporal dimension. Unlike pre-

[1] https://discuss.ego4d-data.org/t/ego4d-v2-0-release-updates/197

Method	Backbone	Pre-train	Top-1 Accuracy		
			Verb	Noun	Action
(VALIDATION SPLIT)					
MoViNet [9]	MoViNet	N/A	72.2	57.3	47.7
MTCN [8]	MFormer-HR	IN-21k+K400 +VGG-Sound	70.7	62.1	49.6
Omnivore [7]	Swin-B	IN-21k+IN-1k+K400+SUN	69.5	61.7	49.9
MeMViT [14]	MViT	K600	71.4	60.3	48.4
MTV [16]	MTV	WTS-60M	69.9	63.9	50.5
M&M-B [15]	MTV	WTS-60M	72.0	66.3	53.6
LaViLa [17]	TSF-L	WIT+Ego4D	72.0	62.9	51.0
Ours	ViT-B	WIT+Ego4D$^+$	69.1	60.6	49.1
	ViT-L		72.6	65.4	54.4
	ViT-L@336px		**73.4**	**66.7**	**55.4**
(TEST SPLIT)					
M&M-B [15]	MTV	WTS-60M	68.0	63.7	49.6
Ours	ViT-L	WIT+Ego4D	70.7	64.3	52.2
Ours	Ensemble (B+L+Audio)	WIT+Ego4D$^+$	**71.7**	**65.8**	**54.3**

Table 3. **Comparison to the state-of-the-art on EK-100 CLS.** Ego4D$^+$ denotes the v2 annotations under train+val splits.

vious works [15, 16], we only apply RandomResizedCrop for data augmentation since we observe slower convergence when using RandomAugmentation. We use a mixup of 0.8, label smoothing of 0.1, a stochastic depth ratio of 0.1, and a dropout layer before the classification head with probability of 0.5. We train a single action-level classification head and compute the verb- and noun-level scores at test time only through marginalization.

Comparison to the State-of-the-art. Table 3 compares our fine-tuned recognition models to the previous state-of-the-art as well as last year's winning entry of the challenge (M&M-B). We can see that our single model with ViT-L backbone outperforms previous works on Top-1 action- and verb-level accuracies. When the input resolution increases from the default 224 to 336, the performance can further improves. Also note that compared to M&M-B, our model relies on a smaller resolution (336 *vs*. 432), fewer input frames (16 *vs*. 64 frames), and publicly available pre-trained models and a smaller pre-training dataset.

It is also noteworthy that the val-test gap between ours is remarkably ($54.4 - 52.2 = 2.2$) smaller than M&M-B ($53.6 - 49.6 = 4$). This indicates that our model is less overfitted probably because Ego4D videos for pre-training are visually more similar to Epic-Kitchens videos compared to YouTube videos from WTS-60M.

Model Ensemble. In Table 4 we enumerate the models used in the ensemble: (1) two ViT-L models trained on Ego4D v2 training videos with the only difference that we train one using random erasing with probability of 0.8 (Row 2) while the other not (Row 1); (2) one ViT-L model trained on Ego4D v2 training+validation videos; and (3) two ViT-L models that take higher-resolution (336 pixels) videos as input. We additionally train a ViT-B model using audio as input following the pre-processing pipline in [15]. We observe that it always improves the ensemble performance by $0.2 \sim 0.3\%$ on the validation split through late fusion.

Backbone	Resolution	Pre-train	Top-1 Accuracy		
			Verb	Noun	Action
ViT-L	224	v2 train	73.3	65.0	54.0
ViT-L	224	v2 train	72.4	64.8	53.5
ViT-L	224	v2 train+val	72.6	65.4	54.4
ViT-L	336	v2 train	73.2	66.2	54.7
ViT-L	336	v2 train+val	73.4	66.7	55.4
ViT-B (Audio)	224	v1 train	50.4	25.4	20.6

Table 4. **Model variants used in the final ensemble.**

3.3. Fine-tuning EK100 Multi-Instance Retrieval

Experimental Setup. When fine-tuning on EK-100 MIR, we also increase input frame number T from 4 to 16 and linearly interpolate PE_t. We use the multi-instance max-margin loss [13] with a margin value of 0.2.

Comparison to the State-of-the-art. Table 5 compares our fine-tuned retrieval models to the previous state-of-the-art as well as last year's winning entry of the challenge (EgoVLP++). We can see that our model with a ViT-Base backbone can achieve a higher average mAP and same nDCG compared to the previous state-of-the-art (LaViLa) with a larger backbone (TimeSformer-Large). This is probably ascribed to the enlarged batch size during fine-tuning. When we switch to a larger backbone (ViT-L), the performance further increases.

Following EgoVLP [10], we also observe that applying dual-softmax [4] on the dot-product between the *unnormalized* video and textual embedding is consistently better than using the cosine similarity matrix between the normalized video and textual embedding. We did not run the adaptive margin loss due to time and computation limit.

Model Ensemble. Finally, we find that model ensembling benefits retrieval performance as well. The intuition is that each row after the second softmax operation in [4] can be interpreted as a probability distribution. Formally, given the video-to-text similarity matrix of the i-th model is $\mathbf{S}_i$, we compute their weighted average, namely

$$\mathbf{S}_{\text{ensemble}} = \frac{\sum_i \alpha_i \mathbf{S}_i}{\sum_i \alpha_i}. \tag{2}$$

We take two models, namely ViT-B and ViT-L, and empirically find that $\alpha_1 = \alpha_2 = 1$ works the best.

Method	Backbone	+DS [4]	mAP			nDCG		
			V→T	T→V	Avg.	V→T	T→V	Avg.
EgoVLP [10]	TSF-B		49.9	40.5	45.0	60.9	57.9	59.4
EgoVLP++ [10]	TSF-B	✓	53.8	41.0	47.4	63.3	59.6	61.4
LaViLa [17]	TSF-L		54.7	47.1	50.9	68.1	64.9	66.5
Ours	ViT-B		56.8	47.4	52.1	68.3	64.7	66.5
	ViT-B	✓	58.2	48.3	53.3	68.8	65.5	67.2
	ViT-L		59.0	49.9	54.4	70.1	67.0	68.5
	ViT-L	✓	61.9	51.8	56.9	71.2	68.2	69.7
Ours	Ensemble (B+L)	✓	63.1	53.7	58.4	72.2	69.2	70.7

Table 5. **Comparison to the state-of-the-art on EK-100 MIR.** EgoVLP++ denotes an improved model which uses adaptive margin in the max-margin loss and dual-softmax (+DS) [4]. It is also the winning entry of the 2022 challenge.

4. Conclusions

In this report, we present the approach behind our submission to the 2023 Epic-Kitchens Action Recognition and Multi-Instance Retrieval Challenge. With the proposed techniques, we are able to efficiently pre-train and fine-tune all our models on eight consumer grade GPUs and also set a new record on the Epic-Kitchens Action Recognition and Multi-Instance Retrieval benchmarks. We believe that our techniques are generic and can be beneficial to scaling video models if more resources are available.

References

[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *ICCV*, 2021. 1

[2] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, 2021. 1

[3] Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. Training deep nets with sublinear memory cost. *arXiv preprint arXiv:1604.06174*, 2016. 1

[4] Xing Cheng, Hezheng Lin, Xiangyu Wu, Fan Yang, and Dong Shen. Improving video-text retrieval by multi-stream corpus alignment and dual softmax loss. *arXiv preprint arXiv:2109.04290*, 2021. 3, 4

[5] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *NeurIPS*, 2022. 1

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 1, 2

[7] Rohit Girdhar, Mannat Singh, Nikhila Ravi, Laurens van der Maaten, Armand Joulin, and Ishan Misra. Omnivore: A single model for many visual modalities. In *CVPR*, 2022. 3

[8] Evangelos Kazakos, Jaesung Huh, Arsha Nagrani, Andrew Zisserman, and Dima Damen. With a little help from my temporal context: Multimodal egocentric action recognition. In *BMVC*, 2021. 3

[9] Dan Kondratyuk, Liangzhe Yuan, Yandong Li, Li Zhang, Mingxing Tan, Matthew Brown, and Boqing Gong. Movinets: Mobile video networks for efficient video recognition. In *CVPR*, 2021. 3

[10] Kevin Qinghong Lin, Jinpeng Wang, Mattia Soldan, Michael Wray, Rui Yan, Eric Z XU, Difei Gao, Rong-Cheng Tu, Wenzhe Zhao, Weijie Kong, et al. Egocentric video-language pretraining. *NeurIPS*, 2022. 1, 2, 3, 4

[11] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *CVPR*, 2020. 1

[12] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 1, 2

[13] Michael Wray, Diane Larlus, Gabriela Csurka, and Dima Damen. Fine-grained action retrieval through multiple parts-of-speech embeddings. In *ICCV*, 2019. 2, 3

[14] Chao-Yuan Wu, Yanghao Li, Karttikeya Mangalam, Haoqi Fan, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Memvit: Memory-augmented multiscale vision transformer for efficient long-term video recognition. In *CVPR*, 2022. 3

[15] Xuehan Xiong, Anurag Arnab, Arsha Nagrani, and Cordelia Schmid. M&m mix: A multimodal multiview transformer ensemble. *arXiv preprint arXiv:2206.09852*, 2022. 3

[16] Shen Yan, Xuehan Xiong, Anurag Arnab, Zhichao Lu, Mi Zhang, Chen Sun, and Cordelia Schmid. Multiview transformers for video recognition. In *CVPR*, 2022. 3

[17] Yue Zhao, Ishan Misra, Philipp Krähenbühl, and Rohit Girdhar. Learning video representations from large language models. In *CVPR*, 2023. 1, 2, 3, 4