Supplementary Material for PERCEIVER-VL: Efficient Vision-and-Language Modeling with Iterative Latent Attention

Zineng Tang* Jaemin Cho* Jie Lei Mohit Bansal
UNC Chapel Hill
{terran, jmincho, jielei, mbansal}@cs.unc.edu

In this appendix, we provide additional efficiency analysis (Sec. B) ablation studies (Sec. C), and full experiment results (Sec. D)

A. Structured Decoding with Cross-Attention and Query Array

We continue Sec.4 to discuss downstream tasks decoder queries.

A.0.1 Visual Question Answering

We tackle visual question answering tasks as a classification task (e.g., VQAv2), by choosing the right answer from the a predefined answer vocabulary, following [24]. Similarly to the VTM task, we create a decoder query with a `[CLS]` embedding ($Q = 1$), then apply a classification head with cross-entropy loss.

A.0.2 Cross-Modal Retrieval

We tackle cross-modal retrieval tasks by first estimating the multi-modal similarity scores s^{VL} of image-text or video-text pairs, then retrieving contents by ranking the similarity scores. We study different types of architecture for this task and explain the details in Sec. 3.5. For multi-stream architecture, similar to the VTM task, we create a decoder query with a `[CLS]` embedding ($Q = 1$), then apply a classification head with cross-entropy loss.

B. Efficiency Analysis

B.1. Scaling Latent Array

PERCEIVER-VL has a complexity of $O(MN)$, while the input size M is fixed for specific tasks and datasets. To complement the latent array scaling analysis on VQAv2 in the main paper Fig. 5, in Fig. 1, we additionally show the effect of varying the size of the latent array N during finetuning in terms of computation and downstream VQAv2 retrieval task performance. Note that we use N=128 during pretraining. We use mixed-stream architecture by default. We can see that the computational cost (GFLOPs) linearly scales with N, while the VQAv2 R@1 remains reasonably well (e.g., CC+Webvid PT: 66.6 $\rightarrow$ 68.0 $\rightarrow$ 69.2 $\rightarrow$ 69.8 with latent array length 32 $\rightarrow$ 64 $\rightarrow$ 128 $\rightarrow$ 256), across three different pretraining setups (Sec. C.2).

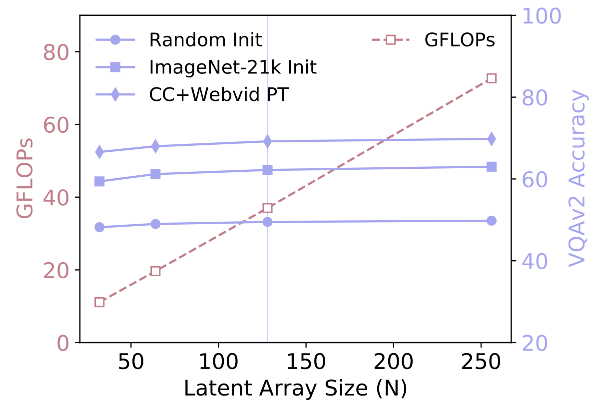

Figure 1. The efficiency-accuracy tradeoff of using different latent array size N during finetuning on VQAv2. During pretraining, we use the latent array size $N = 128$ (blue vertical line).

B.2. LayerDrop to Encoder Cross-Attentions

In Table 1 we analyze the effect of applying LayerDrop (LD) [7] to encoder cross-attention layers, as discussed in main paper Sec. 3.3 on an additional task, VQAv2. First, we observe that LD acts as a regularizer, as we see LD improves the VQAv2 accuracy in the first block, while increasing p^{LD} too high 0.5 $\rightarrow$ 0.7 hurts the performance (69.2 $\rightarrow$ 68.9). The last row in the bottom block achieves the best accuracy (69.5), with LD during both pretraining and finetuning. Second, removing cross-attention layers without LD during finetuning hurts performance (see 69.2 $\rightarrow$ 66.1 in the middle block). Lastly, with LD during finetuning, one can reduce the inference time latency around 16.7% (18.0 ms $\rightarrow$ 15.0 ms), with minimal accuracy drop (see 69.5 $\rightarrow$

*equal contribution

# Cross-attentions in encoder			VQAv2 Acc.	Time
Pretraining	Finetuning	Inference		(ms)
3	3	3	68.7	18.0
$1 \sim 3$ (0.5)	3	3	69.2	18.0
$1 \sim 3$ (0.7)	3	3	68.9	18.0
$1 \sim 3$ (0.5)	1	1	68.2	15.0
$1 \sim 3$ (0.5)	3	3	69.2	18.0
$1 \sim 3$ (0.5)	3	1	66.1	15.0
$1 \sim 3$ (0.5)	$1 \sim 3$ (0.5)	1	68.4	15.0
$1 \sim 3$ (0.5)	$1 \sim 3$ (0.5)	3	69.5	18.0

Table 1. Accuracy and inference time on VQAv2 with varied number of cross-attentions in PERCEIVER-VL encoder. We include the layer dropout probability p^{LD} in brackets if used. Note that PERCEIVER-VL has 3 cross-attention layers in encoder, and we do not apply dropout to the first cross-attention in encoder ($p^{LD} = 0$) to ensure that the latent array always receives signal from the input.

Aggregation Scheme	Weight initialization		GFLOPs ↓
	Random init	ImageNet-21k (ViT-B/32)	
Joint (default)	48.6	62.5	**30.5**
Separate	49.5	62.3	31.3
Separate+	**50.5**	**62.9**	33.2

Table 2. Comparison of different modality aggregation schemes (main paper Sec. 3.2) on VQAv2.

68.4 in the bottom block). This indicates that, with a LD-finetuned model, we can control its latency on demand at the inference time by varying the number of cross-attention layers, without storing checkpoints of multiple models.

C. Ablation Studies

We provide ablation studies regarding PERCEIVER-VL's architectural components and training strategy, including modality aggregation, pretraining dataset, positional encoding for latent arrays, and two-stage training for CLIP weight initialization.

C.1. Modality Aggregation

In Table 2, we compare different modality aggregation schemes for fusing visual and text inputs as we discussed in main paper Sec. 3.2. This study is performed on VQAv2 with two different weight initializations. In our experiments, we do not observe a significant difference among the three methods (*Joint*, *Separate*, *Separate+*) in terms of accuracy and GFLOPs. Thus, we use *Joint* as our default modality aggregation scheme for simplicity.

C.2. Pretraining Datasets

Table 3 shows the ablation of pretraining datasets in terms of two downstream tasks, VQAv2 and MSRVTT.

Pretraining Datasets	Modality			VQAv2	MSRVTT
	Image	Video	Text	Acc	R@1
Random Init (Standard Gaussian)				48.6	6.2
ImageNet-21k (ViT-B/32)	✓			62.3	12.1
ImageNet-21k (ViT-B/32) + CC	✓		✓	68.2	24.6
ImageNet-21k (ViT-B/32) + Webvid		✓	✓	67.5	25.1
ImageNet-21k (ViT-B/32) + CC + Webvid	✓	✓	✓	**69.2**	**26.8**

Table 3. Comparison of different pretraining datasets on VQAv2 and MSRVTT. ImageNet-21k (ViT-B/32) refers to weight initialization from the ViT-B/32 checkpoint pretrained on ImageNet-21k (main paper Sec. 4.3).

Positional Encoding	Weight init	
	Random Init	ImageNet-21k (ViT-B/32)
Learned (default)	49.5	**62.3**
Fourier	**49.7**	62.2

Table 4. Comparison of different position encodings for latent array on VQAv2.

Initializing PERCEIVER-VL parameters with ViT-B/32 ImageNet-21k pretrained weights (main paper Sec. 4.3) greatly improves the performance over random initialization. Further pretraining on image-text (CC) or video-text (Webvid) datasets further improves the performance. One interesting observation is that, pretraining on the data of the same format as the downstream task has slightly more advantages over data of different format – compared to video-text data, pretraining on image-text data gives more performance gain on image-text task (VQAv2), and vice versa. The best performance is achieved by PERCEIVER-VL pretrained on both datasets, showing that our framework benefits from input data from both formats.

C.3. Learned vs. Fourier Positional Encodings for Latent Array

In Table 4, we compare the learned [11, 32] and Fourier feature [30, 26, 16] positional encodings on VQAv2, as discussed in main paper Sec. 3.2. We do not see meaningful difference between the two positional encodings on two different weight initialization settings. Thus, we simply use the learned positional encoding as default positional encoding for the latent array.

Weight init	MSRVTT R@1
One-stage	36.3
Two-stage	45.9

Table 5. Comparison of one-stage vs. two-stage training for CLIP weight initializaiton on MSRVTT.

Model	Pretraining Datasets	Visual Backbone	Text-to-Video Retrieval (R@1/R@5/R@10) ↑				QA Accuracy ↑		GFLOPs ↓	Time (ms) ↓
			MSRVTT	DiDeMo	LSMDC	ActivityNet	TGIF-QA (A/T/F)	MSRVTT-QA		
Models using other input modalities (*e.g.*, audio)										
HERO [21]	TV/HT100M	ResNet152+Slowfast [12, 8]	20.5 / 47.6 / 60.9	-	-	-	-	-	935.2	2200.0
MMT [9]	HT100M	S3D+VGG+DenseNet161 [35, 13, 14]	26.6 / 57.1 / 67.1	-	12.9 / 29.9 / 40.1	-	-	-	-	-
AVLNET [29]	HT100M	ResNet152+ResNeXt [12, 34]	27.1 / 55.6 / 66.6	-	17.0 / 38.0 / 48.6	-	-	-	153.4	2000.0
Models with CLIP initialization										
Hunyuan [27]	-	CLIP (ViT-B/16)	**55.0 / 80.4 / 86.8**	**52.1 / 78.2 / 85.7**	**29.7** / 46.4 / 55.4	**57.3 / 84.8** / 93.1	-	-	2022.8	-
CLIP2TV [10]	-	CLIP (ViT-B/16)	49.3 / 74.7 / 83.6	45.5 / 69.7 / 80.6	-	44.1 / 75.2 / **98.4**	-	-	2212.3	-
DRL [33]	-	CLIP (ViT-B/32)	47.4 / 74.6 / 83.8	49.0 / 76.5 / 84.5	26.5 / **47.6 / 56.8**	46.2 / 77.3 / 88.2	-	-	511.0	320.0
CAMoE(+DSL) [5]	-	CLIP (ViT-B/32)	47.3 / 74.2 / 84.5	-	25.9 / 46.1 / 53.7	-	-	-	399.7	-
MDMMT-2 [19]	-	CLIP (ViT-B/32)	48.5 / 75.4 / 83.9	-	26.9 / 46.7 / 55.9	-	-	-	-	-
Ours$^{N=32}$ + CLIP	CC+Webvid	CLIP (ViT-B/16)	45.9 / 71.0 / 82.1	-	-	-	-	-	**80.0**	**80.0**
HT100M [25]	HT100M	ResNet152+ResNeXt [12, 34]	14.9 / 40.2 / 52.8	-	7.1 / 19.6 / 27.9	-	-	-	164.3	1100.0
ClipBERT [20]	COCO / CC	ResNet50 [17]	22.0 / 46.8 / 69.9	20.4 / 48.0 / 60.8	-	21.3 / 49.0 / 63.5	82.8 / 87.8 / 60.3	37.4	340.0	700.0
Frozen-in-Time [2]	CC / Webvid	Timesformer-B/16 [3]	31.0 / 59.8 / 72.4	**31.0 / 59.8** / 72.4	15.0 / 30.8 / 39.8	-	-	-	89.0	260.0
Ours$^{N=128}$	CC / Webvid	ViT-B/32 [6]	**32.6 / 62.1 / 71.6**	30.5 / 59.7 / **73.0**	**15.8 / 37.6 / 40.1**	**33.9 / 62.1 / 76.4**	**91.4 / 94.9 / 69.2**	**43.2**	**43.9**	**72.0**

Table 6. Full metrics of finetuning performance on text-to-video retrieval and video question answering benchmarks. We report R@1/R@5/R@10 for text-to-video retrieval tasks and report QA accuracy on the FrameQA task. *GFLOPs* shows the inference cost on a single sample, and *Time (ms)* indicates the average inference time across all samples on MSRVTT val split. For a fair comparison, we gray out 1) the models that use input modalities other than video and text (*e.g.*, audio) and 2) the models that use CLIP visual encoder [28] (the cross-attention layers of PERCEIVER-VL cannot be initialized with CLIP parameters and trained from scratch; see the discussion in Sec. 5.1). $^{N=128}$ means latent size N=128.

Model	Pretraining Datasets	Visual Backbone	Text-to-Image-to Retrieval ↑	QA Accuracy ↑		GFLOPs ↓	Time (ms) ↓
			Flickr30k (R@1/R@5/R@10)	VQAv2	NLVR^2 (dev/test-P)		
Models using additional object tag inputs							
VinVL-Base [36]	COCO / CC / SBU / Flickr / OI*	Faster-RCNN [36]	-	75.95	82.05 / 83.08	1023.3	800.0
OSCAR-Base [23]	COCO / CC / SBU / Flickr*	Faster-RCNN [1]	-	73.16	78.07 / 78.36	956.4	1000.0
UNITER-Base [4]	COCO / CC / SBU / VG	Faster-RCNN [1]	**72.5 / 92.4 / 96.1**	**72.70**	75.85 / 75.80	949.9	1000.0
ViLT-B/32 [18]	COCO / CC / SBU / VG	ViT-B/32 [6]	64.4 / 88.7 / 93.8	71.26	75.70 / **76.13**	55.9	32.0
Ours$^{N=128}$	COCO / CC / SBU / VG	ViT-B/32 [6]	62.4 / 87.1 / 93.2	71.62	75.45 / 75.53	**30.5**	**18.0**
LXMERT [31]	COCO / VG*	Faster-RCNN [1]	-	**72.42**	94.90 / 74.50	952.0	1100.0
VisualBERT [22]	COCO	Faster-RCNN [1]	-	70.80	67.40 / 67.00	425.0	1000.0
Pixel-BERT-R50 [15]	COCO / VG	ResNet50 [12]	53.4 / 80.4 / 88.5	71.35	71.70 / 72.40	136.8	150.0
Ours$^{N=128}$	COCO / VG	ViT-B/32 [6]	**61.7 / 86.7 / 92.1**	70.45	**73.30 / 74.87**	**30.5**	**18.0**
Frozen-in-Time [2]	CC / Webvid	Timesformer-B/16 [3]	61.0 / 87.5 / 92.7	-	-	63.9	70.0
Ours$^{N=64}$	CC / Webvid	ViT-B/32 [6]	61.0 / 86.6 / 93.0	70.12	74.04 / 74.52	**17.0**	**8.0**
Ours$^{N=128}$	CC / Webvid	ViT-B/32 [6]	**61.8 / 88.0 / 92.9**	**70.91**	**75.30 / 75.44**	30.5	18.0

Table 7. Finetuning performance on text-to-image retrieval and visual question answering benchmarks. For NLVR^2, we show Test-P accuracy. For Flickr30k, we show text-to-image retrieval R@1. Note that for brevity, we only show the image or video source datasets for *Pretraining Datasets*; the datasets that added additional text annotations are not included in the column (we use * to highlight them). For example, LXMERT is trained with image-text datasets COCO and VG, as well as the three QA datasets based on COCO and VG images, *i.e.*, VQAv2, VGQA and GQA. We also gray out models that use additional object tags in the first block and are not comparable to our model. *GFLOPs* shows the inference cost on a single sample, *Time (ms)* indicates the average inference time over all samples in VQAv2 minival split; For a fair comparison, we gray out models that are pretrained with more data. $^{N=128}$ means latent size N=128.

C.4. Two-stage training for CLIP weight initialization

In Table 5, we compare the two-stage and one-stage training for weight initialization form CLIP, as discussed in main paper Sec. 4.2. We use the architecture with latent size $N = 32$. We see significant improvement with two-stage training on MSRVTT and suggest the training strategy for weight initialization from transformer architecture such as CLIP.

D. Full Experiment Results

In Table 6 and Table 7, we provide the full experiment results with R@1/R@5/R@10 scores for retrieval tasks.

References

[1] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, 2018.

[2] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. *arXiv preprint arXiv:2104.00650*, 2021.

[3] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? *arXiv preprint arXiv:2102.05095*, 2021.

[4] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Learning universal image-text representations. In *ECCV*, 2020.

[5] 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.

[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.

[7] Angela Fan, Edouard Grave, and Armand Joulin. Reducing transformer depth on demand with structured dropout. *arXiv preprint arXiv:1909.11556*, 2019.

[8] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *ICCV*, pages 6202–6211, 2019.

[9] Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. Multi-modal transformer for video retrieval. In *ECCV*, pages 214–229. Springer, 2020.

[10] Zijian Gao, Jingyu Liu, Sheng Chen, Dedan Chang, Hao Zhang, and Jinwei Yuan. Clip2tv: An empirical study on transformer-based methods for video-text retrieval. *arXiv preprint arXiv:2111.05610*, 2021.

[11] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. In *ICML*, pages 1243–1252. PMLR, 2017.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[13] Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, R Channing Moore, Manoj Plakal, Devin Platt, Rif A Saurous, Bryan Seybold, et al. Cnn architectures for large-scale audio classification. In *ICASSP*, pages 131–135. IEEE, 2017.

[14] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *ICCV*, pages 4700–4708, 2017.

[15] Zhicheng Huang, Zhaoyang Zeng, Bei Liu, Dongmei Fu, and Jianlong Fu. Pixel-bert: Aligning image pixels with text by deep multi-modal transformers. *arXiv preprint arXiv:2004.00849*, 2020.

[16] Andrew Jaegle, Felix Gimeno, Andrew Brock, Andrew Zisserman, Oriol Vinyals, and Joao Carreira. Perceiver: General perception with iterative attention. *arXiv preprint arXiv:2103.03206*, 2021.

[17] Huaizu Jiang, Ishan Misra, Marcus Rohrbach, Erik Learned-Miller, and Xinlei Chen. In Defense of Grid Features for Visual Question Answering. In *CVPR*, 2020.

[18] Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. *arXiv preprint arXiv:2102.03334*, 2021.

[19] Alexander Kunitsyn, Maksim Kalashnikov, Maksim Dzabraev, and Andrei Ivaniuta. Mdmmt-2: Multidomain multimodal transformer for video retrieval, one more step towards generalization. *arXiv preprint arXiv:2203.07086*, 2022.

[20] Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L Berg, Mohit Bansal, and Jingjing Liu. Less is more: Clipbert for video-and-language learning via sparse sampling. In *CVPR*, 2021.

[21] Linjie Li, Yen-Chun Chen, Yu Cheng, Zhe Gan, Licheng Yu, and Jingjing Liu. Hero: Hierarchical encoder for video+ language omni-representation pre-training. In *EMNLP*, 2020.

[22] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*, 2019.

[23] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *ECCV*, 2020.

[24] Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. Hierarchical question-image co-attention for visual question answering. *NeurIPS*, 29, 2016.

[25] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *ICCV*, pages 2630–2640, 2019.

[26] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, pages 405–421. Springer, 2020.

[27] Shaobo Min, Weijie Kong, Rong-Cheng Tu, Dihong Gong, Chengfei Cai, Wenzhe Zhao, Chenyang Liu, Sixiao Zheng, Hongfa Wang, Zhifeng Li, et al. Hunyuan_tvr for text-video retrivial. *arXiv preprint arXiv:2204.03382*, 2022.

[28] 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. *arXiv preprint arXiv:2103.00020*, 2021.

[29] Andrew Rouditchenko, Angie Boggust, David Harwath, Brian Chen, Dhiraj Joshi, Samuel Thomas, Kartik Audhkhasi, Hilde Kuehne, Rameswar Panda, Rogerio Feris, et al. Avlnet: Learning audio-visual language representations from instructional videos. *arXiv preprint arXiv:2006.09199*, 2020.

[30] Kenneth O Stanley. Compositional pattern producing networks: A novel abstraction of development. *Genetic programming and evolvable machines*, 8(2):131–162, 2007.

[31] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. In *EMNLP*, 2019.

[32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[33] Qiang Wang, Yanhao Zhang, Yun Zheng, Pan Pan, and Xian-Sheng Hua. Disentangled representation learning for text-video retrieval. *arXiv preprint arXiv:2203.07111*, 2022.

[34] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017.

[35] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *ECCV*, pages 305–321, 2018.

[36] Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. Vinvl: Revisiting visual representations in vision-language models. In *CVPR*, pages 5579–5588, 2021.