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

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 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.

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
Table 1. Accuracy and inference time on VQAv2 with varied number of cross-attentions in PERCEIVER-VL encoder. We include the layer dropout probability $p_L$ 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 ($p_L = 0$) to ensure that the latent array always receives signal from the input.

<table>
<thead>
<tr>
<th># Cross-attentions in encoder</th>
<th>VQA2 Acc.</th>
<th>Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pretraining</td>
<td>Finetuning</td>
<td>Inference</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>1 ∼ 3 (0.5)</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>1 ∼ 3 (0.7)</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>1 ∼ 3 (0.5)</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1 ∼ 3 (0.5)</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>1 ∼ 3 (0.5)</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>1 ∼ 3 (0.5)</td>
<td>1 ∼ 3 (0.5)</td>
<td>1</td>
</tr>
<tr>
<td>1 ∼ 3 (0.5)</td>
<td>1 ∼ 3 (0.5)</td>
<td>3</td>
</tr>
</tbody>
</table>

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

In 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 VQA2 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, VQA2 and MSRVTT.

Pretraining Datasets | Modality | VQA2 | MSRVTT |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Random Init (Standard Gaussian)</td>
<td>✓</td>
<td>48.6</td>
<td>6.2</td>
</tr>
<tr>
<td>ImageNet-21k (ViT-B/32)</td>
<td>✓ ✓</td>
<td>62.3</td>
<td>12.1</td>
</tr>
<tr>
<td>ImageNet-21k (ViT-B/32) + CC</td>
<td>✓ ✓ ✓</td>
<td>68.2</td>
<td>24.6</td>
</tr>
<tr>
<td>ImageNet-21k (ViT-B/32) + Webvid</td>
<td>✓ ✓ ✓</td>
<td>67.5</td>
<td>25.1</td>
</tr>
<tr>
<td>ImageNet-21k (ViT-B/32) + CC + Webvid</td>
<td>✓ ✓ ✓</td>
<td>69.2</td>
<td>26.8</td>
</tr>
</tbody>
</table>

Table 3. Comparison of different pretraining datasets on VQA2 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).

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 (VQA2), and vice versa. The best performance is achieved by PERCEIVER-VL pre-trained 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 VQA2, 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.

Positional Encoding | Weight init |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Learned (default)</td>
<td>49.5</td>
</tr>
<tr>
<td>Fourier</td>
<td>49.7</td>
</tr>
</tbody>
</table>

Table 4. Comparison of different position encodings for latent array on VQA2.
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-CL are not initialized with CLIP parameters and trained from scratch; see the discussion in Sec. 5.1). N=128 means latent size N=128.

Table 7. Finetuning performance on text-to-image retrieval and visual question answering benchmarks. For NLVR2, 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 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. 4.4. 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


