Learning from Streaming Video with Orthogonal Gradients

Supplementary Material

This document provides additional materials including implementation details, analysis, ablation studies, and additional results that support the main paper.

6. More Implementation Details

The implementation details of three scenarios from the main paper Section 4 are shown in Table 4, Table 5 and Table 6.

	DoRA Pretrain	Linear probe
architecture	ViT-S/16	ViT-S/16
embedding dim	384	384
# heads	6	6
# blocks	12	12
encoder out_dim	65536	N/A
dataset	$\text{WT}_{\text{venice}}$	ImageNet
# local crops	6	N/A
# global crops	2	N/A
# input frames	8	N/A
input fps	1	N/A
input resolution	224×224	224×224
learning rate	0.0005	0.01
lr schedule	Warmup + Cosine	Warmup + Cosine
optimizer	N/A (varied)	SGD, m=0.9
weight decay	$0.04 \rightarrow 0.4$	0
learnable param	all	last layer
total batch size	32 clips	512 images
# epochs	1	100

Table 4. Implementation details of the DoRA experiments in the main paper Section 4.1.

	VideoMAE Pretrain	Linear/Attn probe
architecture	ViT-B/16	ViT-B/16
embedding dim	768	768
# heads	12	12
# blocks	12	12
dataset	K400 / SSV2	SSV2
mask ratio	0.9	N/A
# input frames	16	16
input fps	12	12
input resolution	224×224	224×224
learning rate	0.0003	0.0003
lr schedule	Warmup + Cosine	Warmup + Cosine
optimizer	N/A (varied)	AdamW
weight decay	0.05	0
learnable param	all	linear layer / attn block
total batch size	512 clips	32 clips
# iterations	260k	40k

Table 5. Implementation details of the VideoMAE experiments in the main paper Section 4.2.

	Future prediction
architecture	ViT-L/16
embedding dim	1024
# heads	16
# blocks	24
dataset	Ego4d / ScanNet
# input frames	4
input fps	30
input resolution	224×224
# output frames	4
prediction Δt	0.64s
learning rate	0.0001
lr schedule	Warmup + Cosine
optimizer	N/A (varied)
weight decay	1×10^{-5}
learnable param	all
# steps per update	16
# iterations	540k

Table 6. Implementation details of the Future prediction experiments in the main paper Section 4.3.

7. More Analysis on the Orthogonal Optimizer

This section provides more analysis to have a deep understanding about the orthogonal optimizer, in particular Orthogonal-AdamW.

Alternative Option: Downscale Learning Rate. Based on the main paper Figure 2 and Equation 4, readers might question whether the orthogonal optimizer is effectively using a smaller learning rate. We experiment with an alternative design choice that indeed reduces the learning rate based on gradient similarity. For example, one can re-scale the learning rate based on the similarity between the current and the previous gradient. Formally it can be written as,

$$\begin{aligned} \lambda &= 1 - \cos\left(g_t, g_{t-1}\right) \in [0, 2] \\ \theta_t &= \theta_{t-1} - \lambda \eta g_t \end{aligned} \tag{5}$$

where η is the learning rate and λ is the gradient multiplier. From the practical observation (*e.g.* the main paper Figure 3), we notice that $\cos\left(g_t, g_{t-1}\right)$ is mostly positive, therefore the learning rate multiplier λ mostly has a value within $[0, 1]$, having an effect of reducing the learning rate.

We apply the learning rate scaling method in Eq 5 to the AdamW optimizer, and name this variant as 'Slower-

	Ego4D: MSE↓ / PSNR↑	
optimizer	in-s.	out-of-s.
AdamW	0.034 / 15.9	0.026 / 16.8
Slower-AdamW	0.033 / 16.0	0.025 / 17.1
Orthogonal-AdamW	**0.031 / 16.4**	**0.023 / 17.6**

Table 7. Additional results on the future prediction task. The 'in-s.' and 'out-of-s.' denote in-stream results and out-of-stream results respectively, as same as the main paper Table 3.

AdamW'. The experimental results are shown in Table 7. The results show that the proposed Orthogonal-AdamW clearly outperform Slow-AdamW on both the in-stream and out-of-stream settings. Reducing learning rate as in 'Slower-AdamW' would avoid over-optimizing along one gradient direction, but it is insufficient to actually learn the new signals from correlated gradients. This result highlights that our method is different from only changing the learning rate based on the similarity between consecutive gradients.

seq. video processing	BS	optimizer	LP ↑	Attn ↑
batch-along-**time**	512	AdamW	16.4	46.1
batch-along-**time**	512	Orthogonal-AdamW	**18.4**	**48.0**
batch-along-**time**	256	AdamW	19.0	**47.8**
batch-along-**time**	256	Orthogonal-AdamW	**19.9**	47.7
batch-along-**time**	128	AdamW	**20.0**	**49.2**
batch-along-**time**	128	Orthogonal-AdamW	18.7	47.9
batch-along-**video**	512	AdamW	9.5	30.3
batch-along-**video**	512	Orthogonal-AdamW	**10.4**	**32.6**
batch-along-**video**	256	AdamW	8.3	25.7
batch-along-**video**	256	Orthogonal-AdamW	**13.2**	**37.6**
batch-along-**video**	128	AdamW	17.1	41.6
batch-along-**video**	128	Orthogonal-AdamW	**18.3**	**44.0**

Table 8. Effect of batch size on VideoMAE pretrained on SSV2 and evaluated on SSV2, using the same setting as the main paper Table 2. The 'BS' denotes Batch Size.

Impact of the Batch Size. The calculation of orthogonal gradients highly depends on the size of the mini batch. We analyze the impact of batch size on VideoMAE pretraining task on SSV2. The experimental results are shown in Table 8. Note that when reducing the batch size, we proportionally increase the number of training iterations to ensure each experiment is trained on the same number of samples. For example, comparing with BS $= 512$, the experiments using BS $= 256$ and BS $= 128$ are trained with $2\times$ and $4\times$ longer training schedules.

The experimental results show a few interesting trends. First, the absolute performance of 'batch-along-time' strategy does not change much with different batch sizes, and the Orthogonal-AdamW outperforms AdamW on larger batch sizes (512, 256), but underperforms AdamW on smaller batch size (128). Second, the VideoMAE trained with 'batch-along-time' strategy generally performs better with smaller batch size, and the Orthogonal-AdamW clearly outperforms AdamW on this setting.

Impact of the Momentum Parameter. In the main paper Equation 3, we introduce a hyper-parameter β controlling the update rate of the momentum. We experiment with different β values in Table 9. Generally, a large value of β (close to 1.0) leads to a 'smoother' momentum value; a lower value of β (close to 0.0) makes the momentum more fluctuate and less robust to noise, as the current value has large impact to the momentum. At the extreme case when $\beta = 0$, it means the momentum is not used. In our case, it means the orthogonal gradient is computed w.r.t. the previous gradient. The results in Table 9 shows that $\beta \geq 0.9$ works well, and there is almost no difference using 0.9 or 0.99. By default, we use $\beta = 0.9$ in the main paper experiments.

Orthogonal-AdamW	Ego4D: PSNR↑	
β	in-s.	out-of-s.
0	16.1	17.1
0.5	16.3	17.4
0.9	**16.4**	**17.6**
0.99	**16.4**	**17.6**

Table 9. Impact of the momentum parameter in Orthogonal-AdamW. This is a future prediction task on Ego4D-Stream, as same as the main paper Table 3.

optimizer	ImageNet top1↑
AdamW [11]	77.9
AdamW (repro)	77.8
Orthogonal-AdamW	76.5

Table 10. ImageNet classification results with ViT-B/16 architecture. The models are trained from scratch following the recipe in the original ViT paper [11]. Note that the first row is the official ViT result from [11]. 'repro' means our reproduction.

8. Does it Work on ImageNet Classification?

In the main paper Section 4, we have shown the Orthogonal-AdamW outperforms AdamW on various self-supervised video learning scenarios, even on *shuffled* video clips (VideoMAE results in the main paper Table 2). Naturally, we would like to know if the orthogonal optimizer can be applied to general *supervised learning* tasks. In this section, we compare Orthogonal-AdamW with AdamW on the classic ImageNet classification task. Results are shown in

Table 10. We use a ViT-B/16 architecture and follow the training recipe from [11]. First, our reproduction matches the reported ViT performance on ImageNet (77.8 vs 77.9). Second, we find the Orthogonal-AdamW underperforms AdamW by 1.3% on this task. It is probably because ImageNet mini-batches follow IID distributions more closely, and the gradients from consecutive batches have negligible correlation. In this case, optimizing the orthogonal gradients does not bring informative learning signals.