LayerLock: Non-collapsing Representation Learning with Progressive Freezing

Supplementary Material

In this supplementary material we provide additional details on pretraining, evaluation protocol, tasks, readouts and baselines. We also present details of the layer convergence analysis that partly motivates our approach.

A. Additional details

A.1. Layer Convergence Analysis

We describe here the setup used for the convergence analysis shown in Fig. 1. We start with the basic setup used by the 4DS MAE models in [10]. However, due to the large number of experiments required, we use a small model with a relatively short training schedule.

Specifically, we start with a modified ViT-S encoder that has 16 layers. We train the baseline and all layer-freezing ablations for 14K training steps, with batch size 2048. We use the same optimizer settings described below but with a learning rate of 1e-4.

To remove noise in the loss convergence calculations, we take the "final loss" as the average loss in the final 1000 steps of training.

A.2. Pretraining

In Tab. 8 we provide the hyperparameters used to train our 4DS models. As mentioned in the main text, these models were trained on 1B examples. Note during optimization, we do not apply weight decay on layer norm parameters and biases of linear layers.

Tab. 9 provides the hyperparameters used to train our V-JEPA models. These largely follow the hyperparameters in the original paper (scaled for the slightly smaller batch size of 2048 as opposed to 3072) with all models trained on 560M examples from the Kinetics700 dataset [23]. Note this is different from the training dataset in the original paper, which combined multiple Kinetics datasets into a single larger dataset. Also note that even though our training setup is the same as the original paper, we use the same evaluation setup for all models (which is different from the original paper) and hence our V-JEPA numbers are not directly comparable with the ones in the original paper. In contrast to the 4DS models, we use learned positional embeddings instead of RoPE for V-JEPA models, and we do all computations in *float32* precision.

Note for V-JEPA, we still use an EMA of the encoder to compute the targets. This is not strictly necessary to avoid representation collapse as the progressive freezing employed by LayerLock helps avoid such a problem. However, our initial experiments have shown that using an EMA target network results in better downstream performance.

Hyperparameter	
Num. training steps	488,282
Input resolution	224×224
Learning rate	3e-4
Warmup steps	10,000
#N decoding layers	4
Patch size	2×16×16
Minimum resize factor	1.15
Batch size	2048
MAE masking ratio	0.95
AdamW weight decay	0.05
AdamW b1	0.90
AdamW b2	0.95
Num. of steps before first freezing	160,000 - 200,000
Num. steps between freezing	10,000
Num. layers to freeze	1
Target layers	(1, 2, 3, . . . , 32)
RoPE max. wavelength	10,000
RoPE T, H, W proportions	10%, 25%, 25%

Table 8. Pretraining hyperparameters for 4DS ViT-G and ViT-e models trained on 1B examples. Note all hyperparameters are the same for both models except the number of steps before freezing. Minimum resize factor controls how much a video's minimum side gets resized before cropping as function of input resolution.

A.3. Ablation studies

Due to resource constraints, we used models of different sizes trained on different number of examples for ablations. In Tab. 10, we provide the hyperparameters used to train these.

A.4. Evaluation

Our evaluation closely follows the setup in [10], which we discuss in summary here. See the supplementary material in [10] for more details.

As mentioned in the main text, we evaluate all models by training a readout module on top of frozen features. We use 1.28M training examples for each task with a batch sie of 32. We sweep over learning rates (1e-4, 3e-4, 1e-3) and readout depth fraction (0.25, 0.5, 0.75, 0.85, 0.95, 1.0) and report the best results. We use a cosine schedule for the learning rate with linear warmup of 1K steps and a decay to 1e-7. We use a similar cross-attention readout architecture for all tasks, where a set of learned tokens cross-attend to frozen features. A summary of the readout configurations and number of parameters in each case is provided

Hyperparameter	
Num. training steps	262,501
Input resolution	224×224
Initial learning rate	1.3e-4
Learning rate	4.17e-4
End learning rate	6.6e-7
Warmup steps	90,000
Patch size	2×16×16
Stride	4
Minimum resize factor	1.15
Batch size	2048
Target network EMA coef.	0.998
AdamW weight decay	0.04 → 0.4
Multiblock masking parameters	
Num. blocks	8
Block area range	(0.3, 0.3)
Aspect ratio range	(0.75, 1.50)
Num. of steps before first freezing	100,000
Num. steps between freezing	6,000
Num. layers to freeze	1
Target layers	(1, 2, . . . , 24)

Table 9. Pretraining hyperparameters for V-JEPA trained on 560M examples.

in Tab. 11.

In the following, we provide a short description of each evaluation task.

A.4.1. Something-Something v2 action classification

The SSv2 action classification dataset contains 220,000 videos with duration ranging from 2 to 6 seconds at 12fps. Videos contain 174 human actions with everyday objects.

Task definition. Given a video clip of 16 frames of resolution 224x224 with stride 2, the model is tasked to predict an action class. Top-1 accuracy is used to measure the performance.

Readout details. The cross-attention readout module uses 768 parameters with 12 heads and a single learned query to predict logits for 174 classes. In training, we resize the shorter size of the video to 239 and take random temporal crop of shape 224x224 from it. We use colour augmentation with 0.8 probability of randomly adjusting the brightness, saturation, contrast and hue (see Tab. 12), and a 0.1 probability of converting to grayscale. In test time we take one 224x224 central crop from the video without any colour augmentation.

A.4.2. Kinetics700 action classification

The SSv2 action classification dataset contains 545,317 10s video clips from 700 action classes.

Task definition. Given a video clip of 16 frames of resolution 224x224 with stride 2, the model is tasked to predict an action class. At test time 7 equally spaced clips are passed through the trained classifier and their softmax scores are averaged to get predictions. Top-1 accuracy is used to measure the performance.

Readout details. The cross-attention readout module uses 1024 parameters with 16 heads and a single learned query to predict logits for 700 classes.

A.4.3. ScanNet depth estimation

ScanNet [12] is a video dataset captured in various indoor environments, containing rich annotations for 3D camera poses, surface reconstructions, and instance-level semantic segmentations. Data was obtained through an RGB-D capture system that produces depth. RGB frames have 1296x968 resolution while depth frames have 640x480 resolution. There are 1201 videos in the train split, 312 videos in the validation split, and 100 videos in the test split. We use the train and validation splits of ScanNet for this paper.
Task definition. We feed the models 16 RGB frames while adding readout heads on top to output depth for each input frame. We scale the images to the (0, 1) range and mask out target depth values outside of (0.001, 10) meters. We perform random cropping and left-right flipping during training and take a center crop during testing.
Evaluation metrics. We follow prior work on monocular depth estimation and report the mean of the absolute relative error (AbsRel) [38, 47, 53] which is computed as $|d^* - d|/(d + \epsilon)$ where d^* is the predicted depth values, d is the ground truth depth.
Readout details. We use a cross-attention readout head with 1024 parameters and 16 heads with one learned query for each spatio-temporal patch in the input video. We use a patch size of $2 \times 8 \times 8$ and predict 128 ($= 2 * 8 * 8$) depth values for each patch, one for each pixel. We use an L2 loss.

Hyperparameter	ViT-G, 250M	ViT-G, 56M	ViT-H, 100M	ViT-B, 50M
Learning rate	1e-4	1e-4	1e-3	3e-4
Warmup steps	5,000	5,000	1,000	2,000
Batch size	2048	128	8096	512
Weight decay	No	No	No	1e-3
Num. of steps before first freezing	25,000	19,000	–	6,000
Num. steps between freezing	20,000	30,000	–	4,000
Num. layers to freeze	5	5	–	2
Target layers	(4, 8, 12, 16)	(4, 8, 12, 16, . . . , 44)	–	(1, 3, 5, 7)

Table 10. Hyperparameters for models used in ablation studies. Only hyperparameters that are different than **??** shown.

Eval	Architecture	Number of Params

SSv2 Classification	`LearnedQueries(num_channels=768)`
	`CrossAttention(qkv_size=768, num_heads=12)`
	`Linear(output_size=174)`
		7M

Kinetics Classification	`LearnedQueries(num_channels=1024)`
	`CrossAttention(qkv_size=1024, num_heads=16)`
	`Linear(output_size=700)`
		7M

ScanNet Depth Prediction	`LearnedQueries(num_channels=1024)`
	`CrossAttention(qkv_size=1024, num_heads=16)`
	`Linear(output_size=128)`
		18M

Table 11. Configurations and number of parameters of cross-attention-based readout modules used in this paper, for different tasks. Note that the number of parameters are given for the case of a ViT-L backbone, that has 1024-channel outputs.

Brightness	delta in [-0.125, 0.125]
Saturation	factor in [0.6, 1.4]
Contrast	factor in [0.6, 1.4]
Hue	delta in [-0.2, 0.2]

Table 12. Hyper-parameters for color augmentation used when training readout heads on the SSv2 task. Deltas are added to the corresponding channel, while factors multiply the corresponding channel.