Masked Autoencoders are Secretly Efficient Learners

Zihao Wei[1] Chen Wei[2] Jieru Mei[2] Yutong Bai[2] Zeyu Wang[3] Xianhang Li[3]
Hongru Zhu[2] Huiyu Wang[4] Alan Yuille[2] Yuyin Zhou[3] Cihang Xie[3]

[1]University of Michigan, Ann Arbor [2]Johns Hopkins University [3]UC Santa Cruz [4]Meta

Abstract

This paper provides an efficiency study of training Masked Autoencoders (MAE), a framework introduced by He et al. [13] for pre-training Vision Transformers (ViTs). Our results surprisingly reveal that MAE can learn at a faster speed and with fewer training samples while maintaining high performance. To accelerate its training, our changes are simple and straightforward: in the pre-training stage, we aggressively increase the masking ratio, decrease the number of training epochs, and reduce the decoder depth to lower the pre-training cost; in the fine-tuning stage, we demonstrate that layer-wise learning rate decay plays a vital role in unlocking the full potential of pre-trained models. Under this setup, we further verify the sample efficiency of MAE: training MAE is hardly affected even when using only 20% of the original training set.

By combining these strategies, we are able to accelerate MAE pre-training by a factor of 82 *or more, with little performance drop. For example, we are able to* pre-train a ViT-B in ~9 hours using a single NVIDIA A100 GPU *and achieve 82.9% top-1 accuracy on the downstream ImageNet classification task. Additionally, we also verify the speed acceleration on another MAE extension, SupMAE.*

1. Introduction

Masked Image Modeling (MIM) [1, 13] is a powerful self-supervised pretext task that trains a model to predict masked signals, such as raw pixels or semantic tokens, based on visible regions of an image. The most recent instantiation of MIM, the Masked Autoencoder (MAE)[13], has played a pivotal role in enabling the successful pre-training of data-intensive Vision Transformers (ViTs)[10]. Subsequently, these pre-trained ViTs demonstrate state-of-the-art performance on a wide range of downstream recognition tasks and out-of-distribution tests.

However, despite its impressive performance, MIM pre-training typically involves a substantial computational cost. For instance, BEiT [1] requires a long schedule of 800 pre-training epochs on the ImageNet-1k dataset to attain competitive performance. The follow-up work MAE [13], which utilizes a high masking ratio and an asymmetric encoder-decoder design, substantially accelerates pre-training; however, MAE still requires an excessive schedule of 1,600 pre-training epochs. This elevated pre-training overhead impedes the wider exploration of MIM in the ultra-large-computation regime, which is a key factor in unlocking the emerging properties of deep learning systems [23, 24].

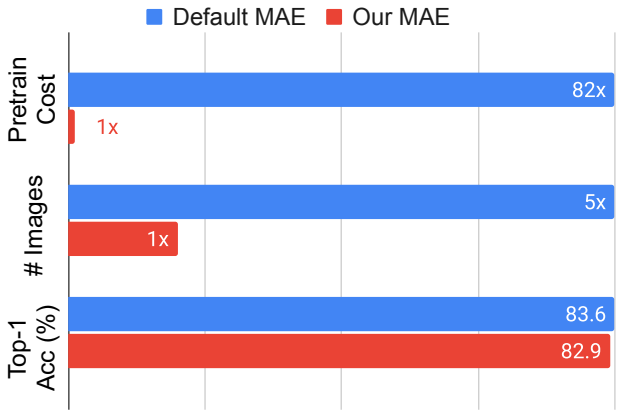


Figure 1. We show that MAE can actually pre-train at a much faster speed on a smaller amount of images, but still attain a competitive accuracy on ImageNet with ViT-B.

In this paper, we embark on a systematic investigation into the efficiency of training MAE. Particularly, our motivation stems from a simple yet intriguing observation: aggressively lightening the pre-training setups of MAE (*i.e.*, decreasing the number of pre-training epochs from 1600 to 100, increasing the masking ratio from 75% to 90%, and reducing the decoder depth from 8 to 1) . In essence, models that undergo light MAE pre-training are nearly as adept at modeling masked signals as their fully pre-trained counterparts. Based on this observation, *we posit that these lightly pre-trained models should yield comparable performance when fine-tuned to different downstream tasks*.

Interestingly, contrary to our initial assumption, we observe a significant decrease in performance on ImageNet classification when fine-tuning lightly MAE pre-trained models using the official setup from [13]. Specifically, top-1 accuracy drops from 83.6% to 80.4% with ViT-B. However, this decrease in accuracy can be mitigated by a longer fine-tuning period. For instance, by doubling the fine-tuning duration from 100 to 200 epochs, the accuracy increases to 81.8% (+1.4%). This phenomenon suggests that our lightly MAE pre-trained model does not fully converge with the original setup, and invites us to ponder the following question: *Can we develop a fine-tuning recipe that enables lightly MAE pre-trained models to converge more effectively and achieve performance comparable to fully pre-trained models?*

We identify that the appropriate usage of **Layer-wise Learning Rate Decay** (LLRD), a hyperparameter that scales the learning rate of each layer differently, is a key factor in effective fine-tuning of lightly MAE pre-trained models. LLRD was first introduced by LAMB for BERT [8, 30] and then applied to the Vision Transformers by BEiT [1] to secure high fine-tuning performance. With LLRD, the learning rate of each layer decreases from top to bottom, with the learning rate of the first layer being the smallest and that of the last layer being the largest. For instance, in the official MAE fine-tuning recipe, a small scaling factor of 0.65 is adopted for ViT-B encoder, making the learning rate of the first layer less than 1% of that of the last layer. The main motivation behind LLRD is that the encoder has already learned strong low-level features during pre-training, rendering any significant changes to early layers unnecessary during fine-tuning. However, given the aggressive nature of our pre-training strategy, resulting feature representations may not fully converge. Consequently, the default setup of LLRD, which restricts the learning rate to be small (particularly for early layers), may not be optimal for fine-tuning.

Therefore, we propose a new fine-tuning recipe for lightly pre-trained MAE. Specifically, we adapt a larger value for LLRD to effectively improve fine-tuning accuracy. By increasing the LLRD from 0.65 to 0.75, a lightly MAE pre-trained ViT-B, which is $\sim$82$\times$ *faster than its vanilla counterpart*, achieves a top-1 ImageNet accuracy of 82.9%, resulting in a performance boost of 1.4%. Furthermore, we demonstrate the enhanced sample efficiency of our approach, as it learns effective feature representations even when only 20% of the original training set is available. Finally, we showcase the generalizability of our findings to another MAE-based method, SupMAE, through a set of ablation studies and provide a simple and efficient strategy for tuning LLRD. We hope our findings can benefit future research in studying the efficiency of MIM, or self-supervised learning in general.

2. Related Works

Hand-crafted self-supervised learning. In the field of visual self-supervised learning, the goal is to train models to learn effective feature representations using supervision signals derived from the images or videos themselves, without the need for manual annotation. Early methods in this area employed a variety of hand-crafted pretext tasks to provide this supervision. Representative examples of such tasks include image colorization [18], image inpainting [22], solving jigsaw puzzles [21, 27], predicting rotations [17], and temporal information verification [20]. However, these hand-crafted pretext tasks generally are less effective than contrastive learning and masked image modeling, which we will review next.

Contrastive Learning. Contrastive learning is a widely adopted self-supervised learning paradigm that utilizes the task of distinguishing different views of the same image from other images [3, 5, 7, 12, 14]. Its core principle is to train the model to pull positive sample pairs (*e.g.* views from the same image) closer together and push negative sample pairs (*e.g.* views from different images) farther apart in the feature space [6]. The use of Siamese architectures has proven to be an essential element in the success of contrastive learning [6]. Recently, ViT has also been introduced in the field of contrastive learning, utilizing the class token to represent the entire image [2, 7].

Masked Image Modeling. The masked image modeling paradigm, which builds upon the success of the masked language modeling approach in Natural Language Processing, has been adapted for use in the visual domain. Early works in this field include IGPT [4], which learns image representations by regressing images pixel by pixel, and BEiT [1], which encodes image patches into semantic tokens and trains the model to predict them. Subsequently, studies such as MaskFeats [26] and MFM [28] have investigated the efficacy of predicting features such as masked HOG or masked frequency.

The work of MAE [13] has further demonstrated the effectiveness of a simple raw pixel reconstruction objective in training models to learn effective representations. Notably, it's crucial to set a large masking ratio (*e.g.* 75%) for these tasks to avoid simple extrapolation from visible neighboring patches. Additionally, MAE's asymmetric encoder-decoder architecture, which only takes unmasked image patches as input to the encoder, results in significant training acceleration. Further developments such as SupMAE [19] and VideoMAE [11, 25] introduce golden labels and extend the method to the video domain respectively. In this work, instead of proposing new designs, we aim to investigate the underlying efficiency of the vanilla MAE approach.

3. Revisiting Masked AutoEncoders

Our work is built upon the foundation of MAE [13], with the goal of further reducing training cost while preserving effectiveness. In this section, we first provide a brief overview of MAE, and then review Layer-wise Learning Rate Decay, the key factor that enables extremely efficient pre-training with MAE.

3.1. Masked AutoEncoders

In essence, MAE pre-training involves masking random patches of the input image and reconstructing the missing pixels from remaining visible patches. The main components of MAE include:

Masking strategy. By using ViT [9], MAE operates on non-overlapping image patches. A small subset of the embedded patches are randomly sampled without replacement while the rest of the patches are masked and used as the prediction target of the decoder. In MAE [13], it has been observed that a high masking ratio (*e.g.*, 75%) is crucial in preventing shortcut learning (by simply extrapolating from visible neighboring patches).

MAE encoder. The MAE encoder is a vanilla ViT [9] that takes only visible patches as input. This design significantly reduces training time and memory cost, especially when combined with a high masking ratio. For instance, it leads to more than $4\times$ training acceleration with a 75% masking ratio.

MAE decoder. The MAE decoder is another vanilla ViT [9] that processes both visible patches and mask tokens. Note that the MAE decoder is solely used for the image reconstruction task in the pre-training stage, and is typically much narrower and shallower than the encoder. The default MAE decoder depth is 8, but it has been observed that a 1-layer decoder is sufficient to produce compelling results [13].

3.2. Layer-wise Learning Rate Decay

Layer-wise learning rate decay (LLRD) strategy is a technique that adaptively adjusts the learning rate of each layer in a deep neural network. It was first introduced by LARS to facilitate the training of ResNet with a large batch size [29]. Subsequently, it was simplified by ELECTRA [8] and applied to fine-tune MIM pre-trained models in BEiT [1].

LLRD involves multiplying the learning rate of each layer by a scaling factor, which is computed using a scaling function that takes as input the layer weights. In BEiT, a simple polynomial function is used to calculate the scaling factor. In a model with h layers, the learning rate of the i-th layer, η^i, is given by:

$$\eta^i = \eta\,\alpha^{(h-i)}, \tag{1}$$

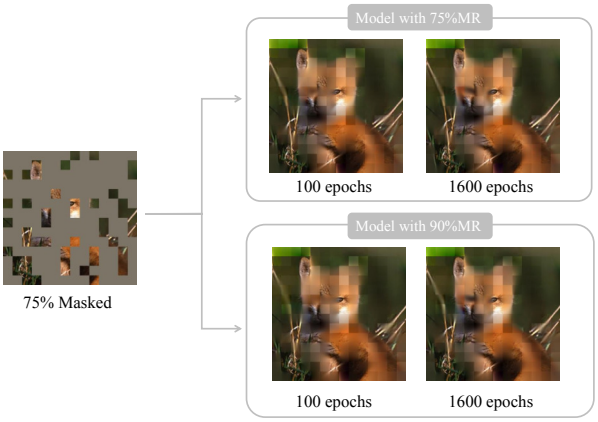


Figure 2. **Comparison on the reconstructed images.** We show the reconstruction results of a certain masked image from models pre-trained for different epochs and masking ratios (MR). We observe that a short pre-training schedule can achieve close reconstruction quality to that of a much longer pre-training schedule, for both 75% and 90% MR.

where η is the overall learning rate provided by the optimizer and α is the hyper-parameter that controls the decay rate of the learning rate from the last layer to the first layer.

The default value of α for ViT-B in the fine-tuning recipe of BEiT and MAE is 0.65, a value specifically designed for slower updates of low-level features. However, we observe that in our lightly pre-trained models, the low-level features need to be updated in a more rapid manner for better convergence. Therefore, selecting a suitable LLRD becomes crucial to unlock the potential of efficient MAE learning.

4. Efficient Training with MAE

In this section, we conduct a systematic investigation into the efficiency of training MAE. We first introduce the default MAE setup and our modifications to accelerate pre-training. We then progressively devise key factors in our fine-tuning recipe, including Layer-wise Learning Rate Decay and fine-tuning learning rate, to attain a comparable performance with fully MAE pre-trained models.

4.1. Pilot Study Setup

We follow MAE's pre-training and fine-tuning setup outlined in Tables 1a and Table 1b. We use the ViT-B architecture as the backbone and evaluate the performance using the top-1 accuracy metric on the ImageNet-1k dataset. To improve the efficiency of our model, we use a one-layer decoder as the default setting. This design choice is motivated by the fact that using an eight-layer decoder, while providing only marginal improvements, consumes over 50% of the total FLOPs in the MAE model [13]. In practice, switching to a one-layer decoder setup leads to an acceleration of over 60% compared to the speed of an eight-layer decoder.

config	MAE	ours
optimizer	AdamW	AdamW
base lr	1.5e-4	1.5e-4
weight decay	0.05	0.05
optimizer	β_1, β_2=0.9, 0.95	β_1, β_2=0.9, 0.95
batch size	4096	1024
lr schedule	cosine	cosine
warmup epochs	40	5
full epochs	1600	100 or 200
masking ratio	75%	75% or 90%
decoder depth	8	1
augmentation	RRC	RRC

(a) Pre-training

config	MAE	ours
optimizer	AdamW	AdamW
base lr	5e-4	1e-3
weight decay	0.05	0.05
optimizer	β_1, β_2=0.9, 0.999	β_1, β_2=0.9, 0.999
layer-wise lr decay	0.65	0.82 (100 / 90)
batch size	1024	1024
lr schedule	cosine	cosine
warmup epochs	5	5
training epochs	100	100
augmentation	RandAug (9, 0.5)	RandAug (9, 0.5)
label smoothing	0.1	0.1
mixup	0.8	0.8
cutmix	1.0	1.0
drop path	0.1	0.1

(b) End-to-end fine-tuning

Table 1. **Hyper-parameter comparison.** The differences between MAE's default setting and our recipe are bolded. We use an LLRD found by our low-cost parameter searching w.r.t. each pre-train setup, *i.e.*, pre-train epochs and masking ratios.

fine-tune epochs	top-1 acc. (%)
100	80.4
200	81.8

Table 2. **Aggressive training schedule results in not converged pre-trained models.** Increasing the fine-tuning epochs boosts the top-1 accuracy by 1.4%, indicating the pre-trained model is far from convergence.

batch size	top-1 acc. (%)
512	81.8
1024	81.6
4096	80.4

Table 3. **MAE pre-training with different batch sizes.** We note that a smaller batch size leads to better convergence for shortly pre-trained models.

We begin by aggressively reducing the number of pre-training epochs from 1600 to 100 and increasing the masking ratio from 75% to 90%, which further speeds up the MAE training by 23×. Based on our analysis of the reconstruction quality, as shown in Figure 2, we can see that the MAE model is able to decently reconstruct images even with this light pre-training recipe. This visualization leads us to conjecture that this lightly pre-trained MAE model would perform similarly to the one trained with the full MAE setup after fine-tuning. However, our assumption is not supported by our experimental results. Table 2 shows the fine-tuning results on the ImageNet-1k dataset. The result from 100 epochs fine-tuning is 3.2% worse than the result from the original MAE model (80.4% vs 83.6%).

Interestingly, we find that an additional 100 epochs of fine-tuning can improve the top-1 accuracy by 1.4% (80.4% vs 81.8%). This highlights that the pre-trained model does not fully converge with such a limited training budget. Despite this improvement, the final result is still inferior to the original MAE setup by 1.8% (81.8% vs 83.6%). In conclusion, increasing the fine-tuning duration can alleviate the challenges introduced by reduced pre-training to some extent. However, fine-tuning is much more expensive than pre-training, as the model needs to compute with the whole image patches, making it not ideal to increase the fine-tuning duration. As such, we investigate alternative approaches that do not incur additional computational cost.

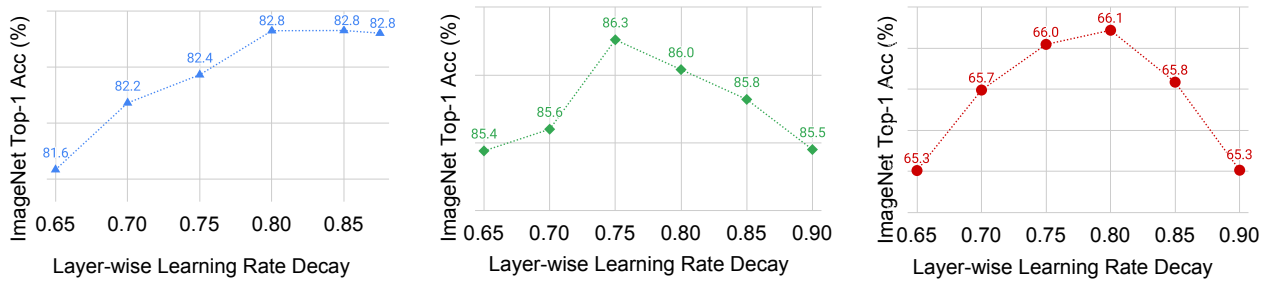
Pre-training batch size. We here investigate the impact of batch size on the performance of the MAE pre-training. Previous research has shown that a limited pre-training length combined with a large batch size can lead to the model stuck in a sharp minimum [15, 16]. To explore this phenomenon, we conduct experiments with various batch sizes, as reported in Table 3. Our results indicate that a smaller batch size, such as 512 or 1024, consistently outperforms the default batch size of 4096 by a significant margin (over 1%) in our modified pre-training setup. Therefore, we opt for a batch size of 1024 as a balance between efficiency and accuracy in our proposed pre-training recipe.

Despite these improvements, we note that our pre-training results are still suboptimal compared to those obtained using a 1600-epoch pre-training approach (81.6% vs 83.6%). To further compensate for this gap, we adopt more aggressive fine-tuning techniques, as introduced next.

4.2. Layer-wise Learning Rate Decay

In the original MAE setup, a small value for LLRD is used under the assumption that lower-level features are well-learned during pre-training and do not require aggressive updates during fine-tuning. However, this may not be the case with our reduced pre-training budget.

Therefore, we experiment using a larger LLRD value



(a) ImageNet-1k (b) ImageNet-200 (c) ImageNet-10%

Figure 3. **Optimal Layer-wise learning rate decay of different datasets.** (a) Optimal LLRD outperforms the default one 0.65 by 1.3%. (b)(c) We observe a similar trend for LLRD between target dataset ImageNet-1k and small proxy datasets like ImageNet-200 & ImageNet-10%, which enables low cost LLRD search.

during fine-tuning. This significantly increases learning rates for shallow layers while only slightly increasing the learning rates for deeper layers. Our results, as shown in Figure 3(a), illustrate that this approach significantly improved performance. For instance, when using a masking ratio of 90%, the best LLRD rate outperforms the default LLRD of 0.65 by 1.3%. These results demonstrate the crucial role of a proper layer-wise learning rate decay in the success of MAE learning under a limited computational budget.

Optimal LLRD of different pre-train recipes. To further investigate the impact of low-level feature quality on fine-tuning performance, we examined the optimal layer-wise learning rate decay (LLRD) rate for models pre-trained for varying numbers of epochs, as shown in Figure 4. The results validate that models pre-trained for shorter periods typically require larger LLRD values, indicating a need for larger and faster updates on the weights of shallow layers. Furthermore, fine-tuning performance improves more significantly with fewer pre-training epochs when using an optimized LLRD rate. For example, as shown in Figure 4, our proposed recipe outperforms the default recipe by 1.2% when the model is pre-trained for only 100 epochs, but the performance difference is only 0.1% when the model is well-pre-trained (*i.e.*, 1600 epochs + 8-layer decoder).

Low-cost parameter searching. Finding the optimal value for layer-wise learning rate decay rate is crucial to our method. Figure 3 shows a typical relation between layer-wise learning rate decay rate and final model performance: LLRD has a sweet point, *i.e.*, the fine-tuning accuracy will first increase and then decrease. To determine this optimal value, a parameter search is typically required, which can be computationally expensive, especially with a large training set. This motivates us to explore the possibility of performing parameter searches with smaller proxy datasets, such as ImageNet-200 and ImageNet-10%, to efficiently and effectively locate the optimal value of LLRD. Specifically, these datasets are subsets of ImageNet-1k, where 200 classes are randomly chosen from the 1000 classes in ImageNet-200, and 10% of the images are kept for each class in ImageNet-10%, respectively.

As demonstrated in Figure 3(b) and Figure 3(c), the relationship between layer-wise learning rate decay (LLRD) and model performance on ImageNet-200 and ImageNet-10% follows a similar trend as it does on the full ImageNet-1k dataset. For example, the optimal value for LLRD is ~0.8, which is consistent with the region identified on the full ImageNet-1k dataset. By conducting the parameter search on these smaller proxy datasets, we are able to significantly expedite the search process and reduce the computational cost by a factor of 10.

4.3. Fine-tuning Learning Rate

Lastly, we delved into the influence of different learning rates. Figure 5 illustrates the results of using different learning rates during fine-tuning. We find that increasing the learning rate brings a consistent improvement in performance when using the default LLRD of 0.65. This is expected as a larger learning rate can help feature convergence. However, when examining the combined effect of learning rate and LLRD, we discover that using excessively high learning rates results in suboptimal performance, as it leads to oscillations around the optimal performance due to overly high learning rates for deep layers. Therefore, we choose a moderate learning rate of 1e-3 as our default fine-tuning learning rate in order to achieve a balance between performance and stability.

5. Experiments

We follow the general MAE pre-training and fine-tuning setup outlined in Section 3, with some modifications to a few hyper-parameters. Specifically, for pre-training, we use a single-layer decoder instead of the eight-layer decoder, decrease the number of training epochs from 1600 to 100, increase the masking ratio from 75% to 90%, and use a batch size of 1024 instead of 4096. For fine-tuning, we in-

method	masking ratio	pre-train epochs	pre-train hours	normalized pre-train cost	fine-tune epochs	fine-tune hours	total hours	normalized total cost	top-1 acc. (%)
VIT-B	-	300	-	-	-	-	-	-	82.3
DEIT-B	-	300	-	-	-	-	-	-	81.8
BEiT-B	40%	800	-	-	100	-	-	-	83.2
MAE-B	75%	1600	202.2	59.2×	100	15.7	217.9	11.4×	83.6
Ours	75%	200	9.9	2.7×	100	15.7	25.6	1.3×	83.3
Ours	75%	100	4.9	1.4×	100	15.7	20.6	1.1×	83.1
MAE-B	90%	1600	148.9	43.6×	100	15.7	164.6	8.6×	83.1
MAE-B	90%	100	3.4	1.0×	100	15.7	19.1	1.0×	80.4
Ours	90%	100	3.4	1.0×	100	15.7	19.1	1.0×	83.0

Table 4. **Comparison with other methods on ImageNet-1k.** ViT-B is used as the model backbone. We benchmark the speed on a machine with 8 NVIDIA A5000 GPUs. The normalized cost is calculated relative to our method. Our method achieves competitive results with much less computational cost.

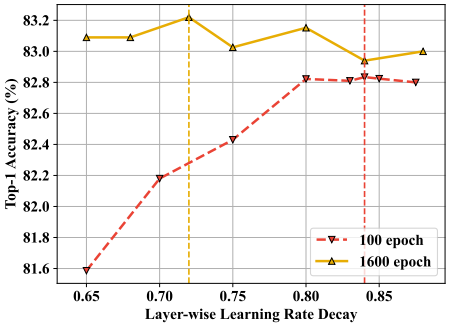


Figure 4. **Optimal LLRD of different pre-train recipes.** We empirically find that the models pre-trained for a longer time tend to have a smaller optimal LLRD.

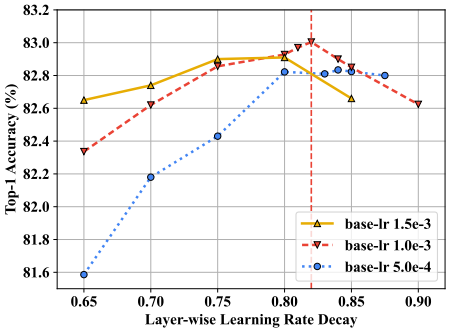


Figure 5. **Fine-tuning learning rate.** First, increasing the learning rate consistently boosts the performance using MAE's default LLRD 0.65. Second, a moderate learning rate 1e-3 is optimal when jointly searching for learning rate and LLRD.

crease the base learning rate from 5e-4 to 1e-3 and apply a layer-wise learning rate decay determined by our low-cost parameter search for each pre-training setup. Unless otherwise noted, all other hyper-parameters remain consistent with the original MAE recipe. The implementation details can be found in Table 1.

5.1. Main Results

ImageNet-1k. We compare our method to other supervised and self-pre-trained methods on ImageNet-1k in Table 4. We use ViT-B as the baseline model and report each method's training speed using an 8 NVIDIA A5000 GPU server.

We evaluate MAE under several different training schedules. In its original settings, MAE achieved 83.6% top-1 accuracy on the downstream task, with 1600 pre-training epochs and a 75% masking ratio. However, when we increased the masking ratio to 90%, we observed a decrease in performance to 83.1%. Additionally, when we further reduced the pre-training length to 100 epochs, we observed a significant decrease in performance to 80.4% top-1 accuracy.

However, when trained with our proposed recipe, without any changes to the MAE structure, the model achieves 83.0% top-1 accuracy on the downstream task, which is an improvement of 1.6% over the original MAE counterpart. Notably, our performance is comparable to the fully trained 90%-masked-trained MAE (*i.e.*, the gap is only 0.1%) and even to the 75%-mask one. It's also important to highlight that our method significantly reduces pre-training cost by over 40 times when benchmarked by time spent.

Lastly, we evaluate our methods in less aggressive training scenarios. By decreasing the masking ratio from 90% to 75%, we observed a slight improvement in top-1 accuracy by 0.1%, with an optimal LLRD of 0.85. Moreover, by increasing the number of pre-training epochs to 200, we observed a 0.2% improvement in top-1 accuracy, with an optimal LLRD of 0.775.

Image Segmentation We test our method with the original MAE on the Image Segmentation Task. We choose

method	mask ratio	pre-train epochs	fine-tune iterations	mIOU
MAE-B	75%	100	160k	42.4
MAE-Ours	75%	100	160k	**44.3**

Table 5. **Comparison with the Orginal MAE on ADE20K.** The model are samely pretrained as the ImageNet settings. In limited training budget, our method also shows a significant improvement compared to the original MAE recipe.

method	masking ratio	pre-train epochs	fine-tune epochs	top-1 acc.(%)
DEIT-S	-	300	-	77.9
MAE-S	75%	100	100	77.9
Ours-S	75%	100	100	79.8
Ours-S	90%	100	100	79.9

Table 6. **Performance on ViT-S.** We validate our methods on ViT-S, which shows a similar behaviour to our ViT-B experiments. With our recipe, we obtain a largely improved performance (79.9%) against the original one (77.9%).

LLRD	0.8	0.825	0.85	0.875
masking ratio = 75%	79.66	**79.81**	79.73	79.75
masking ratio = 90%	79.78	**79.88**	79.83	79.82

Table 7. **Influence of LLRD on ViT-S.** We demonstrate how the downstream task's performance will change with respect to LLRD of the region between 0.8 and 0.9. This trend is similar to our ViT-B experiments.

the ADE20K[31] as our benchmark dataset. By applying 100 epoch pre-train and 160K iteration fine-tune to both the original MAE recipe and Our recipe. The result shows significant improvement of our method (+1.9%) to the original MAE as shown in Table 5.

Scaling to different size Next, we evaluate our recipe in models of different scales. In Tab. 6 and Tab. 7, we validate the proposed recipe on ViT-S. Our recipe drastically decreases the computational cost for MAE training on ViT-S, while attains significant performance improvement compared with the original recipe. Specifically, as shown in Tab. 6, our approach has 2% top-1 accuracy improvement in downstream task over the original MAE fine-tune recipe. We also conduct a low cost parameter searching for layer-wise learning rate decay in Tab. 7, where we observe a similar increasing-peaking-decreasing pattern as Fig. 3

Generalizing to other MIM Method In addition to MAE, we also demonstrate the effectiveness of our approach on another MIM algorithm, SupMAE [19]. SupMAE introduces golden labels into the pre-training stage to help the model learn global information and is able to shorten the pre-training length by 4×. By applying our recipe, SupMAE can be further accelerated. With the best LLRD, the model reaches 83.0% top-1 accuracy with only 50 epochs of pre-training, reducing the total training cost by another half. We demonstrate the influence of LLRD for SupMAE in Table 8 and observe the following: (1) Proper LLRD outperforms the default setting by 0.4%, showing the effectiveness of our recipe. (2) SupMAE is relatively robust in a large region from 0.8 to 0.85, simplifying the parameter search process.

LLRD	0.65	0.775	0.8	0.825	0.85	0.90
Acc.(%)	82.6	82.9	**83.0**	**83.0**	**83.0**	80.5

Table 8. **Generalization to SupMAE.** We combine our methods with SupMAE. The experiments shows that with only 50 epoch's pre-train, we reach 83.0 top-1 accuracy by using the best LLRD. We further verify that it has a similar trend in how LLRD affects final results as we find in MAE.

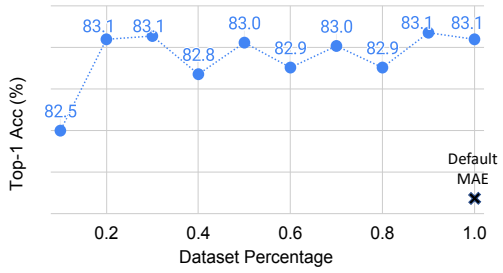
5.2. Sample Efficiency

In addition to the training efficiency of MAE, we also investigate the sample efficiency of MAE, *i.e.*, whether MAE can still achieve competitive performance even if only a small amount of images are available for pre-training. We would like to stress that this setup is practically meaningful, especially in the medical image domain where even unlabelled data are difficult to collect at scale.

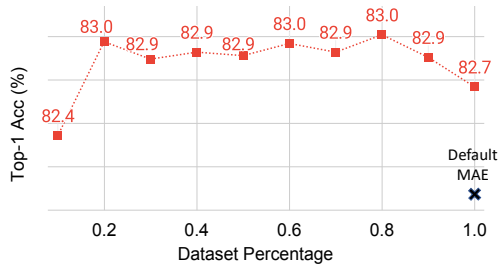
By using a 100 epoch, 75% masking ratio, and 1-layer decoder MAE as the baseline model, we pre-trained MAE on subsets of ImageNet. To ensure a fair comparison between different setups, we adjust training epochs accordingly to maintain the same computational resource as the baseline, *e.g.*, we will double the training epochs if the dataset size is halved. The results, as illustrated in Figure 6, demonstrate that using our fine-tuning recipe, MAE can robustly achieve competitive performance even when significantly fewer data during pre-training. Additionally, we observed that in certain cases, smaller subsets of the data could even result in better performance. For example, using a 50% budget, we were able to achieve a performance of 82.9% using only 20% of the data. This suggests that, with a limited pre-training budget, the performance will exhibit a tradeoff between the size of the dataset and the number of epochs.

5.3. Ablation Study

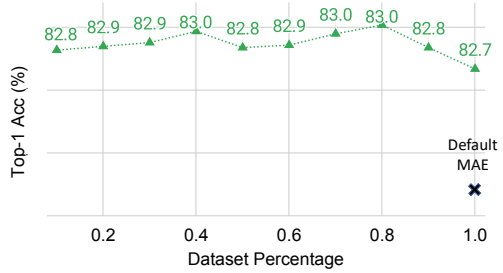
Convergence of MAE model using our recipe In Tab. 2, we show that in the naive MAE settings, improving the fine-tuning length leads to a 1.4% improvement in downstream tasks, suggesting that the model does not fully converge. Here, we also test our recipe with a longer fine-tune duration and show the results in Tab. 9. Doubling the fine-tune epochs marginally improves the accuracy by 0.1%. This result indicates that the pre-trained model can achieve good



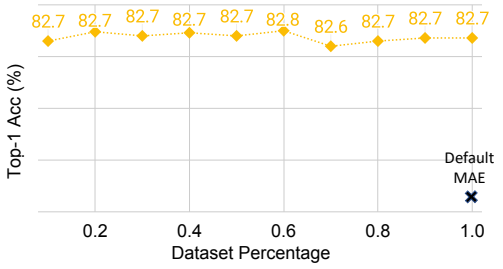
(a) 100% Computational Resource



(b) 75% Computational Resource



(c) 50% Computational Resource



(d) 25% Computational Resource

Figure 6. **The sample efficiency of MAE.** By fine-tuning MAE with our recipe, we find that for a fixed computational resource, the performance on downstream tasks remains consistent. This holds true even when we change the total computational budget. We also include how the default MAE performs under the same pre-train budget.

convergence with our recipe, thus avoiding extra fine-tune computational overhead.

Decoder depth Although previous work shows that increasing decoder depth may not help improve the model's performance when the model has already been well pre-trained [13], we wonder whether a larger decoder can assist reconstruction process and bring better downstream performance under the short pre-train schedule.

fine-tuning length	top-1 (%)
100	83.0
200	83.1

Table 9. **Ablation on the our recipe's convergence.** We study the convergence of the MAE model pre-trained with our recipe by varying the fine-tune epochs. Enlarging the fine-tune epochs brings marginal improvements, suggesting that our recipe can fully exploit the model's potential during fine-tuning without introducing extra computational burden.

decoder depth	top-1 (%)
1	82.8
8	82.7

Table 10. **Ablation on the decoder depth.** We show that the decoder depth has only a marginal influence in our recipe. This result is consistent with the findings in MAE [13].

As demonstrated in Tab. 10, adding decoder depth has little influence on the performance of our training recipe. The difference between the heavy eight-layers decoder and the lightweight one-layer decoder is minor, which is consistent with the conclusion of MAE [13]. Using the heavier eight-layers decoder even slightly deteriorates the performance by 0.1%, which may come from the fact that the heavier decoder is harder to converge with the short fine-tuning stage.

6. Conclusion

In this paper, we delve deep into the MAE pre-training and fine-tuning recipe. Through extensive experiments, we systematically showcase MAE as an efficient learner. With our proposed training recipe, we achieve a remarkable 82× speed up with little performance loss by aggressively reducing pre-training cost and tuning layer-wise learning rate decay in the fine-tuning stage. Our exploration also extends to sample efficiency, where we demonstrate the ability to achieve comparable performance using only 20% of the data. Furthermore, our proposed recipe demonstrates versatility across different model scales and MIM methods. We hope that our work can boost fast experimental prototyping and validation in this research area.

Acknowledgement

This work is partially supported by TPU Research Cloud program, Google Cloud Research Credits program, and AWS Cloud Credit for Research program.

References

[1] Hangbo Bao, Li Dong, and Furu Wei. Beit: Bert pre-training of image transformers. In *ICLR*, 2021. 1, 2, 3

[2] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021. 2

[3] 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*, pages 1691–1703. PMLR, 2020. 2

[4] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1691–1703. PMLR, 13–18 Jul 2020. 2

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, pages 1597–1607. PMLR, 2020. 2

[6] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021. 2

[7] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *ICCV*, pages 9640–9649, 2021. 2

[8] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*, 2020. 2, 3

[9] 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*, 2020. 3

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 1

[11] Christoph Feichtenhofer, Haoqi Fan, Yanghao Li, and Kaiming He. Masked autoencoders as spatiotemporal learners. *arXiv preprint arXiv:2205.09113*, 2022. 2

[12] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33:21271–21284, 2020. 2

[13] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, pages 16000–16009, 2022. 1, 2, 3, 8

[14] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. 2

[15] Elad Hoffer, Itay Hubara, and Daniel Soudry. Train longer, generalize better: closing the generalization gap in large batch training of neural networks. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, 2017. 4

[16] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *International Conference on Learning Representations*, 2017. 4

[17] Nikos Komodakis and Spyros Gidaris. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations (ICLR)*, Vancouver, Canada, 2018. 2

[18] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Learning representations for automatic colorization. In *European conference on computer vision*, pages 577–593. Springer, 2016. 2

[19] Feng Liang, Yangguang Li, and Diana Marculescu. Supmae: Supervised masked autoencoders are efficient vision learners. *arXiv preprint arXiv:2205.14540*, 2022. 2, 7

[20] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *European conference on computer vision*, pages 527–544. Springer, 2016. 2

[21] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pages 69–84. Springer, 2016. 2

[22] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016. 2

[23] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 1

[24] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021. 1

[25] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *NeurIPS*, 2022. 2

[26] Chen Wei, Haoqi Fan, Saining Xie, Chao-Yuan Wu, Alan Yuille, and Christoph Feichtenhofer. Masked feature prediction for self-supervised visual pre-training. In *CVPR*, pages 14668–14678, 2022. 2

[27] Chen Wei, Lingxi Xie, Xutong Ren, Yingda Xia, Chi Su, Jiaying Liu, Qi Tian, and Alan L. Yuille. Iterative reorganization with weak spatial constraints: Solving arbitrary jigsaw puzzles for unsupervised representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vi-*

sion and Pattern Recognition (CVPR), June 2019. 2

[28] Jiahao Xie, Wei Li, Xiaohang Zhan, Ziwei Liu, Yew Soon Ong, and Chen Change Loy. Masked frequency modeling for self-supervised visual pre-training. *arXiv preprint arXiv:2206.07706*, 2022. 2

[29] Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks. *arXiv preprint arXiv:1708.03888*, 2017. 3

[30] Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. Large batch optimization for deep learning: Training bert in 76 minutes. In *International Conference on Learning Representations*, 2020. 2

[31] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5122–5130, 2017. 7