Asymmetric Masked Distillation for Pre-Training Small Foundation Models Supplementary Material

Zhiyu Zhao[1,2] Bingkun Huang[1,2] Sen Xing[2] Gangshan Wu[1] Yu Qiao[2] Limin Wang[1,2]
[1] State Key Laboratory for Novel Software Technology, Nanjing University [2] Shanghai AI Lab

In this supplementary material, we provide more details of our AMD and present more experiment results. Specifically, we provide more implementation details in our experiments in Section A. Then, we present the differences between the different alignment methods in Section B. After this, We discussed on the reconstruction task in Section C. Then, we continued our analysis of the teacher performing no masking in Section E. Next, we discussed the masking of the student model in Section D. Finally, we continued our analysis of the comparison with DMAE in Section F.

A. Implementation Details

We conduct the experiments with 32 A100-80G GPUs for pre-training on SSV2 and K400. Additionally, we fine-tune the SSV2 with 16 GPUs, the K400 and the AVA with 32 GPUs. All ablation experiments conduct with 16 GPUs. The experiments on UCF101 and HMDB51 both worked with 8 GPUs. Our implementation is based on VideoMAE [9] and follows the data augmentation settings of pre-training and fine-tuning. To speed up model training and improve the stability, we perform the repeated sampling [5]. The training schedule we give is the total number of times a sample has been sampled. The pre-training settings on the SSV2 and K400 datasets are shown in Table S1.

SSV2. We pretrain AMD on SSV2 for 800 epochs by default. For fine-tuning, we perform the spare sampling [10] and report the 2 clips $\times 3$ crops evaluation results and the settings are shown in Table S2.

K400. We pretrain AMD on K400 for 800 epochs by default. For fine-tuning, we report the 5 clips $\times 3$ crops evaluation results and the settings are shown in Table S2.

HMDB51 and UCF101. We only fine-tune the model pre-trained on K400 to the HMDB51 and UCF101 dataset. We report the 5 clips $\times 3$ crops evaluation results and the settings are shown in Table S3.

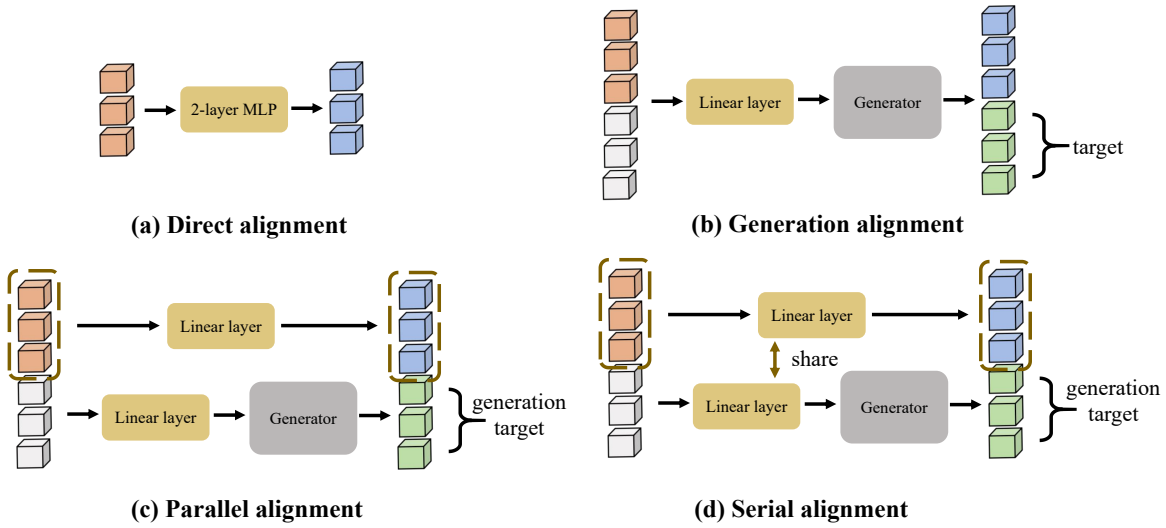
AVA. We refer to the most classic two-stage structure to detect key frames of the video. In the first stage, we use the box detected in AIA [8]. While in the second stage, we use the ViT backbone to classify the objects detected in the first stage. Follwing VideoMAE, the short side size of the input is resized to 256 pixels. The ground-truth person boxes are only used for training. In term of the inference, we use the detected boxes with confidence ≥ 0.8. The settings are shown in Table S3.

config	**SSV2**	**K400**
optimizer	AdamW	
learning rate	1.2e-3	
weight decay	0.05	
optimizer momentum	β_1, β_2=0.9, 0.95 [2]	
batch size	2048	
repeated sampling [5]	4	
learning rate schedule	cosine decay [6]	
epochs	800	
warmup epochs	40	
sampling rate	2	4
flip augmentation	*no*	
augmentation	MultiScaleCrop [10]	

Table S1. AMD pre-training setting for both ViT-S and ViT-B backbone.

config	**SSV2**	**K400**
optimizer	AdamW	
base learning rate	1e-3 (S), 5e-4(B)	1e-3 (S),7e-4 (B)
weight decay	0.05	
optimizer momentum	β_1, β_2=0.9, 0.999	
batch size	512 (S,B)	512 (S),1024 (B)
learning rate schedule	cosine decay [6]	
warmup epochs	5	
training epochs	40 (S), 30 (B)	150 (S), 90 (B)
sampling rate	*sparse* [10]	4
repeated sampling [5]	2	
flip augmentation	*no*	*yes*
RandAug [3]	(9, 0.5)	
label smoothing [7]	0.1	
mixup [12]	0.8	
cutmix [11]	1.0	
drop path	0.1	
head dropout	*None*	
layer-wise lr decay [1]	0.7 (S),0.75 (B)	0.75 (S,B)

Table S2. AMD fine-tuning setting of SSV2 and K400.



(a) Direct alignment **(b) Generation alignment**

(c) Parallel alignment **(d) Serial alignment**

Figure S1. We apply four feature alignment methods in our work, with the serial alignment being our default setting.

config	HMDB51	UCF101	AVA
optimizer		AdamW	
base learning rate	1e-3	5e-4	2.5e-4
weight decay		0.05	
optimizer momentum		β_1, β_2=0.9, 0.999	
batch size	128	256	128
learning rate schedule		cosine decay [6]	
warmup epochs		5	
training epochs	60	100	30
sampling rate	2	4	4
repeated sampling [5]	2	2	*no*
flip augmentation		*yes*	
RandAug [3]	(9, 0.5)	(9, 0.5)	–
label smoothing [7]	0.1	0.1	–
mixup [12]	0.8	0.8	–
cutmix [11]	1.0	1.0	–
drop path	0.1	0.2	0.2
head dropout	0.5	0.5	*None*
layer-wise lr decay [1]	0.75	0.70	0.75

Table S3. Fine-tuning setting of HMDB51, UCF101 and AVA.

B. Different alignment methods Details

Four alignment strategies are described in this paper, with specific structural details described below and the structures are shown in Figure S1.

Direct alignment. We employ a 2layer MLP for alignment, where the hidden layer has the same dimension as the teacher model and the activation function is GELU [4]. Additionally, the features to be aligned have not been normalised.

Generation alignment. We applied a decoder-like generator to align teacher features, where the number of `[MASK]` tokens is the number of tokens that the teacher has more than the student. A linear projection layer is needed to align the teacher's dimension before the student features are fed into the generator. And the features used to calculate the alignment loss is also those features that the teacher have more of than the student.

Parallel alignment. We have combined the two alignment methods in a parallel way, where the direct alignment part uses only a simple linear projection layer. It is worth noting that the projection layer of the two alignment methods do not share parameters.

Serial alignment. We combine the two alignment methods in a serial way as our default setting, and the two aligned linear projection layers share parameters, which can reduce the difficulty of generation alignment.

Method	Model	Epochs	Reconstruction	Top-1
AMD	ViT-B	800	✗	73.0
AMD	ViT-B	800	✓	**73.3**

Table S4. We compared the results with and without the reconstruction task with 800 epochs of training.

C. Discussion on reconstruction task

To verify the effect of the reconstruction task in distillation, we have made a comparison in Table S4. The results show that distillation using the reconstruction task performs better. We consider that the reconstruction task provides a regularisation for model distillation and allows students to learn more semantic information that is beneficial for generalisation, which also allows AMD to benefit from a longer training schedule.

D. Analysis of the masking of the student

Note that the VideoMAE's optimal masking ratio on the reconstruction task is 90%, and AMD also focuses on the

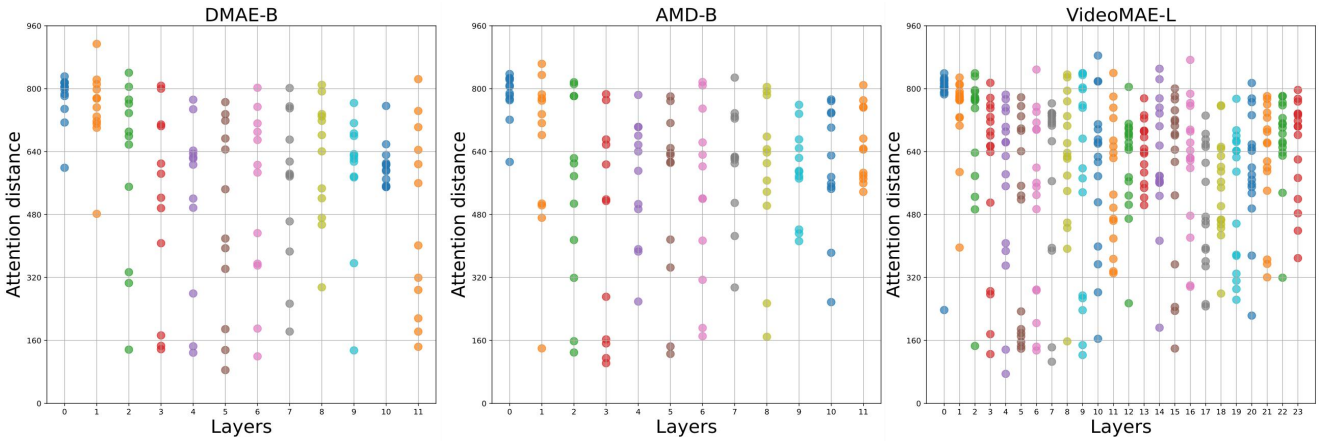


Figure S2. **The average attention distance in different attention heads at each layer depth.** Distances are calculated over 16 frames, and frame spacing is calculated over the maximum distance of each frame. Results are averaged over SSV2 test set.

pre-training, so we fixed the masking ratio of the student model at 90% in order not to damage the reconstruction difficulty. We supplement an experiment with a student masking ratio of 80% and a teacher masking ratio of 75%, whose accuracy is 73.1% after 800 epochs of training, lower than the default setting (73.3%).

E. Analysis of teacher performing no masking

We found that the performance degradation occurs when the teacher masking is extremely low. We think that there might be a conflict in our training goals. The conflict becomes more apparent under an extremely low masking ratio of the teacher model. Our training aims to do two things: 1) reconstructing the image pixels and 2) aligning with the teacher's features. However, a low masking ratio in the teacher model means it covers more global information. This can lead to a mismatch with the student model's reconstruction task. We have noticed that the reconstruction loss rises when the masking ratio of the teacher model becomes quite low, which may support our conjecture about the conflict.

Furthermore, we supplemented a experiment with a teacher's masking ratio of 25%, which resulted in 72.3%. So the peak in accuracy might occur roughly at a teacher masking ratio of 45%. However, when the teacher's masking ratio was reduced from 60% to 45%, its training cost increases but its gain is very limited. So we suggest choosing the teacher's masking ratio from efficiency considerations.

F. Comparison with DMAE

Overall, the two main differences between AMD and DMAE are asymmetric masking and generation alignment which are discussed in Table 1. In addition, to understand the distinct impacts of the AMD and DMAE masking distillation

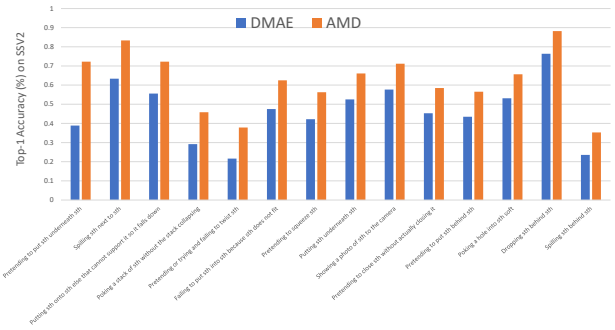


Figure S3. **Detailed breakdown of accuracy comparison between AMD and DMAE by categories.** We checked the performance gap on SSV2 in terms of categories on the test set.

strategies on the video model pre-training, we provided a comparison of 14 categories in SSV2 with the most accuracy difference in Figure S3. It shows that AMD has a stronger ability to infer object interactions, spatial relationships, and action outcomes.

Furthermore, we examined the average attention distance of DMAE, AMD and VideoMAE (the teacher model) to reveal the properties of models in Figure S2. We find that at shallow layers, each model has diverse attention heads which means model's attention is both local and global. While at deep layers especially the last layer, most attention heads of AMD and VideoMAE tend to extract global informations, which is different from DMAE. In the video domain, the more global attention means that the model is able to capture more global information about the action, which is beneficial for action recognition. Therefore AMD is better at tasks that require temporal understanding, which is due to the asymmetric masking strategy that allows the teacher model to see more contextual information.

References

[1] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. BEit: BERT pre-training of image transformers. In *International Conference on Learning Representations*, 2022. 1, 2

[2] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *International Conference on Machine Learning*, 2020. 1

[3] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020. 1, 2

[4] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv: Learning*, 2016. 2

[5] Elad Hoffer, Tal Ben-Nun, Itay Hubara, Niv Giladi, Torsten Hoefler, and Daniel Soudry. Augment your batch: Improving generalization through instance repetition. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8126–8135, 2020. 1, 2

[6] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 1, 2

[7] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016. 1, 2

[8] Jiajun Tang, Jin Xia, Xinzhi Mu, Bo Pang, and Cewu Lu. Asynchronous interaction aggregation for action detection. In *European Conference on Computer Vision*, 2020. 1

[9] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In *Advances in Neural Information Processing Systems*, 2022. 1

[10] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks for action recognition in videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 1

[11] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *IEEE/CVF International Conference on Computer Vision*, 2019. 1, 2

[12] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 1, 2