MAE ARE SECRETLY EFFICIENT LEARNERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Masked Autoencoders (MAE), introduced by (He et al., 2022), provides a strong framework to pre-train Vision Transformers (ViTs). In this paper, we accelerate MAE training by **59**$\times$ or more while with little performance drop. Our changes are simple and straightforward: in the pre-training stage, we aggressively increase the masking ratio, decrease the training epochs, and reduce the decoder depth, for lowering pre-training cost; in the fine-tuning stage, we reveal layer-wise learning rate decay plays a vital role on unleashing the power of pre-trained models. With this setup, we are able to **pre-train a ViT-B in 12.6 hours using a single the latest NVIDIA A100 GPU**, which competitively attains 83.0% top-1 accuracy on the downstream ImageNet classification task. We additionally verify the speed acceleration on another MAE extension, SupMAE.

1 INTRODUCTION

Masked Image Modeling (MIM) (Bao et al., 2021; He et al., 2022), which trains a model to predict masked signals (either as raw pixels or semantic tokens) based on visible image regions, is a powerful self-supervised pretext task. Thanks to MIM, especially the most recent instantiation Masked Autoencoder (MAE), we are able to successfully pre-train the data-hungry Vision Transformers (ViTs) (Dosovitskiy et al., 2021) by only using middle-scale datasets like ImageNet-1k (Deng et al., 2009); such trained ViTs attain state-of-the-art performance on a variety of downstream recognition tasks and out-of-distribution tests.

Despite its strong performance, MIM pre-training typically introduces a significant amount of computations. For example, BEiT (Bao et al., 2021) requires a long schedule of 800 pre-training epochs on ImageNet-1k dataset to achieve competitive performance. The follow-up work MAE (He et al., 2022), which takes advantage of a high masking ratio (*i.e.*, 75%) and a asymmetric encoder-decoder design, substantially accelerate pre-training; but MAE still requires an excessively long schedule of 1,600 pre-training epochs. This heavy pre-training cost may potentially limits the wider explorations of MIM in the large-computation regime (*e.g.*, larger models and larger datasets), which is the key factor to release the emerging and astonishing properties of deep learning systems (Rombach et al., 2021; Ramesh et al., 2022).

In this paper, we are interested in improving the pre-training efficiency of MIM while with little-to-no performance drop. We take MAE as the study case and our motivation starts from a simple yet interesting observation: aggressively lightening MAE pretraining (*i.e.*, decreasing pre-training epochs from 1600 to 100, increasing the masking ratio from 75% to 90%, and reducing decoder depth from 8 to 1) will only moderately degrade the image reconstruction quality, as shown in Fig 1. In other words, such lightly pre-trained models are nearly equivalently capable of modeling masked inputs as those fully pre-trained models, and conjecturally, *we expect them should still be comparable when fine-tuning to different downstream tasks*. Nonetheless, interestingly, contradictory to our assumption, we observe a significant performance drop on ImageNet classification by following the official fine-tuning setup in (He et al., 2022), *i.e.*, the top-1 accuracy is decreased from 83.6% to 80.4% with ViT-B. We further note this 80.4% accuracy could be improved if we simply adapt a longer fine-tuning epoch, *e.g.*, the accuracy will go up to 81.8% (+1.4%) if doubling the fine-tuning length to 200 epochs. This phenomenon suggests that our lightly pre-trained model does not fully converge when following the original fine-tuning recipe, and invites us to ponder the question: can we find a fine-tuning recipe that allows lightly pre-trained models to converge faster, and more preferably, to achieve comparable fine-tuning performance with those fully pre-trained models?

We identify **Layer-wise Learning Rate Decay** (LLRD), which is a hyper-paramter widely applied in modern Transformer optimization to scale the learning rate of each layer, plays a vital role in effective fine-tuning. Specifically, LLRD is originally used by LAMB for enabling stable and fast BERT training (You et al., 2020; Clark et al., 2020), and then applied to the Transformers in the vision domain by BEiT (Bao et al., 2021). From top to bottom, the learning rate of each layer is multiplied by a constant factor, therefore, the learning rate of the first layer is the smallest and the learning rate of the last layer is the largest. For instance, in the official MAE fine-tuning recipe, a small scaling factor of 0.65 is adopted for ViT-B encoder; this setup makes the learning rate of the first layer is less than 1% of that of the last layer. The main motivation of adapting LLRD is that the encoder has already learned strong low-level features that can transfer well to different downstream tasks during pretraining, therefore there is no need to largely alter the weights of early layers during fine-tuning.

However, with our aggressive pre-train strategy, the pre-trained model can only learn mediocre features, including both low-level and high-level features. In this case, there is no need to lower the learning rates of early layers anymore. To this end, we proposed to simply apply a larger value of LLRD in fine-tuning, for increasing the learning rates of early layers while leaving the learning rates of deep layers intact. In this way, we can successfully train the model to converge within a short pre-training and fine-tuning length. For example, with a suitable LLRD, we are able to achieve an top-1 accuracy of 83.0% on ImageNet, with a masking ratio of 90%, a pre-training length of 100 epochs, and a decoder depth of 1, leading to a performance boost of 1.5% compared to the result of the same pre-training recipe in Fig. 4. In other words, we are able to drastically reduce the pre-training cost by $\sim$60$\times$ without significantly hurting performance. We also empirically show that our finding can generalize to another MAE-based method, SupMAE. Lastly, we demonstrate the selection of LLRD is generalizable across different downstream tasks. This property allows us to first extensively search the best LLRD on small datasets like ImageNet-200 and ImageNet-1K (which is computationally cheap); the found LLRD can then be directly applied to the larger datasets for securing strong performance.

Contributions. Our paper's contribution can be summarized as follows:

- With the help of adjusting LLRD, We successfully accelerate MAE training significantly with barely no performance drop. Specifically, with our recipe, the MAE-Base model reaches 83.0% top-1 accuracy on ImageNet, 0.6% lower than the performance of original recipe, using about $\sim$60$\times$ less training budget.
- We empirically show that our finding can generalize to other MAE extensions, such as SupMAE, which potentially helps researches in related areas on fast prototyping and experiment exploration.

2 Related Works

Hand-crafted Self-supervised Learning. Visual self-supervised learning aims at learning good feature representations with supervision signal constructed by the images or video itself. Early self-supervised learning methods have designed various pre-text tasks to get the supervision for model training. The representative works include image colorization (Larsson et al., 2016), image inpainting (Pathak et al., 2016), solving jigsaw puzzles (Wei et al., 2019; Noroozi & Favaro, 2016), predicting image rotations (Komodakis & Gidaris, 2018), and temporal information verification (Misra et al., 2016).

Contrastive Learning. Contrastive learning is a popular self-supervised learning paradigm, where the model will try to distinguish different views from the same image and other images (Chen et al., 2020c; He et al., 2020; Chen et al., 2020a; Grill et al., 2020; Chen et al., 2021). Its core idea is to train the model to pull positive sample pairs together and push negative sample pairs away (Chen & He, 2021). And the Siamese architectures is shown to be an essential reason for the success of contrastive learning (Chen & He, 2021). Recently, Vision Transformer (VIT) (Dosovitskiy et al., 2020) has been introduced to this field of contrastive learning, where the class token is used to represent the entire image (Chen et al., 2021; Caron et al., 2021).

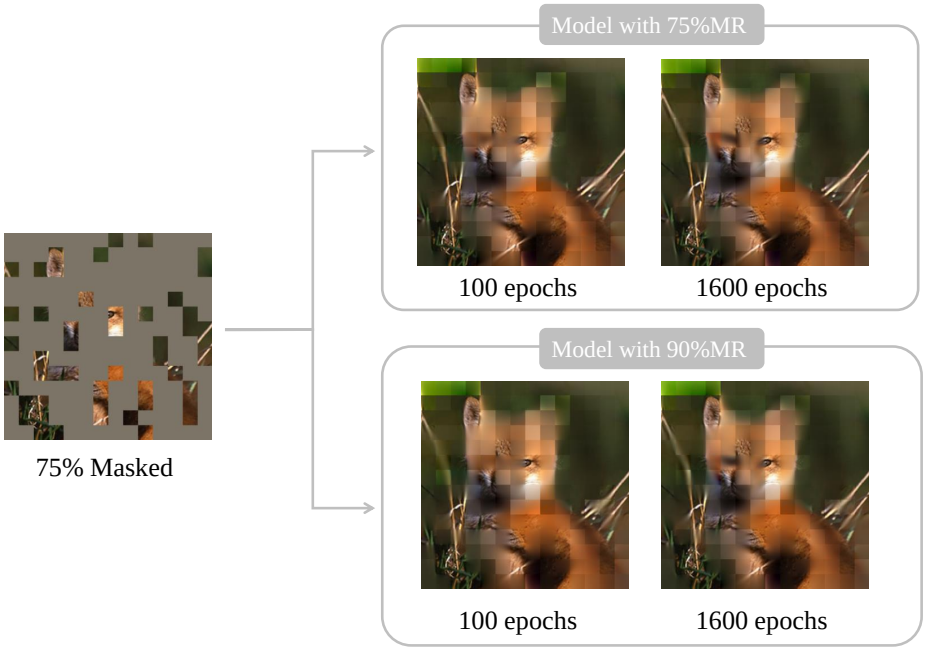


Figure 1: **Comparison of reconstructed images of MAE pre-trained for different lengths.** We show the reconstruction results of a certain masked image with models pretrained with different length and masking ratio (MR). We observe that a short pre-training length can achieve close reconstruction quality to that of a much longer pre-training length, for both 75% MR and 90% MR .

Masked Image Modeling. The masked language modeling (MLM) and its auto-regressive counterparts have achieved great success as a pre-training paradigm in Natural Language Processing (NLP). Recently, it has been introduced to the vision domain. The pioneer work IGPT (Chen et al., 2020b) learns image representations by regressing images pixel by pixel. Later BEIT (Bao et al., 2021) follows the BERT's (Devlin et al., 2018) idea of encoding image patches into semantic tokens and training the model to predict them. MaskFeats (Wei et al., 2022) explore the feasibility of predicting masked HOG (Dalal & Triggs, 2005) feature. SimMIM (Xie et al., 2022) and MAE (He et al., 2022) further demonstrate that a simple raw pixel reconstruction objective suffice to train the model to learn good representations, in which a large masking ratio (*i.e*. 75%) is shown to be the key. Moreover, MAE employs an asymmetric encoder-decoder architecture, which only takes unmasked image patches as input to the encoder, and greatly shortens the pre-training time. SupMAE (Liang et al., 2022) introduces golden label into MAE pre-training, which enables global information learning and 4x speed up compared to the original MAE. VideoMAE (Tong et al., 2022) further extends the MAE methods to video domain. In this work, we do not seek to design novel architecture or learning objective. Instead, our goal is to test the limit of MAE pre-training, and reduces the training cost as much as possible.

3 Revisiting Masked AutoEncoders

Our work is built on MAE (He et al., 2022), with the goal of cutting training cost as much as possible but still keep MAE's effectiveness. In this section, we first give a brief description about MAE, and then introduce Layer-wise Learning Rate Decay, the key to enable extremely quick MAE pre-training.

3.1 Masked AutoEncoders

MAE randomly masks some image patches, and trains the model to predict the pixel values of the masked patches based on the remaining visible patches. Its main components are as follows:

Masking Strategy. Similar to ViT (Dosovitskiy et al., 2020), MAE operates on non-overlapping image patches. A small subset of the embeded patches is sampled without replacement, and the rest

patches are masked and used as the prediction target of the decoder. In MAE (He et al., 2022), it is observed that a high masking ratio (*e.g*. 75%) is the key to prevent shortcut learning (*i.e*. extrapolating from visible neighboring patches).

MAE Encoder. MAE encoder is a vanilla ViT (Dosovitskiy et al., 2020) that only takes visible patches as input, a design that significantly reduce training time and memory cost when combined with a high masking ratio. For example, the speedup can be more than 4x for a masking ratio of 75%, due to the quadratic self-attention complexity (He et al., 2022).

MAE Decoder. MAE decoder is another vanilla ViT (Dosovitskiy et al., 2020) that operates on both visible patches and mask tokens, a shared learnable vectors that represents the masked patches. MAE decoder is only 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 is is also observed that a one-layer decoder can still produces decent results (He et al., 2022).

3.2 Layer-wise Learning Rate Decay

The layer-wise learning rate adaptive strategy is first introduced by LARS to help large batch size training of ResNet (You et al., 2017). The main idea is to multiply the learning rate of each layer with a scaling factor computed from a scaling function that takes as input the layer weights. This strategy is simplified by ELECTRA (Clark et al., 2020) and then introduced to MIM's downstream tasks by BEIT. BEIT uses a simple polynomial function to calculate the scaling factor. So in a model with h layers, given the learning rate η by the optimizer, the learning rate of the ith layer η^i is:

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

The newly introduced hyper-parameter α controls how fast the learning rate decays from the last layer to the first layer, which is why α is called **layer-wise learning rate decay weight (LLRD)**. α is usually less than 1. The default value of α is set to 0.65 for VIT-B in the fine-tuning recipe of BEIT and MAE, a value specifically designed for smaller and slower updates of low-level features learnt in the pre-training stage. In this work, we find that with an aggressively reduced pre-training cost, the learnt low-level features need to be updated in a way similar to the high-level features. And thus a suitable LLRD becomes the key to unleash the power of efficient MAE learning.

4 Method

We start from the MAE's original setup, and study how to accelerate MAE training with comparable performance.

Pilot Study Setup We follow the MAE's pre-training setup in Tab. 3a and fine-tuning setup in Tab. 3b. We use ViT-B as the backbone and report the top-1 accuracy on the ImageNet-1k dataset. We use a one-layer decoder as our default setting, as the default eight-layers decoder occupies over 50% FLOPs in the MAE model and a heavier decoder only brings marginal improvement over a light decoder in MAE (He et al., 2022). This empirically brings over 60% acceleration compared to the speed of an eight-layer decoder.

We start from 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 observation from Fig. 1, we can see that the MAE model already learns to produces decent reconstructed images in this setting. Intuitively, we assume that such a model, though pre-trained with less computational cost, would lead to just as good a model as the model pre-trained with original MAE setup after fine-tuning. However, the experiment results disagree with this assumption. We show the results of the pre-trained model fine-tuned for different lengths in Tab. 1. The result of 100 epochs fine-tuning is 3.2% worse than the result of original MAE model (80.4% vs 83.6%). The top-1 accuracy can be improved by an absolute 1.4% (80.4% vs 81.8%) with another 100 epochs fine-tuning, demonstrating that the pre-trained model does not converge with such a limited training budget. Nevertheless, the result is still worse than that the original MAE setup by 1.8% (81.8% vs 83.6%). In short, increasing the fine-tuning length could alleviate the problem brought by reduced pre-training cost to some extent. However, fine-tuning cost more than pre-training in MAE due to the asymmetrical design (more than 2× with a masking ratio of 90%), making increasing

fine-tuning length a less than ideal solution to the problem brought by reduced pre-training budget. Next, we study how to remedy this issue without bringing any additional computational cost.

Pre-train Batch Size Previous studies show that with a limited pre-training length, a large batch size will lead to the model stucking in a sharp minima (Hoffer et al., 2017; Keskar et al., 2017). Following this spirit, we study the influence of different batch sizes in Tab. 2. We can see that a smaller batch size like 512 or 1024 surpasses the default value by a clear margin (over 1%) in our modified pre-training recipe, which is consistent to Hoffer et al. (2017); Keskar et al. (2017). Thus, we choose 1024 as the default value for pre-training batch size in our proposed recipe as a trade-off of efficiency and accuracy.

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

Table 1: **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

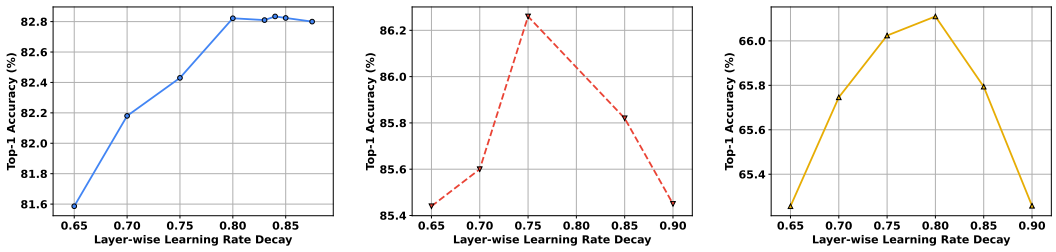
Table 2: **Pre-train the MAE models with different batch size and fine-tune with the MAE recipe.** Smaller batch size results in better convergence for shortly pre-trained models.

Even we have better convergence of the pre-trained model by changing the batch size, it's still sub-optimal compared with 1600 epochs pre-training (81.6% vs 83.6%). We adopt more aggressive fine-tuning for further compensation.

Layer-wise Learning Rate Decay (LLRD) Using a small value for LLRD in the original MAE fine-tuning recipe comes from the idea that the low-level features are well-learnt during pre-training and thus don't need aggressive updaing during fine-tuning. Yet, with our new recipe, that may not be the case because of the limited pre-training budget. Thus, we try to use a larger value for α in the fine-tuning stage. Compared with the MAE setup, the learning rates of shallow layers are increased significantly, while the learning rates of deep layers are only slightly increased. As shown in Fig. 2a, a larger LLRD rate significantly boosts the performance. For instance, the result of best LLRD rate surpasses that of the default value, 0.65, by 1.3% when the masking ratio is 90%. These results demonstrate that a proper layer-wise learning rate decay rate is essential to the success of MAE learning under limited budget.

Optimal LLRD of different pre-train recipes To verify our hypothesis that with a low-cost recipe the low-level features learnt from pre-training are not good enough from another perspective, we studies the best layer-wise learning rate decay rate for models pre-trained for different number of epochs in Fig. 3. It shows typically a model pre-trained for longer epochs need a larger LLRD, which means larger and faster updates on weights of shallow layers. Tuning layer-wise learning rate decay rate brings more gain with less pre-training epochs. For example, as shown in Fig. 3, when the model is only pre-trained for 100 epochs, our proposed recipe beats its default counterpart by 1.2%, but only by 0.1% in well pre-trained cases (*i.e*. 1600 epochs + 8 decoder).

Low-cost parameter searching Finding a good value for layer-wise learning rate decay rate is crucial to our method. Fig. 2 show a typical relation between layer-wise learning rate decay rate and final model performance. When LLRD increases, the fine-tuning accuracy will first increase and then decrease. A parameter search is needed to determine the best LLRD value, which is expensive itself. Alternatively, we show the choice of layer-wise learning rate decay is robust to datasets and can be cheaply determined on smaller proxy datasets, such as ImageNet-200 and ImageNet-10%.



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

Figure 2: **Optimal Layer-wise learning rate decay of different datasets.** (a) Optimal LLRD outperforms the default one (0.65) by 1.3%. (b)(c) Similar trend for LLRD between target dataset ImageNet-1k and small proxy datasets like ImageNet-200 & ImageNet-10% enables low cost LLRD search.

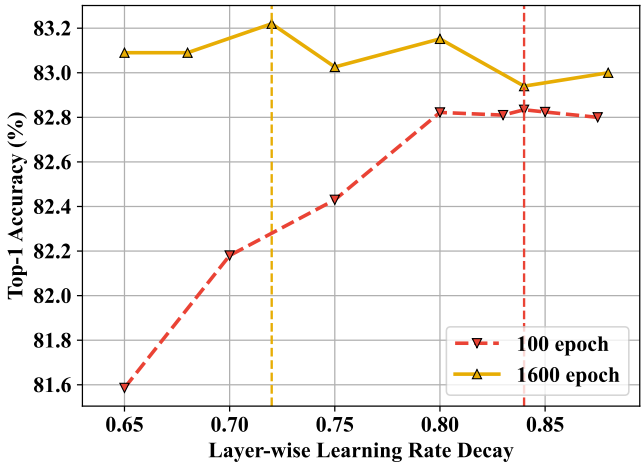


Figure 3: **Optimal LLRD of different pretrain recipes.** We empirically find that the models pre-trained for longer time tend to have smaller optimal LLRD.

Both datasets are subsampled from ImageNet-1k, where 200 classes are randomly chosen from 1000 classes in ImageNet-200, and 10% images are kept for each class in ImageNet-10% respectively.

From Fig. 2b and Fig. 2c, we can see that the relationship between LLRD and model performance on ImageNet-200 and ImageNet-10% follows a similar trend as it does on ImageNet-1k: increasing-peaking-decreasing. And we can see the best LLRD lies in around 0.8, which matches region of the value we use for the whole ImageNet-1k dataset. By proxying the parameter search on those small datasets, we are able to accelerate the whole searching procedure by 10 times.

Fine-tuning Learning Rate We have also tried a larger learning rate in fine-tuning recipe for faster convergence. As shown in Fig. 4, increasing the learning rate consistently boosts the performance using the MAE default LLRD 0.65. However, when jointly considering learning rate and LLRD, too large learning rate is sub-optimal, as it results in oscillating around the optimal due to too large learning rates for deep layers. We choose the moderate 1e-3 as our default fine-tuning learning rate to reach the balance overall.

5 Experiments

Taking the knowledge from Sec. 3 together, we follow the MAE's pre-training and fine-tuning setup, with a few hyper-parameters modified. For pre-training, we use a lightweight one-layer decoder instead of the heavy eight-layers decoder, reduce the number of epoch 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,

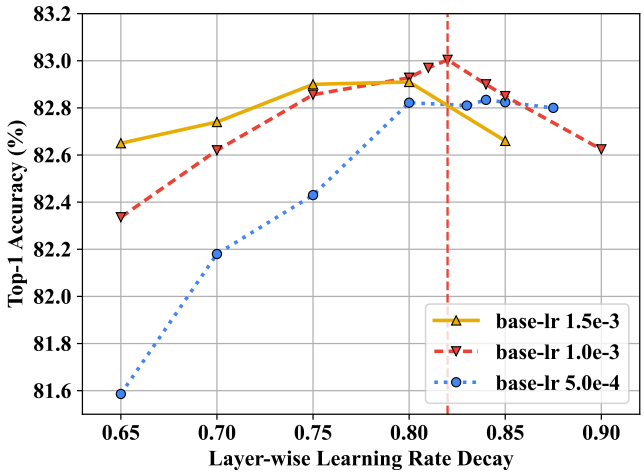


Figure 4: **Fine-tuning Learning rate.** (1) Increasing the learning rate consistently boosts the performance using the MAE default LLRD 0.65. (2) Moderate value 1e-3 is optimal when jointly considering learning rate and LLRD.

we increase the base learning rate of from 5e-4 to 1e-3, and use a layer-wise learning rate decay found by our low cost parameter searching with respect to each pre-train setup. The other hyper-parameters are exactly the same as MAE's original recipe unless specified. The implementation details can be found in Tab. 3

5.1 Main Results

Comparision on ImageNet-1k We compare our method with the other supervised or self-pre-trained methods on ImageNet-1k in Tab. 4. We compare each methods' speed on 8 NVIDIA A5000 GPUs. With our proposed recipe, we are able to cut the training cost by over 40 times, while still attain a competitive accuracy of 83.0% top-1 accuracy on the downstream task. Decreasing the masking ratio from 90% to 75% slightly improves the top-1 accuracy by 0.1%, with the optimal LLRD 0.85. Further increasing the number of pre-train epochs to 200 gives us a 0.2% top-1 accuracy improvement, where the optimal LLRD equals 0.775.

Scaling to different size We also validate the proposed recipe on ViT-S in Tab. 5. 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. 5a, our approach has 2% top-1 accuracy improvement in downstream task over the original MAE fine-tune recipe. We also conduct the low cost parameter searching for layer-wise learning rate decay in Tab. 5b, where we observe a similar increasing-peaking-decreasing pattern as Fig. 2

Generalizing to other MIM Method Besides MAE, We also validate the effectiveness of our approach on another MIM algorithm. SupMAE (Liang et al., 2022) introduces golden labels into the pre-training stage and helps the model learn global information. It is able to shorten the pre-training length by 4×. With the help of 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, cutting down the total training cost by another half. We study the influence of LLRD for SupMAE in Tab. 6 with the following observation: (1) Proper LLRD outperforms the default setting by 0.4%, showing the effectiveness of our recipe. (2) SupMAE is fairly robust in a relatively large region from 0.8 to 0.85, which further simplifies the parameter searching.

5.2 Ablation Study

Convergence of MAE model using our recipe In Tab. 7a, a pre-trained MAE model using our recipe is fine-tuned with different length. Doubling the fine-tune epochs marginally improve the accuracy by 0.1%, indicating the pre-trained model has good convergence with our recipe without introducing extra fine-tune computational burden.

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

(a) Pre-training setting.

config	MAE	ours
optimizer	AdamW	AdamW
base learning rate	5e-4	1e-3
weight decay	0.05	0.05
optimizer momentum	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
learning rate schedule	cosine decay	cosine decay
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 setting.

Table 3: **Hyper-parameter comparison.** The difference of between the MAE's default setting and our recipe's default setting are bolded. Our recipe use a LLRD found by our low cost parameter searching with respect to each pre-train setup (pre-train epochs / masking ratio).

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
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 are used as the model backbone. We benchmark the speed on a machine with 8 NVIDIA A5000 GPUs. The normalized cost are calculated relative to our method. Our method achieve competitive results with much less computational cost.

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

(a) Performance on VIT-S.

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

(b) Influence of LLRD on VIT-S.

Table 5: **Results on VIT-S.** We validate our methods on ViT-S, which shows a similar behaviour to our VIT-Base experiments. With our recipe, we can gain dramatically improved performance against the original one. Here we demonstrate how the downstream task's performance will change corresponding to LLRD of the region between 0.8 and 0.9.

LLRD	0.65	0.775	0.8	0.825	0.85	0.90
top-1 acc.(%)	82.6	82.9	**83.0**	**83.0**	**83.0**	80.5

Table 6: **Generalization to Sup-MAE.** We combine our methods with the SupMAE. The experiments shows that with only 50 epoch's pretrain, we can reach 83.0 top 1 accuracy by using a using the best LLRD. We further verify that it has a similar trend in how LLRD affect final results as we have tested in pure MAE architecture.

Decoder depth Tab. 7b demonstrates how adding decoder depth can influence our recipe's performance. We use 0.8 as the LLRD for both cases, since it works well in most situations. The difference between the heavy eight-layers decoder and the lightweight one-layer decoder is minor, which is consistent with the conclusion of MAE (He et al., 2022). Using the heavier eight-layers decoder even slightly deteriorates the performance by 0.1%, which may come from the fact the heavier decoder is harder to converge.

finetuning length	top-1 (%)
100	83.0
200	83.1

(a) Convergence of our pre-trained model.

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

(b) Decoder depth.

Table 7: **Ablation Study.** (a) We study the convergence of the MAE model pre-trained with our recipe by varing the fine-tune epochs. (b) We show the depth of the decoder have marginal influence in our recipe, which is consistent with He et al. (2022).

6 Conclusion

In this paper, we take a closer look at MAE training recipe. Through extensive experiments, we demonstrate that we can achieve about 60× speed up with little performance loss by aggressively reducing re-training cost and tuning layer-wise learning rate decay in the fine-tuning stage. Our proposed recipe works with different model scales, and MIM methods. We hope that our work can boost fast experimental prototyping and validation in this research area.

References

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

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*, pp. 9650–9660, 2021.

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*, pp. 1691–1703. PMLR, 2020a.

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 (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1691–1703. PMLR, 13–18 Jul 2020b.

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

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

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

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.

Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pp. 248–255, 2009.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

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.

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. URL https://openreview.net/forum?id=YicbFdNTTy.

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.

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*, pp. 9729–9738, 2020.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, pp. 16000–16009, 2022.

Elad Hoffer, Itay Hubara, and Daniel Soudry. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, 2017.

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. URL https://openreview.net/forum?id=H1oyRlYgg.

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

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

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

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

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

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*, pp. 2536–2544, 2016.

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.

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

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

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 Vision and Pattern Recognition (CVPR)*, June 2019.

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*, pp. 14668–14678, 2022.

Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. In *CVPR*, 2022.

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

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.