SCALING SUPERVISED LOCAL LEARNING WITH AUGMENTED AUXILIARY NETWORKS

Chenxiang Ma[1], **Jibin Wu**[1*], **Chenyang Si**[2], **Kay Chen Tan**[1]
[1]The Hong Kong Polytechnic University, Hong Kong SAR, China
[2]Nanyang Technological University, Singapore

A APPENDIX

A.1 THEORETICAL ANALYSIS ON THE PARALLELIZATION OF AUGLOCAL

To analyze the parallelization, we compare the training time of AugLocal with that of BP. For simplicity, we assume the forward time t_f and backward time t_b of each layer to be the same. We denote the maximum depth of the auxiliary networks as d and the depth of the primary network as $L+1$, consistent with the notations in our paper. N denotes the number of training iterations.

For BP training, the time to train N iterations can be calculated as $(L+1)(t_f+t_b)N$.

In AugLocal, we define a local layer as a hidden layer along with its associated auxiliary networks. The training time for any local layer ℓ per iteration can be represented as $(\ell-1)t_f+(d+1)(t_f+t_b)$. By parallelizing the training of these local layers once their inputs are available, the time of $(d+1)(t_f+t_b)$ can be shared among all local layers. Furthermore, starting from the second iteration, the forward pass of the $(\ell-1)^{th}$ hidden layer can be parallelized with the backward pass of the ℓ^{th} auxiliary network. Based on these considerations, the training time of AugLocal for N iterations can be calculated as $t_f L+(d+1)(t_f+t_b)N$, which is approximated to $(d+1)(t_f+t_b)N$ after omitting the constant term.

Consequently, the ratio of the training time between AugLocal and BP is approximately $\frac{d+1}{L+1}$. This suggests that as the maximum depth of the auxiliary network d decreases, AugLocal demonstrates higher parallelization and faster training speed compared to BP. It is worth noting that to achieve the theoretical training speed-up, we need the customized parallel implementation that we consider as future work.

A.2 IMPLEMENTATION DETAILS

Our experiments are based on four widely used benchmark datasets (i.e., CIFAR-10 (Krizhevsky et al., 2009), SVHN (Netzer et al., 2011), STL-10 (Coates et al., 2011), and ImageNet (Deng et al., 2009)). We compare our proposed AugLocal method with the end-to-end backpropagation (BP) (Rumelhart et al., 1985) algorithm and three state-of-the-art supervised local learning methods, including DGL (Belilovsky et al., 2020), PredSim (Nøkland & Eidnes, 2019), and InfoPro (Wang et al., 2021). We re-implement all of these methods in PyTorch using their official implementations [1]. We utilize consistent training configurations across all learning methods. All experiments are conducted on a machine equipped with 10× NVIDIA RTX3090.

Datasets CIFAR-10 (Krizhevsky et al., 2009) dataset consists of 60K 32×32 colored images that are categorized into 10 classes with 50K images for training and 10K images for test. We use the standard data augmentation (He et al., 2016; Nøkland & Eidnes, 2019; Wang et al., 2021) in the training set, where 4 pixels are padded on each side of samples followed by a 32×32 crop and a random horizontal flip. SVHN (Netzer et al., 2011) dataset contains 32×32 digit images, each with a naturalistic background in RGB format. The standard split of 73,257 images for training and 26,032 images for test is adopted. Following Tarvainen & Valpola (2017); Wang et al. (2021), we

*Corresponding author: jibin.wu@polyu.edu.hk

[1]InfoPro: `https://github.com/blackfeather-wang/InfoPro-Pytorch`, PredSim: `https://github.com/anokland/local-loss`, and DGL: `https://github.com/eugenium/DGL`.

Table 1: Auxiliary networks in local learning methods for the three stages of ResNet-110. '/' is used to separate two auxiliary networks for two local losses in both PredSim and InfoPro. The former network is used for the cross-entropy loss, and the latter one serves another loss function.

Stage	PredSim	InfoPro	DGL	AugLocal ($d = 2$)
1	AP-10FC / 16C3	32C3-AP-128FC-10FC / 12C3-3C3	AP-16C1-16C1-16C1-AP-64FC-64FC-10FC	64R-AP-10FC
2	AP-10FC / 32C3	64C3-AP-128FC-10FC / 12C3-3C3	AP-32C1-32C1-32C1-AP-128FC-128FC-10FC	64R-AP-10FC
3	AP-10FC / 64C3	64C3-AP-128FC-10FC / 12C3-3C3	AP-64C1-64C1-64C1-AP-256FC-256FC-10FC	64R-AP-10FC

Table 2: Comparison of computational costs including FLOPs, GPU memory and computational overhead among PredSim, DGL, InfoPro and our proposed AugLocal method as well as BP and gradient checkpoint (Chen et al., 2016) on CIFAR-10 using the ResNet-110 architecture.

	BP	Gradient Checkpoint	PredSim	DGL	InfoPro
FLOPs (G)	0.25	0.25	0.25	0.26	0.34
GPU Memory (GB)	9.27	3.03	1.54	1.61	3.98
Computational Overhead (Wall-clock Time)	-	34.1%	65.9%	76.2%	292.3%
Acc.	94.61±0.18	94.61±0.18	74.95±0.36	85.69±0.32	86.95±0.46
	AugLocal ($d = 2$)	AugLocal ($d = 3$)	AugLocal ($d = 4$)	AugLocal ($d = 5$)	AugLocal ($d = 6$)
FLOPs (G)	0.63	0.69	0.80	0.98	1.13
GPU Memory (GB)	1.71	1.62	1.70	1.71	1.72
Computational Overhead (Wall-clock Time)	87.2%	115.9%	135.6%	180.8%	214.6%
Acc.	90.98±0.05	92.62±0.22	93.22±0.17	93.75±0.20	93.96±0.15

augment training samples by padding 2 pixels on each side of images followed by a 32×32 crop. STL-10 (Coates et al., 2011) provides 5K labeled images for training and 8K labeled images for test. The size of each image is 96×96. Data augmentation is performed by 4×4 random translation followed by random horizontal flip (Wang et al., 2021). ImageNet (Deng et al., 2009) is a 1,000-class dataset with 1.2 million images for training and 50,000 images for validation. Following He et al. (2016); Huang et al. (2017); Wang et al. (2021), a 224×224 random crop followed by random horizontal flip is adopted for training samples, and a 224×224 resize and a central crop are applied for test samples.

Training setups For CIFAR-10, SVHN, and STL-10 experiments using ResNet-32 (He et al., 2016), ResNet-110 (He et al., 2016), and VGG19 (Simonyan & Zisserman, 2014), we use the SGD optimizer with a Nesterov momentum of 0.9 and the L2 weight decay factor of 1e-4. We adopt a batch size of 1024 on CIFAR-10 and SVHN and a batch size of 128 on STL10. We train the networks for 400 epochs, setting the initial learning rate to 0.8 for CIFAR-10/SVHN and 0.1 for STL-10, with the cosine annealing scheduler (Loshchilov & Hutter, 2019). For ImageNet experiments, we train VGG13 (Simonyan & Zisserman, 2014) with an initial learning rate of 0.1 for 90 epochs, and train ResNet-34 (He et al., 2016) and ResNet-101 (He et al., 2016) with initial learning rates of 0.4 and 0.2 for 200 epochs, respectively. We set batch sizes of VGG13, ResNet-34, and ResNet-101 to 256, 1024, and 512, respectively. We keep other training configurations consistent with the ones on CIFAR-10. It is worth noting that, to reduce the computational costs of auxiliary networks, we change the number of hidden neurons in each auxiliary network's classifier from 4096 to 512 on VGG13.

A.3 Comparison of computational costs among local learning methods

Auxiliary networks We keep the original configurations as stated in their respective papers for the auxiliary networks of DGL, PredSim, and InfoPro in our experiments. The auxiliary networks in these local learning baselines and AngLocal ($d = 2$) are provided in Table 1. For clarity, we show the auxiliary nets based on the three stages of ResNet-110, each stage with the same number of output channels. We use the following notations: R denotes a residual block, C is a convolutional layer, AP signifies average pooling, and FC indicates a fully-connected layer. C1 and C3 refer to 1×1 and 3×3

convolutional kernel sizes, respectively. The value preceding C, R, and FC denotes the number of output channels.

Computational costs We compare the computational costs including FLOPs, GPU memory, and wall-clock time among BP, PredSim, DGL, InfoPro and our proposed AugLocal method. Note that the wall-clock time across all local learning methods is measured specifically under the sequential implementation setting, where each hidden layer is trained sequentially after receiving a batch of samples. Table 2 demonstrates that AugLocal achieves significantly higher accuracy at slightly higher computational costs than the other methods using the ResNet-110 architecture on CIFAR-10. The accuracy of AugLocal can be further improved by employing deeper auxiliary networks and larger computational costs. Additionally, we further compare AugLocal with gradient checkpointing (Chen et al., 2016). Our results in Table 2 demonstrate that AugLocal can achieve a much lower GPU memory footprint than gradient checkpoint, albeit accompanied by a moderate increase in wall-clock time. It is worth noting that the actual memory overhead of AugLocal does not follow a perfect linear growth due to the PyTorch backward implementation has been optimized for e2e BP training. In our future work, we will explore more efficient CUDA implementation to address this issue.

A.4 Generalization to different ConvNets

To evaluate the generalization ability of AugLocal across different convolutional networks (ConvNets), we conduct experiments on three popular ConvNets: MobileNet (Sandler et al., 2018), EfficientNet (Tan & Le, 2019), and RegNet (Radosavovic et al., 2020).

Network architectures MobileNetV2 (Sandler et al., 2018) is a lightweight architecture and comprises two types of building blocks. One is the inverted bottleneck residual block with a stride of 1, and another is the variant with a stride of 2 for downsizing. Each block contains 3 convolutional layers, including two point-wise convolution and one depth-wise convolution. EfficientNetB0 (Tan & Le, 2019) employs a compound scaling strategy to jointly scale network's depth, width, and resolution, offering a superior performance in terms of efficiency. The building block of EfficientNetB0 is the inverted residual block with an additional squeeze and excitation (SE) layer. RegNetX_400MF (Radosavovic et al., 2020) is derived from a low-dimensional network design space consisting of simple and regular networks. The standard residual bottleneck blocks with group convolution are adopted as its building blocks, each of which comprises a 1×1 convolution, followed by a 3×3 group convolution and a final 1×1 convolution.

Training setups As the minimal indivisible units, the building blocks in the three architectures are their local layers, which are independently trained with local learning rules. For AugLocal, the downsampling operation in auxiliary networks is performed by changing the stride of the corresponding auxiliary layer to 2. Other training configrations are the same as the previous ones on CIFAR-10.

Our experimental results in Table 3 demonstrate that AugLocal consistently obtains comparable accuracy to BP, regardless of the network structure, highlighting the potential of AugLocal to generalize across different network architectures.

Table 3: Performances of AugLocal on different ConvNets. The experiments are conducted on CIFAR-10.

	BP	AugLocal ($d = 3$)	($d = 4$)	($d = 5$)	($d = 6$)
MobileNetV2 ($L = 19$)	94.89±0.15	92.16±0.24	93.94±0.23	94.43±0.04	**94.52±0.08**
EfficientNetB0 ($L = 17$)	93.52±0.15	92.70±0.14	92.84±0.15	93.03±0.16	**93.13±0.08**
RegNetX_400MF ($L = 23$)	95.70±0.12	94.42±0.11	94.72±0.01	94.96±0.12	**95.09±0.10**

A.5 Comparison of local learning methods with comparable FLOPs

This experiment compares AugLocal to other local learning methods with comparable FLOPs. We scale up the auxiliary networks of DGL by using 3×3 convolutions with the same network depth and a multiplier to scale up the channel numbers of the convolutional layers to ensure similar FLOPs as AugLocal. We further implement PredSim and InfoPro, which incorporate additional local losses and auxiliary networks, resulting in higher FLOPs than DGL. The ResNet-110 architecture on the

Table 4: Comparison of local learning methods with comparable FLOPs on ResNet-110.

Method	$(d = 2)$	$(d = 3)$	$(d = 4)$	$(d = 5)$	$(d = 6)$
AugLocal	90.98±0.05	92.62±0.22	93.22±0.17	93.75±0.20	93.96±0.15
DGL	83.03±0.24	85.82±0.08	87.84±0.46	89.19±0.20	90.01±0.05
PredSim	72.06±0.63	80.16±0.47	86.00±0.56	88.27±0.72	88.34±0.34
InfoPro	83.71±0.20	89.14±0.17	90.75±0.22	91.45±0.05	92.10±0.17

CIFAR-10 dataset is adopted in this experiment. Our results in Table 4 consistently demonstrate that AugLocal outperforms the other methods with similar FLOPs, reaffirming the effectiveness of our approach in constructing auxiliary networks for improved performance in supervised local learning.

A.6 Ablation Study of AugLocal's Auxiliary Networks

We conduct ablation experiments to investigate the impact of altering the auxiliary architecture on AugLocal's performance. In this ablation study, we focus on AugLocal with a depth (d) of 6, which equals the depth of DGL Belilovsky et al. (2020). We use the ResNet-110 architecture that has three stages, each having the same number of output channels. To align with DGL, which uses the same auxiliary networks for layers in the same stage, we gradually modify AugLocal's auxiliary architectures to match those of DGL.

Initially, we replace the auxiliary networks in the second stage with a repetitive selection strategy combined with downsampling while keeping the other two stages unchanged. This modification results in a 1.21% accuracy drop. Subsequently, we remove the downsampling operation, leading to a further accuracy drop of 1.02%. Based on these modified auxiliary architectures, we replace the first stage layers with the repetitive auxiliary networks and downsampling, resulting in an accuracy degradation of around 1.5%. Finally, we replace all residual blocks in the auxiliary networks with 3×3 convolutional layers while maintaining the same number of channels. This change significantly affects the accuracy, resulting in a drop to 88.28%. It is worth noting that DGL further adopts convolutional 1×1 layers and fully-connected layers, which achieve a baseline accuracy of 85.69%.

These ablation experiments clearly demonstrate the important role of each component in the auxiliary networks of AugLocal. The results highlight the effectiveness of our approach in constructing auxiliary networks for improved performance in local learning.

We provide the details of auxiliary networks in the ablation experiment in Table 5. For consistency, we adopt the same notations as A.3, with the addition of s2 representing a stride of 2 for downsampling.

Table 5: Results of the ablation study for AugLocal by gradually modifying auxiliary networks to match those of the baseline.

Method	Acc.	Stage 1	Stage 2	Stage 3
AugLocal ($d = 6$)	93.96±0.15	Uniform Selection	Uniform Selection	Uniform Selection
Replace with repe. and downsampling (ds.) in Stage 2	92.75±0.09	Uniform Selection	32Rs2-32R-32R-32R-32R-AP-10FC	Uniform Selection
Replace with repe. in Stage 2	91.73±0.11	Uniform Selection	32R-32R-32R-32R-32R-AP-10FC	Uniform Selection
Replace with repe. and ds. in both Stage 1 and 2	91.40±0.08	16Rs2-16R-16R-16R-16R-AP-10FC	32Rs2-32R-32R-32R-32R-AP-10FC	Uniform Selection
Replace with repe. and ds. in Stage 1 and with repe. in Stage 2	89.92±0.37	16Rs2-16R-16R-16R-16R-AP-10FC	32R-32R-32R-32R-32R-AP-10FC	Uniform Selection
Replace with 3×3 convlutional layers and downsampling in all stages	88.28±0.24	AP-16C3-16C3-16C3-16C3-16C3-AP-10FC	AP-32C3-32C3-32C3-32C3-32C3-AP-10FC	AP-64C3-64C3-64C3-64C3-64C3-AP-10FC
DGL	85.69±0.32	AP-16C1-16C1-16C1-AP-64FC-64FC-10FC	AP-32C1-32C1-32C1-AP-128FC-128FC-10FC	AP-64C1-64C1-64C1-AP-256FC-256FC-10FC

A.7 Results of AugLocal on Downstream Tasks

To evaluate the generalization ability of AugLocal on downstream tasks, we conduct experiments on the challenging COCO dataset (Lin et al., 2014) for object detection and instance segmentation.

Table 6: Influence of pyramidal depth on computational efficiency. This complements Figure **??** with explicit values of FLOPs (G). The FLOPs for BP is 0.25G.

	$\tau = 1$	$\tau = 0.9$	$\tau = 0.8$	$\tau = 0.7$	$\tau = 0.6$
$d = 5$	0.79	0.83	0.87	0.89	0.93
$d = 6$	0.90	0.95	0.99	1.03	1.09
$d = 7$	0.99	1.06	1.11	1.17	1.22
$d = 8$	1.09	1.16	1.22	1.29	1.36
$d = 9$	1.18	1.26	1.35	1.41	1.49

Table 7: Results of AugLocal on the COCO object detection and instance segmentation benchmarks.

Method	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
BP	36.3	56.4	39.4	33.8	53.9	36.1
AugLocal	36.2	56.0	39.1	33.4	53.2	35.7

Following the common practice, we use the pre-trained ResNet-34 on ImageNet as a backbone and integrate it with the Mask R-CNN detector (He et al., 2017). To ensure fair comparisons, we maintain consistent training configurations for both AugLocal and BP. Specifically, we utilize the AdamW optimizer, a $1\times$ training schedule consisting of 12 epochs and a batch size of 16. The results in Table 7 show that AugLocal consistently achieves comparable performance to BP across all average precision (AP) metrics, suggesting that AugLocal can effectively generalize pre-trained models for downstream tasks.

A.8 Convergence Speed

In this experiment, we aim to investigate the convergence speed of our proposed AugLocal method by comparing it with BP and other local learning rules. To this end, we visualize the learning curves of these methods on ResNet-32 and ResNet-110. As shown in Figure 1, AugLocal achieves a faster decrease in the network output loss as compared to other local learning rules. Moreover, as the depth of auxiliary networks increases, the convergence speed of AugLocal improves and gradually approaches the one of BP. This finding reconfirms the efficacy of our AugLocal method in optimizing networks to achieve high performance.

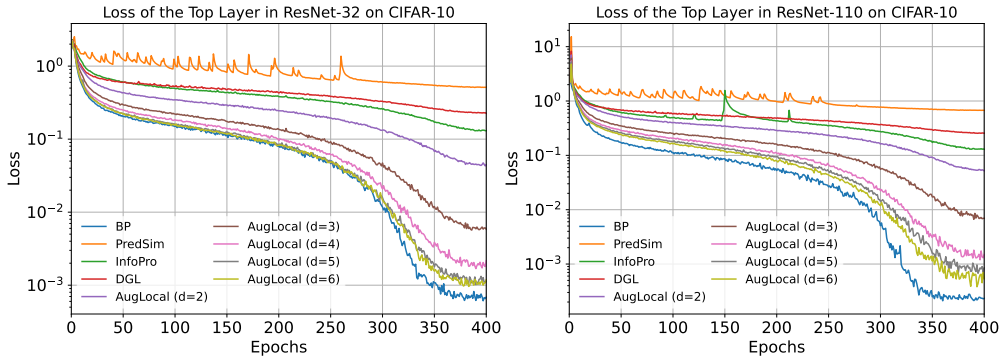

Figure 1: Learning curves of different learning rules on ResNet-32 and ResNet-110. The y-axis is in log scale.

A.9 Synergy between AugLocal and InfoPro

Our proposed AugLocal method is orthogonal to existing supervised local learning works (Nøkland & Eidnes, 2019; Wang et al., 2021) that propose advanced local loss functions. In this part, we investigate the potential benefits of combining AugLocal with InfoPro (Wang et al., 2021). Specifically, each hidden layer additionally incorporates a reconstruction loss with an auxiliary network. Following (Wang et al., 2021), we adopt the auxiliary network containing two convolutions and

Table 8: Performance of AugLocal with the InfoPro loss (Wang et al., 2021) on ResNet-110. The results of AugLocal with the cross-entropy (CE) loss are provided as a baseline.

Loss	$(d = 2)$	$(d = 3)$	$(d = 4)$	$(d = 5)$	$(d = 6)$	$(d = 7)$	$(d = 8)$	$(d = 9)$
CE	90.98±0.05	92.62±0.22	93.22±0.17	93.75±0.20	93.96±0.15	94.03±0.13	94.01±0.06	94.30±0.17
InfoPro	91.59±0.11	92.75±0.50	93.71±0.14	94.11±0.19	94.02±0.09	94.17±0.13	94.29±0.07	94.08±0.18

up-sampling operations. It is worth noting that the original augmented auxiliary network with the cross-entropy loss in each hidden layer keeps unchanged.

The results in Table 8 demonstrate that incorporating the additional reconstruction loss can lead to accuracy improvements in most cases. This suggests that AugLocal can generalize and synergize with advanced local objectives to improve performance further.

REFERENCES

Eugene Belilovsky, Michael Eickenberg, and Edouard Oyallon. Decoupled greedy learning of cnns. In *International Conference on Machine Learning*, pp. 736–745. PMLR, 2020.

Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. Training deep nets with sublinear memory cost. *arXiv preprint arXiv:1604.06174*, 2016.

Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *AISTATS*, pp. 215–223, 2011.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255. Ieee, 2009.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4700–4708, 2017.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pp. 740–755. Springer, 2014.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

Arild Nøkland and Lars Hiller Eidnes. Training neural networks with local error signals. In *International Conference on Machine Learning*, pp. 4839–4850. PMLR, 2019.

Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10428–10436, 2020.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520, 2018.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pp. 6105–6114. PMLR, 2019.

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Yulin Wang, Zanlin Ni, Shiji Song, Le Yang, and Gao Huang. Revisiting locally supervised learning: an alternative to end-to-end training. In *International Conference on Learning Representations (ICLR)*, 2021.