



MetaLR: Meta-tuning of Learning Rates for Transfer Learning in Medical Imaging

Yixiong Chen^{1,2}, Li Liu^{3(✉)}, Jingxian Li⁴, Hua Jiang^{1,2}, Chris Ding¹,
and Zongwei Zhou⁵

¹ The Chinese University of Hong Kong, Shenzhen, China

² Shenzhen Research Institute of Big Data, Shenzhen, China

³ The Hong Kong University of Science and Technology (Guangzhou),
Guangzhou, China

avrillliu@hkust-gz.edu.cn

⁴ Software School, Fudan University, Shanghai, China

⁵ Johns Hopkins University, Baltimore, USA

Abstract. In medical image analysis, transfer learning is a powerful method for deep neural networks (DNNs) to generalize on limited medical data. Prior efforts have focused on developing pre-training algorithms on domains such as lung ultrasound, chest X-ray, and liver CT to bridge domain gaps. However, we find that model fine-tuning also plays a crucial role in adapting medical knowledge to target tasks. The common fine-tuning method is manually picking transferable layers (*e.g.*, the last few layers) to update, which is labor-expensive. In this work, we propose a meta-learning-based learning rate (LR) tuner, named MetaLR, to make different layers automatically co-adapt to downstream tasks based on their transferabilities across domains. MetaLR learns LRs for different layers in an online manner, preventing highly transferable layers from forgetting their medical representation abilities and driving less transferable layers to adapt actively to new domains. Extensive experiments on various medical applications show that MetaLR outperforms previous state-of-the-art (SOTA) fine-tuning strategies. [Codes](#) are released.

Keywords: Medical image analysis · Meta-learning · Transfer learning

1 Introduction

Transfer learning has become a standard practice in medical image analysis as collecting and annotating data in clinical scenarios can be costly. The pre-trained parameters endow better generalization to DNNs than the models trained from scratch [8, 23]. A popular approach to enhancing model transferability is by pre-training on domains similar to the targets [9, 21, 27–29]. However, utilizing specialized pre-training for all medical applications becomes impractical due to the

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-43907-0_67.

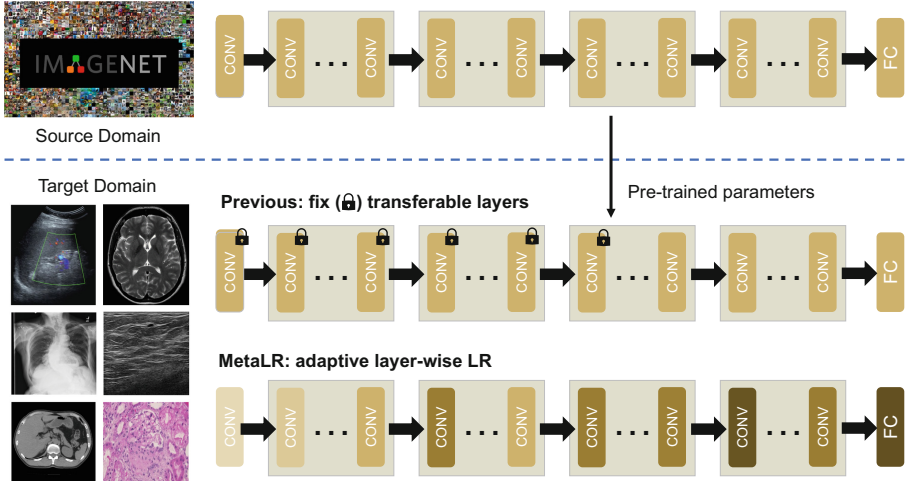


Fig. 1. The motivation of MetaLR. Previous works fix transferable layers in pre-trained models to prevent them from catastrophic forgetting. It is inflexible and labor-expensive for this method to find the optimal scheme. MetaLR uses meta-learning to automatically optimize layer-wise LR for fine-tuning.

diversity between domains and tasks and privacy concerns related to pre-training data. Consequently, recent work [2, 6, 14, 22] has focused on improving the generalization capabilities of existing pre-trained DNN backbones through fine-tuning techniques.

Previous studies have shown that the transferability of lower layers is often higher than higher layers that are near the model output [26]. Layer-wise fine-tuning [23], was thus introduced to preserve the transferable low-level knowledge by fixing lower layers. But recent studies [7] revealed that the lower layers may also be sensitive to small domains like medical images. Given the two issues, transferability for medical tasks becomes more complicated [24, 25]. It can even be irregular among layers for medical domains far from pre-training data [7]. *Given the diverse medical domains and model architectures, there is currently no universal guideline to follow to determine whether a particular layer should be retrained for a given target domain.*

To search for optimal layer combinations for fine-tuning, manually selecting transferable layers [2, 23] can be a solution, but it requires a significant amount of human labor and computational cost. In order to address this issue and improve the flexibility of fine-tuning strategies, we propose controlling the fine-tuning process with layer-wise learning rates (LRs), rather than simply manually fixing or updating the layers (see Fig. 1). Our proposed algorithm, Meta Learning Rate (MetaLR), is based on meta-learning [13] and adaptively adjusts LR for each layer according to transfer feedback. It treats the layer-wise LR as meta-knowledge and optimizes them to improve the model generalization. Larger LR indicates less transferability of corresponding layers and require more updating,

while smaller LR preserve transferable knowledge in the layers. Inspired by [20], we use an online adaptation strategy of LR with a time complexity of $O(n)$, instead of the computationally-expensive bi-level $O(n^2)$ meta-learning. We also enhance the algorithm’s performance and stability with a proportional hyper-LR (LR for LR) and a validation scheme on training data batches.

In summary, this work makes the following three contributions. 1) We introduce MetaLR, a meta-learning-based LR tuner that can adaptively adjust layer-wise LR based on transfer learning feedback from various medical domains. 2) We enhance MetaLR with a proportional hyper-LR and a validation scheme using batched training data to improve the algorithm’s stability and efficacy. 3) Extensive experiments on both lesion detection and tumor segmentation tasks were conducted to demonstrate the superior efficiency and performance of MetaLR compared to current SOTA medical fine-tuning techniques.

2 Method

This section provides a detailed description of the proposed MetaLR. It is a meta-learning-based [13, 18] approach that determines the appropriate LR for each layer based on its transfer feedback. It is important to note that fixing transferable layers is a special case of this method, where fixed layers always have zero LR. First, we present the theoretical formulation of MetaLR. Next, we discuss online adaptation for efficiently determining optimal LR. Finally, we demonstrate the use of a proportional hyper-LR and a validation scheme with batched training data to enhance performance.

2.1 Formulation of Meta Learning Rate

Let (x, y) denote a sample-label pair, and $\{(x_i, y_i) \mid i = 1, \dots, N\}$ be the training data. The validation dataset $\{(x_i^v, y_i^v) \mid i = 1, \dots, M\}$ is assumed to be independent and identically distributed as the training dataset. Let $\hat{y} = \Phi(x, \theta)$ be the prediction for sample x from deep model Φ with parameters θ . In standard training of DNNs, the aim is to minimize the expected risk for the training set: $\frac{1}{N} \sum_{i=1}^N L(\hat{y}_i, y_i)$ with fixed training hyper-parameters, where $L(\hat{y}, y)$ is the loss function for the current task. The model generalization can be evaluated by the validation loss $\frac{1}{M} \sum_{i=1}^M L(\hat{y}_i^v, y_i^v)$. Based on the generalization, one can tune the hyper-parameters of the training process to improve the model. The key idea of MetaLR is considering the layer-wise LR as self-adaptive hyper-parameters during the training and automatically adjusting them to achieve better model generalization. We denote the LR and model parameters for the layer j at the iteration t as α_j^t and θ_j^t . The LR scheduling scheme $\alpha = \{\alpha_j^t \mid j = 1, \dots, d; t = 1, \dots, T\}$ is what MetaLR wants to learn, affecting which local optimal $\theta^*(\alpha)$ the model parameters $\theta^t = \{\theta_j^t \mid j = 1, \dots, d\}$ will converge to. The optimal parameters $\theta^*(\alpha)$ are given by optimization on the training data. At the same time, the best LR tuning scheme α^* can be optimized based on the feedback for $\theta^*(\alpha)$ from

Algorithm 1. Online Meta Learning Rate Algorithm**Input:**

Training data \mathcal{D} , validation data \mathcal{D}^v , initial model parameter $\{\theta_1^0, \dots, \theta_d^0\}$, LR s $\{\alpha_1^0, \dots, \alpha_d^0\}$, batch size n , max iteration T ;

Output:

Final model parameter $\theta^T = \{\theta_1^T, \dots, \theta_d^T\}$;

```

1: for  $t = 0 : T - 1$  do
2:    $\{(x_i, y_i) \mid i = 1, \dots, n\} \leftarrow \text{TrainDataLoader}(\mathcal{D}, n)$  ;
3:    $\{(x_i^v, y_i^v) \mid i = 1, \dots, n\} \leftarrow \text{ValidDataLoader}(\mathcal{D}^v, n)$  ;
4:   Step forward for one step to get  $\{\hat{\theta}_1^t(\alpha_1^t), \dots, \hat{\theta}_d^t(\alpha_d^t)\}$  with Eq. (2);
5:   Update  $\{\alpha_1^t, \dots, \alpha_d^t\}$  to become  $\{\alpha_1^{t+1}, \dots, \alpha_d^{t+1}\}$  with Eq. (3);
6:   Update  $\{\theta_1^t, \dots, \theta_d^t\}$  to become  $\{\theta_1^{t+1}, \dots, \theta_d^{t+1}\}$  with Eq. (4);
7: end for

```

the validation loss. This problem can be formulated as the following bi-level optimization problem:

$$\begin{aligned}
 \min_{\alpha} \quad & \frac{1}{M} \sum_{i=1}^M L(\Phi(x_i^v, \theta^*(\alpha)), y_i^v), \\
 \text{s.t. } \quad & \theta^*(\alpha) = \arg \min_{\theta} \frac{1}{N} \sum_{i=1}^N L(\Phi(x_i, \theta), y_i).
 \end{aligned} \tag{1}$$

MetaLR aims to use the validation set to optimize α through an automatic process rather than a manual one. The optimal scheme α^* can be found by a nested optimization [13], but it is too computationally expensive in practice. A faster and more lightweight method is needed to make it practical.

2.2 Online Learning Rate Adaptation

Inspired by the online approximation [20], we propose efficiently adapting the LR s and model parameters online. The motivation of the online LR adaptation is updating the model parameters θ^t and LR s $\{\alpha_j^t \mid j = 1, 2, \dots, d\}$ within the same loop. We first inspect the descent direction of parameters θ_j^t on the training loss landscape and adjust the α_j^t based on the transfer feedback. Positive feedback (lower validation loss) means the LR s are encouraged to increase.

We adopt Stochastic Gradient Descent (SGD) as the optimizer to conduct the meta-learning. The whole training process is summarized in Algorithm 1. At the iteration t of training, a training data batch $\{(x_i, y_i) \mid i = 1, \dots, n\}$ and a validation data batch $\{(x_i^v, y_i^v) \mid i = 1, \dots, n\}$ are sampled, where n is the size of the batches. First, the parameters of each layer are updated once with the current LR according to the descent direction on training batch.

$$\hat{\theta}_j^t(\alpha_j^t) = \theta_j^t - \alpha_j^t \nabla_{\theta_j} \left(\frac{1}{n} \sum_{i=1}^n L(\Phi(x_i, \theta_j^t), y_i) \right), \quad j = 1, \dots, d. \tag{2}$$

This step of updating aims to get feedback for LR of each layer. After taking derivative of the validation loss *w.r.t.* α_j^t , we can utilize the gradient to know how the LR for each layer should be adjusted. So the second step of MetaLR is to move the LR along the meta-objective gradient on the validation data:

$$\alpha_j^{t+1} = \alpha_j^t - \eta \nabla_{\alpha_j} \left(\frac{1}{n} \sum_{i=1}^n L(\Phi(x_i^v, \hat{\theta}_j^t(\alpha_j^t)), y_i^v) \right), \quad (3)$$

where η is the hyper-LR. Finally, the updated LR can be employed to optimize the model parameters through gradient descent truly.

$$\theta_j^{t+1} = \theta_j^t - \alpha_j^{t+1} \nabla_{\theta_j} \left(\frac{1}{n} \sum_{i=1}^n L(\Phi(x_i, \theta_j^t), y_i) \right). \quad (4)$$

For practical use, we constrain the LR for each layer to be $\alpha_j^t \in [10^{-6}, 10^{-2}]$. Online MetaLR optimizes the layer-wise LR as well as the training objective on a single task, which differentiates it from traditional meta-learning algorithms [12, 19] that train models on multiple small tasks.

2.3 Proportional Hyper Learning Rate

In practice, LR is often tuned in an exponential style (*e.g.*, 1e-3, 3e-3, 1e-2) and are always positive values. However, if a constant hyper-LR is used, it will linearly update its corresponding LR regardless of numerical constraints. This can lead to fluctuations in the LR or even the risk of the LR becoming smaller than 0 and being truncated. To address this issue, we propose using a proportional hyper-LR $\eta = \beta \times \alpha_j^t$, where β is a pre-defined hyper-parameter. This allows us to rewrite Eq. (3) as:

$$\alpha_j^{t+1} = \alpha_j^t (1 - \beta \nabla_{\alpha_j} \left(\frac{1}{n} \sum_{i=1}^n L(\Phi(x_i^v, \hat{\theta}_j^t(\alpha_j^t)), y_i^v) \right)). \quad (5)$$

The exponential update of α_j^t guarantees its numerical stability.

2.4 Generalizability Validation on Training Data Batch

One limitation of MetaLR is that the LR is updated using separate validation data, which reduces the amount of data available for the training process. This can be particularly problematic for medical transfer learning, where the amount of downstream data has already been limited. In Eq. 2 and Eq. 3, the update of model parameter θ_j^t and LR α_j^t is performed using different datasets to ensure that the updated θ_j^t can be evaluated for generalization without being influenced by the seen data. As an alternative, but weaker, approach, we explore using **another batch of training data** for Eq. 3 to evaluate generalization. Since this batch was not used in the update of Eq. 2, it may still perform well for validation in meta-learning. The effect of this approach is verified in Sect. 3.2, and the differences between the two methods are analyzed in Sect. 3.4.

3 Experiments and Analysis

3.1 Experimental Settings

We extensively evaluate MetaLR on four transfer learning tasks (as shown in Table 1). To ensure the reproducibility of the results, all pre-trained models (USCL [9], ImageNet [11], C2L [28], Models Genesis [29]) and target datasets (POCUS [5], BUSI [1], Chest X-ray [17], LiTS [4]) are publicly available. In our work, we consider models pre-trained on both natural and medical image datasets, with three target modalities and three target organs, which makes our experimental results more credible. For the lesion detection tasks, we used ResNet-18 [15] with the Adam optimizer. The initial learning rate (LR) and hyper-LR coefficient β are set to 10^{-3} and 0.1, respectively. In addition, we use 25% of the training set as the validation set for meta-learning. For the segmentation task, we use 3D U-Net [10] with the SGD optimizer. The initial LR and hyper-LR coefficient β are set to 10^{-2} and 3×10^{-3} , respectively. The validation set for the LiTS segmentation dataset comprises 23 samples from the training set of size 111. All experiments are implemented using PyTorch 1.10 on an Nvidia RTX A6000 GPU. We report the mean values and standard deviations for each experiment with five different random seeds. For more detailed information on the models and hyper-parameters, please refer to our supplementary material.

Table 1. Pre-training data, algorithms, and target tasks.

Source	Pre-train Method	Target	Organ	Modality	Task	Size
US-4 [9]	USCL [9]	POCUS [5]	Lung	US	COVID-19 detection	2116 images
ImageNet [11]	supervised	BUSI [1]	Breast	US	Tumor detection	780 images
MIMIC-CXR [16]	C2L [28]	Chest X-ray [17]	Lung	X-ray	Pneumonia detection	5856 images
LIDC-IDRI [3]	Models Genesis [29]	LiTS [4]	Liver	CT	Liver segmentation	131 volumes

3.2 Ablation Study

In order to evaluate the effectiveness of our proposed method, we conduct an ablation study *w.r.t.* the basic MetaLR algorithm, the proportional hyper-LR, and batched-training-data validation (as shown in Table 2). When applying only the basic MetaLR, we observe only marginal performance improvements for the four downstream tasks. We conjecture that this is due to two reasons: Firstly, the constant hyper-LR makes the training procedures less stable than direct training, which is evident from the larger standard deviation of performance. Secondly, part of the training data are split for validation, which can be detrimental to the performance. After applying the proportional hyper-LR, significant improvements are in both the performance and its stability. Moreover, although the generalization validation on the training data batch may introduce bias, providing sufficient training data ultimately benefits the performance.

Table 2. Ablation study for MetaLR, hyper-LR, and validation data. The baseline is the direct tuning of all layers with constant LR. The default setting for MetaLR is a constant hyper-LR of 10^{-3} and a separate validation set.

MetaLR	Prop. hyper-LR	Val. on trainset	POCUS	BUSI	Chest X-ray	LiTS
			91.6 ± 0.8	84.4 ± 0.7	94.8 ± 0.3	93.1 ± 0.4
✓			91.9 ± 0.6	84.9 ± 1.3	95.0 ± 0.4	93.2 ± 0.8
✓	✓		93.6 ± 0.4	85.2 ± 0.8	95.3 ± 0.2	93.3 ± 0.6
✓		✓	93.0 ± 0.3	86.3 ± 0.7	95.5 ± 0.2	93.9 ± 0.5
✓	✓	✓	93.9 ± 0.4	86.7 ± 0.7	95.8 ± 0.3	94.2 ± 0.5

• Final MetaLR outperforms baseline with p-values of 0.0014, 0.0016, 0.0013, 0.0054.

Table 3. Comparative experiments on lesion detection. We report sensitivities (%) of the abnormalities, overall accuracy (%), and training time on each task.

Method	POCUS				BUSI				Chest X-ray		
	COVID	Pneu.	Acc	Time	Benign	Malignant	Acc	Time	Pneu.	Acc	Time
Last Layer	77.9 ± 2.1	84.0 ± 1.3	84.1 ± 0.2	15.8 m	83.5 ± 0.4	47.6 ± 4.4	66.8 ± 0.5	4.4 m	99.7 ± 1.3	87.8 ± 0.6	12.7 m
All Layers	85.8 ± 1.7	90.0 ± 1.9	91.6 ± 0.8	16.0 m	90.4 ± 1.5	77.8 ± 3.5	84.4 ± 0.7	4.3 m	98.8 ± 0.2	94.8 ± 0.3	12.9 m
Layer-wise	87.5 ± 1.0	92.3 ± 1.3	92.1 ± 0.3	2.4 h	90.8 ± 1.2	75.7 ± 2.6	85.6 ± 0.4	39.0 m	97.9 ± 0.3	95.2 ± 0.2	1.9 h
Bi-direct	90.1 ± 1.2	92.5 ± 1.5	93.6 ± 0.2	12.0 h	92.2 ± 1.0	77.1 ± 3.5	86.5 ± 0.5	3.2 h	98.4 ± 0.3	95.4 ± 0.1	9.7 h
AutoLR	89.8 ± 1.6	89.7 ± 1.5	90.4 ± 0.8	17.5 m	90.4 ± 1.8	76.2 ± 3.2	84.9 ± 0.8	4.9 m	95.4 ± 0.5	93.0 ± 0.8	13.3 m
MetaLR	94.8 ± 1.2	93.1 ± 1.5	93.9 ± 0.4	24.8 m	92.2 ± 0.7	75.6 ± 3.6	86.7 ± 0.7	6.0 m	97.4 ± 0.4	95.8 ± 0.3	26.3 m

3.3 Comparative Experiments

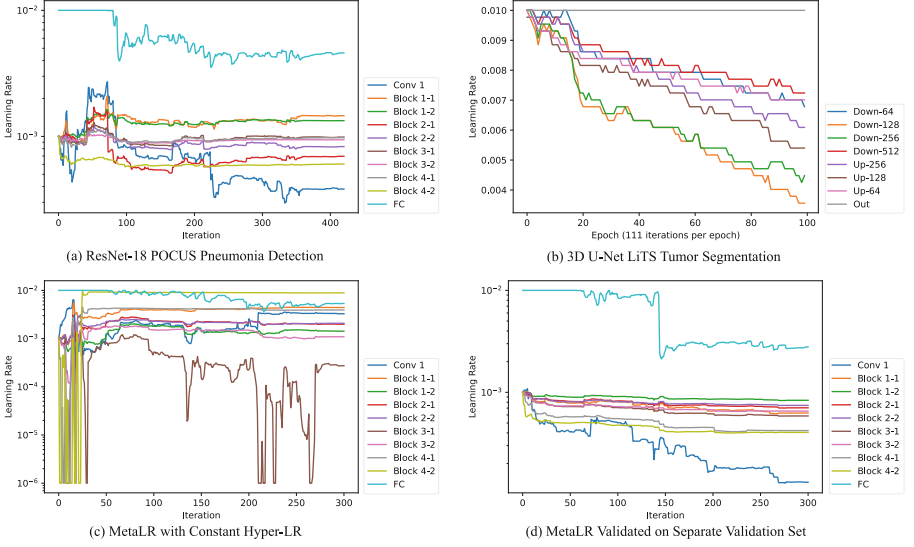
In our study, we compare MetaLR with several other fine-tuning schemes, including tuning only the last layer / all layers with constant LR, layer-wise fine-tuning [23], bi-directional fine-tuning [7], and AutoLR [22]. The U-Net fine-tuning scheme proposed by Amiri *et al.* [2] was also evaluated.

Results on Lesion Detection Tasks. MetaLR consistently shows the best performance on all downstream tasks (Table 3). It shows 1%–2.3% accuracy improvements compared to direct training (*i.e.*, tuning all layers) because it takes into account the different transferabilities of different layers. While manual picking methods, such as layer-wise and bi-directional fine-tuning, also achieve higher performance, they require much more training time ($5\times$ – $50\times$) for searching the best tuning scheme. On the other hand, AutoLR is efficient, but its strong hypothesis harms its performance sometimes. In contrast, MetaLR makes no hypothesis about transferability and learns appropriate layer-wise LR on different domains. Moreover, its performance improvements are gained with only $1.5\times$ – $2.5\times$ training time compared with direct training.

Results on Segmentation Task. MetaLR achieves the best Dice performance on the LiTS segmentation task (Table 4). Unlike ResNet for lesion detection, the U-Net family has a more complex network topology. With skip connections, there are two interpretations [2] of depths for layers: 1) the left-most layers are the shallowest, and 2) the top layers of the “U” are the shallowest. This makes the handpicking methods even more computationally expensive. However, MetaLR

Table 4. Comparative experiments on LiTS liver segmentation task.

Method	PPV	Sensitivity	Dice	Time
Last Layer	26.1 ± 5.5	71.5 ± 4.2	33.5 ± 3.4	2.5 h
All Layers	94.0 ± 0.6	93.1 ± 0.7	93.1 ± 0.4	2.6 h
Layer-wise	92.1 ± 1.3	96.4 ± 0.4	93.7 ± 0.3	41.6 h
Bi-direc	92.4 ± 1.1	96.1 ± 0.2	93.8 ± 0.1	171.2 h
Mina <i>et al.</i>	92.7 ± 1.2	93.2 ± 0.5	92.4 ± 0.5	2.6 h
MetaLR	94.4 ± 0.9	93.6 ± 0.4	94.2 ± 0.5	5.8 h

**Fig. 2.** The LR curves for MetaLR on POCUS detection (a), on LiTS segmentation (b), with constant hyper-LR (c), and with a separate validation set (d).

updates the LR for each layer according to their validation gradients, and its training efficiency is not affected by the complex model architecture.

3.4 Discussion and Findings

The LR's Learned with MetaLR. For ResNet-18 (Fig. 2 (a)), the layer-wise LR's fluctuate drastically during the first 100 iterations. However, after iteration 100, all layers except the first layer “Conv1” become stable at different levels. The first layer has a decreasing LR (from 2.8×10^{-3} to 3×10^{-4}) throughout the process, reflecting its higher transferability. For 3D U-Net (Fig. 2 (b)), the middle layers of the encoder “Down-128” and “Down-256” are the most transferable and have the lowest LR's, which is difficult for previous fine-tuning schemes to

discover. As expected, the randomly initialized “FC” and “Out” layers have the largest LR_s since they are not transferable.

The Effectiveness of Proportional Hyper-LR and Training Batches Validation. We illustrate the LR curves with a constant hyper-LR instead of a proportional one. The LR curves of “Block 3-1” and “Block 4-2” become much more fluctuated (Fig. 2 (c)). This instability may be the key reason for the instability of performance when using a constant hyper-LR. Furthermore, we surprisingly find that the learned LR_s are similar to the curves learned when validated on the training set when using a separate validation set Fig. 2 (d)). With similar learned LR curves and more training data, it is reasonable that batched training set validation can be an effective alternative to the basic MetaLR.

Limitations of MetaLR. Although MetaLR improves fine-tuning for medical image analysis, it has several limitations. First, the gradient descent of Eq. (3) takes more memory than the usual fine-tuning strategy, it may restrict the batch size available during training. Second, MetaLR sometimes does not get converged LR_s after the parameters converge, which may harm its performance in some cases. Third, MetaLR is designed for medical fine-tuning instead of general cases, what problem it may encounter in other scenarios remains unknown.

4 Conclusion

In this work, we proposed a new fine-tuning scheme, MetaLR, for medical transfer learning. It achieves significantly superior performance to the previous SOTA fine-tuning algorithms. MetaLR alternatively optimizes model parameters and layer-wise LR_s in an online meta-learning fashion with a proportional hyper-LR. It learns to assign lower LR_s for the layers with higher transferability and higher LR_s for the less transferable layers. The proposed algorithm is easy to implement and shows the potential to replace manual layer-wise fine-tuning schemes. Future works include adapting MetaLR to a wider variety of clinical tasks.

Acknowledgement. This work was supported by the National Natural Science Foundation of China (No. 62101351) and the GuangDong Basic and Applied Basic Research Foundation (No.2020A151110376).

References

1. Al-Dhabyani, W., Gomaa, M., Khaled, H., Fahmy, A.: Dataset of breast ultrasound images. *Data Brief* **28**, 104863 (2020)
2. Amiri, M., Brooks, R., Rivaz, H.: Fine-tuning u-net for ultrasound image segmentation: different layers, different outcomes. *IEEE Trans. Ultrason. Ferroelectr. Freq. Control* **67**(12), 2510–2518 (2020)
3. Armato, S.G., III, McLennan, G., Bidaut, L., et al.: The lung image database consortium (LIDC) and image database resource initiative (IDRI): a completed reference database of lung nodules on CT scans. *Med. Phys.* **38**(2), 915–931 (2011)

4. Bilic, P., et al.: The liver tumor segmentation benchmark (LITS). arXiv preprint [arXiv:1901.04056](https://arxiv.org/abs/1901.04056) (2019)
5. Born, J., Wiedemann, N., Cossio, M., et al.: Accelerating detection of lung pathologies with explainable ultrasound image analysis. *Appl. Sci.* **11**(2), 672 (2021)
6. Chambon, P., Cook, T.S., Langlotz, C.P.: Improved fine-tuning of in-domain transformer model for inferring COVID-19 presence in multi-institutional radiology reports. *J. Digit. Imaging*, 1–14 (2022)
7. Chen, Y., Li, J., Ding, C., Liu, L.: Rethinking two consensuses of the transferability in deep learning. arXiv preprint [arXiv:2212.00399](https://arxiv.org/abs/2212.00399) (2022)
8. Chen, Y., Zhang, C., Ding, C.H., Liu, L.: Generating and weighting semantically consistent sample pairs for ultrasound contrastive learning. *IEEE TMI* (2022)
9. Chen, Y., et al.: USCL: pretraining deep ultrasound image diagnosis model through video contrastive representation learning. In: de Bruijne, M., et al. (eds.) *MICCAI 2021. LNCS*, vol. 12908, pp. 627–637. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-87237-3_60
10. Çiçek, Ö., Abdulkadir, A., Lienkamp, S.S., Brox, T., Ronneberger, O.: 3D U-net: learning dense volumetric segmentation from sparse annotation. In: Ourselin, S., Joskowicz, L., Sabuncu, M.R., Unal, G., Wells, W. (eds.) *MICCAI 2016. LNCS*, vol. 9901, pp. 424–432. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46723-8_49
11. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: a large-scale hierarchical image database. In: *CVPR*, pp. 248–255 (2009)
12. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: *ICML*, pp. 1126–1135. PMLR (2017)
13. Franceschi, L., Frasconi, P., Salzo, S., Grazzi, R., Pontil, M.: Bilevel programming for hyperparameter optimization and meta-learning. In: *ICML*, pp. 1568–1577. PMLR (2018)
14. Guo, Y., Shi, H., Kumar, A., Grauman, K., Rosing, T., Feris, R.: Spottune: transfer learning through adaptive fine-tuning. In: *CVPR*, pp. 4805–4814 (2019)
15. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: *CVPR*, pp. 770–778 (2016)
16. Johnson, A.E., et al.: Mimic-cxr-jpg, a large publicly available database of labeled chest radiographs. arXiv preprint [arXiv:1901.07042](https://arxiv.org/abs/1901.07042) (2019)
17. Kermany, D., Zhang, K., Goldbaum, M.: Large dataset of labeled optical coherence tomography (OCT) and chest x-ray images. *Mendeley Data* **3**, 10–17632 (2018)
18. Li, Z., Zhou, F., Chen, F., Li, H.: Meta-SGD: learning to learn quickly for few-shot learning. arXiv preprint [arXiv:1707.09835](https://arxiv.org/abs/1707.09835) (2017)
19. Nichol, A., Achiam, J., Schulman, J.: On first-order meta-learning algorithms. arXiv preprint [arXiv:1803.02999](https://arxiv.org/abs/1803.02999) (2018)
20. Ren, M., Zeng, W., Yang, B., Urtasun, R.: Learning to reweight examples for robust deep learning. In: *ICML*, pp. 4334–4343. PMLR (2018)
21. Riasatian, A., et al.: Fine-tuning and training of densenet for histopathology image representation using TCGA diagnostic slides. *Med. Image Anal.* **70**, 102032 (2021)
22. Ro, Y., Choi, J.Y.: Autolr: layer-wise pruning and auto-tuning of learning rates in fine-tuning of deep networks. In: *AAAI*, vol. 35, pp. 2486–2494 (2021)
23. Tajbakhsh, N., et al.: Convolutional neural networks for medical image analysis: full training or fine tuning? *IEEE TMI* **35**(5), 1299–1312 (2016)
24. Vrbančič, G., Podgorelec, V.: Transfer learning with adaptive fine-tuning. *IEEE Access* **8**, 196197–196211 (2020)
25. Wang, G., et al.: Interactive medical image segmentation using deep learning with image-specific fine tuning. *IEEE TMI* **37**(7), 1562–1573 (2018)

26. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? In: NeurIPS, vol. 27 (2014)
27. Zhang, C., Chen, Y., Liu, L., Liu, Q., Zhou, X.: HiCo: hierarchical contrastive learning for ultrasound video model pretraining. In: Wang, L., Gall, J., Chin, T.J., Sato, I., Chellappa, R. (eds.) ACCV 2022. LNCS, vol. 13846, pp. 229–246. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-26351-4_1
28. Zhou, H.-Y., Yu, S., Bian, C., Hu, Y., Ma, K., Zheng, Y.: Comparing to learn: surpassing imagenet pretraining on radiographs by comparing image representations. In: Martel, A.L., et al. (eds.) MICCAI 2020. LNCS, vol. 12261, pp. 398–407. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-59710-8_39
29. Zhou, Z., et al.: Models genesis: generic autodidactic models for 3D medical image analysis. In: Shen, D., et al. (eds.) MICCAI 2019. LNCS, vol. 11767, pp. 384–393. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-32251-9_42