Momentum Auxiliary Network for Supervised Local Learning

Junhao Su[1*], Changpeng Cai[1*], Feiyu Zhu[2*], Chenghao He[4]
Xiaojie Xu[5], Dongzhi Guan[1✉], and Chenyang Si[3✉]

[1] Southeast University
[2] University of Shanghai for Science and Technology
[3] Nanyang Technological University
[4] East China University of Science and Technology
[5] The Hong Kong University of Science and Technology

Abstract. Deep neural networks conventionally employ end-to-end backpropagation for their training process, which lacks biological credibility and triggers a locking dilemma during network parameter updates, leading to significant GPU memory use. Supervised local learning, which segments the network into multiple local blocks updated by independent auxiliary networks. However, these methods cannot replace end-to-end training due to lower accuracy, as gradients only propagate within their local block, creating a lack of information exchange between blocks. To address this issue and establish information transfer across blocks, we propose a Momentum Auxiliary Network (MAN) that establishes a dynamic interaction mechanism. The MAN leverages an exponential moving average (EMA) of the parameters from adjacent local blocks to enhance information flow. This auxiliary network, updated through EMA, helps bridge the informational gap between blocks. Nevertheless, we observe that directly applying EMA parameters has certain limitations due to feature discrepancies among local blocks. To overcome this, we introduce learnable biases, further boosting performance. We have validated our method on four image classification datasets (CIFAR-10, STL-10, SVHN, ImageNet), attaining superior performance and substantial memory savings. Notably, our method can reduce GPU memory usage by more than 45% on the ImageNet dataset compared to end-to-end training, while achieving higher performance. The Momentum Auxiliary Network thus offers a new perspective for supervised local learning. Our code is available at: `https://github.com/JunhaoSu0/MAN`.

Keywords: Local Learning · Image Classification · Momentum Auxiliary Network

*Equal Contribution.

✉Corresponding Authors: Chengyang Si (chenyang.si.mail@gmail.com) and Dongzhi Guan (guandongzhi@seu.edu.cn).

Fig. 1: Comparison between different methods with MAN and the original methods in terms of accuracy. Results are obtained using ResNet-110 (K=55) on the various datasets.

1 Introduction

In deep learning, the prevalent use of end-to-end backpropagation [35] is essential for training complex neural networks, which requires heavy computational work for loss evaluation and successive gradient descent through network layers to refine parameters [17, 24, 29]. This method starkly contrasts with the local signal processing of biological synapses [6, 7, 40] and imposes a 'locking' constraint [19] that delays parameter updates until the entire forward and backward pass is complete. Such constraints exacerbate challenges like reduced parallelism and increased GPU memory usage [28,38], compromising training efficiency and scalability. An emerging alternative to traditional end-to-end (E2E) training is Local Learning [2,3,5,18,19,29,32,37,41], which promises to mitigate the drawbacks of traditional methods. These methods divide the network into discrete, gradient-isolated blocks, each of which is updated by its dedicated auxiliary network according to distinct local objectives [3, 32, 37]. This training strategy enables immediate parameter updates upon local error reception, circumventing the sequential update bottleneck of E2E training, thereby significantly enhancing the efficiency of parallel training [13,22]. Moreover, local learning reduces the demand on GPU memory by only retaining gradients for the local and auxiliary networks, thus eliminating the need to store extensive global gradient information and saving computational resources [28, 38]. However, even with these advantages, there is still a significant performance gap between local learning and traditional E2E methods, especially as the network is divided into more local blocks. Current local learning techniques mainly focus on improving the design of auxiliary network structures [3] and making local loss functions better to close this performance gap [32,37]. Yet, these improvements do not completely address the inherent short-sightedness of local learning: the separation into blocks can make each part of the network only focus on its local objectives, possibly ig-

noring the overall objectives of the network. This can lead to the discarding of globally beneficial information due to the lack of inter-block communication.

In this paper, we introduce the Momentum Auxiliary Network (MAN), a novel network architecture designed to mitigate the inherent limitations of supervised local learning by enhancing inter-block communication. Specifically, MAN not only accepts the current local block as input but also absorbs the parameters of its next block. Upon completion of a forward pass, parameter updates are performed using local gradients refined through Exponential Moving Average (EMA) techniques [14]. This novel approach enables each local block to integrate information from the following block, thereby extending the operational perspective of each block beyond its immediate objectives and further aligning with the global objective of the network. This addresses the critical limitations in local learning frameworks, ensuring a more cohesive alignment with the overarching network objectives. Furthermore, we identify that directly applying EMA parameters is constrained by the feature incongruities across gradient-isolated blocks. To mitigate this, we implement additional learnable bias terms to each auxiliary block, enhancing their ability to share information effectively. The proposed MAN requires only a minimal increase in memory use but significantly improves performance by enhancing the information sharing ability of local blocks The efficacy of the MAN approach is validated on a suite of benchmark image classification datasets, including CIFAR-10 [21], STL-10 [9], SVHN [30], and ImageNet [12]. The results demonstrate the effectiveness of our method in surpassing the limitations of traditional supervised local learning.

The contributions of this paper could be summarized as follows:

- We propose Momentum Auxiliary Network (MAN), designed to facilitate inter-block communication with Exponential Moving Average (EMA), mitigating the short-sightedness issue in conventional supervised local learning techniques and culminating in an elevated overall performance of the network.
- MAN is a versatile, plug-and-play approach that can seamlessly integrate with any supervised local learning technique, markedly amplifying their efficacy while requiring only a negligible increase in memory usage.
- MAN has demonstrated its effectiveness through experiments on benchmark image classification datasets, achieving state-of-the-art performance. Notably, on the ImageNet [12] dataset, it outperforms E2E training while using significantly less memory.

2 Related Work

2.1 Local Learning

Local learning is first proposed as an innovative deep learning algorithm with the intention of utilizing memory more efficiently and adhering more closely to principles of biological plausibility [10]. This approach emerges as a response to

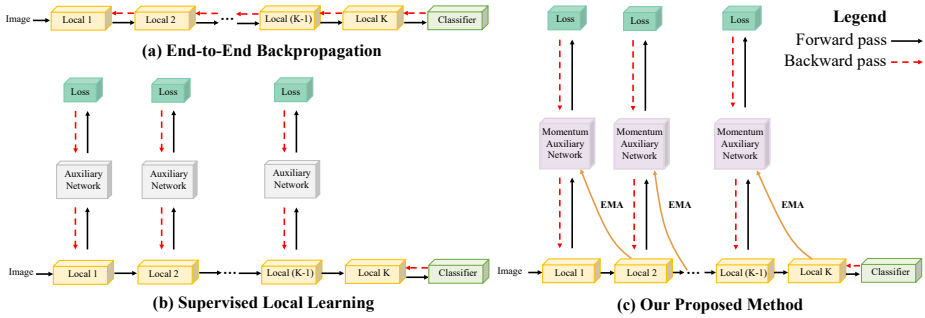


Fig. 2: Comparison of (a) end-to-end backpropagation, (b) other supervised local learning methods, and (c) our proposed method. Unlike E2E, supervised local learning separates the network into K gradient-isolated local blocks.

the limitations presented by global E2E training [16], prompting the development of alternative supervised local learning methodologies. Examples of these methodologies include a differentiable search algorithm, which decouples network blocks for block-level learning and selects a manually assisted network for each local block [33], as well as a self-supervised contrasting loss function that leverages its local learning rules [18, 41]. In supervised local learning, training primarily relies on the design of reasonable supervised local loss functions or the manual construction of auxiliary networks.

Supervised Local Loss: InfoPro [37] presents a method that involves the reconstruction loss for each hidden layer, coupled with cross-entropy loss. This approach forces the hidden layers to preserve vital input information, thereby preventing the early layers from discarding information relevant to the task. PredSim [32] employs layer-wise loss functions for network training. It makes use of two distinct supervised loss functions to generate local error signals for hidden layers. The optimization of these losses takes place during supervised local learning, thus eliminating the necessity to propagate global errors back to the hidden layers. These methods have achieved advanced performance in the field of supervised local learning, but there is still a significant gap compared to the performance of E2E training.

Auxiliary Networks: InfoPro [37] incorporates two auxiliary networks into its approach. One network consists of a single convolutional layer followed by two fully-connected layers, which are employed for the cross-entropy loss. The other network utilizes two convolutional layers with upsampling for the reconstruction loss. Alternatively, DGL [3] designs a structure that uses three consecutive convolutional layers connected to a global pooling layer, followed by three consecutive fully connected layers. Moreover, DGL restricts the parameter of the auxiliary network to only 5% of the main network capacity, offering advantages in terms of speed and memory savings. However, these method still perform poorly when the network is divided into a large number of local blocks.

2.2 Alternative Learning Rules to E2E Training

Due to certain limitations inherent in E2E training, the pursuit of alternative methods to E2E training has gradually garnered attention in recent years [27]. Several efforts have been dedicated to addressing the biologically implausible aspects of E2E training, such as the training methodology of target propagation [4, 23, 25] attempts to train a specialized backward connection by utilizing local reconstruction targets. Additionally, some recent studies have attempted to completely avoid backpropagation in neural networks through forward gradient learning [11, 34]. Meanwhile, the weight transport problem [10]. This has been tackled either by employing distinct feedback connections [1, 26] or by directly disseminating global errors to each hidden unit [8,31]. While these methods have somewhat alleviated the biological implausibility of E2E training, they still rely on global objectives, which fundamentally differ from biological neural networks that rely on local synapses for information transmission. Furthermore, these methods currently struggle to be effective on large datasets [12].

3 Method

3.1 Preliminaries

To set the stage, we first provide a brief overview of traditional end-to-end supervised learning and backpropagation mechanisms. We denote a data sample as x and its corresponding ground-truth label as y. The entire deep network is fragmented into several local blocks. During the forward propagation process, the output from the j-th block serves as the input for the (j+1)-th block, expressed as $x_{j+1} = f_{\theta_j}(x_j)$. Here, θ_j symbolizes the parameters of the j-th local block and $f(\cdot)$ represents the forward calculation of the block. We evaluate the loss function $\mathcal{L}(\hat{y}, y)$ between the output of the last block and the ground truth label, and propagate it back iteratively to the preceding blocks.

Supervised local learning strategies [3, 32, 37] integrate auxiliary networks for local supervision. For each local block, an auxiliary network is affixed. The output from a local block is fed to its corresponding auxiliary network, generating the local supervisory signal as $\hat{y_j} = g_{\gamma_j}(x_{j+1})$. Here, γ_j denotes the parameters of the j-th auxiliary network.

In this setup, we update the parameters of the j-th auxiliary network and local block, γ_j, θ_j, as follows:

$$\gamma_j \leftarrow \gamma_j - \eta_a \times \nabla_{\gamma_j} \mathcal{L}(\hat{y_j}, y) \tag{1}$$

$$\theta_j \leftarrow \theta_j - \eta_l \times \nabla_{\theta_j} \mathcal{L}(\hat{y_j}, y) \tag{2}$$

where η_a, η_l are the learning rates of the auxiliary networks and local blocks, respectively. By attaching auxiliary networks, each local block becomes gradient-isolated and can be updated with local supervision rather than global backpropagation.

3.2 Momentum Auxiliary Network

Existing techniques incorporate supervision signals into individual local blocks, which allows for parallel parameter updates and reduces memory overhead. However, this approach can lead to a short-sightedness problem, where each local block neglects the information from subsequent blocks, ultimately resulting in suboptimal final accuracy.

To address this issue, we propose a comprehensive information exchange module—the Momentum Auxiliary Network (MAN). The MAN employs the Exponential Moving Average (EMA) mechanism [14] as a conduit for transferring information from subsequent blocks to the current block. In the context of MAN, we update the parameters of the j-th auxiliary network and local block as follows:

$$\gamma_j \leftarrow \gamma_j - \eta_a \times \nabla_{\gamma_j} \mathcal{L}(\hat{y_j}, y) \tag{3}$$

$$\gamma_j \leftarrow EMA(\gamma_j, \theta_{j+1}) \tag{4}$$

$$\theta_j \leftarrow \theta_j - \eta_l \times \nabla_{\theta_j} \mathcal{L}(\hat{y_j}, y) \tag{5}$$

where γ_j represents the parameters of the j-th auxiliary network and θ_j represents those of the j-th local block. After updating with local gradients, γ_j undergoes further refinement by incorporating the parameters of the subsequent local block via the EMA, which is a weighted sum operation.

However, through experimental results, it becomes apparent that directly applying EMA parameters to update the auxiliary network has its limitations, and it provides limited improvements to each local block. Upon analyzing, we find that the features learned between each local block vary to a certain degree, which hampers the effectiveness of the EMA update method. As a consequence, a learnable bias is introduced to augment the learning capability of the hidden layers within the local block, while also compensating for the deficiencies of the EMA update method. Therefore, the parameter update method for the j-th auxiliary network can be written as:

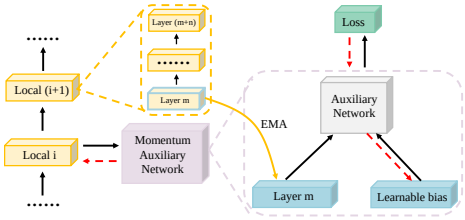


Fig. 3: Details of the Momentum Auxiliary Network. Local (i+1) represents the (i+1)-th gradient-isolated local block, which contains layers from layer m to layer (m+n), totaling n+1 layers (n⩾0). We only use the parameters of the first layer to ensure a balance in GPU memory usage.

$$(\gamma_j, b_j) \leftarrow (\gamma_j, b_j) - \eta_a \times \nabla_{(\gamma_j, b_j)} \mathcal{L}(\hat{y_j}, y) \tag{6}$$

$$\gamma_j \leftarrow EMA(\gamma_j, \theta_{j+1}) \tag{7}$$

where (γ_j, b_j) represent the parameters of the j-th auxiliary network. During this process, we update the parameters γ_j of the j-th auxiliary network jointly through the learnable bias b_j and EMA.

Essentially, the proposed MAN facilitates communication between subsequent blocks through the EMA mechanism [14], effectively resolving the shortsightedness problem inherent in traditional supervised local learning methods. Concurrently, it introduces an independent learnable bias to mitigate information discrepancies caused by feature variations among different local blocks, thereby aligning the output of each block closer to the global target.

In addition to the two innovative methods mentioned above, MAN demonstrates strong versatility. It can be seamlessly integrated into existing supervised local learning methods, and it excels across various datasets, reflecting its flexibility and generalizability.

4 Experiments

4.1 Experimental Setup

We conduct experiments using four widely adopted datasets: CIFAR-10 [21], SVHN [30], STL-10 [9], and ImageNet [12], with ResNets [15] of varying depths serving as the network architectures.

Three state-of-the-art supervised local learning methods are selected for comparison: PredSim [32], DGL [3], and InfoPro [37]. We then partition the network into K local blocks, each containing an approximately equal number of layers. Our proposed Momentum Auxiliary Network is incorporated only into the first K-1 local blocks. The K-th local block does not employ an auxiliary network and is directly connected to the output classifier. We compare these configurations against traditional E2E and original supervised local learning methods to maintain consistent training settings and eliminate confounding variables.

4.2 Implement Details

In our experiments on CIFAR-10 [21], SVHN [30], and STL-10 [9] datasets with ResNet-32 [15] and ResNet-110 [15], we utilize the SGD optimizer with Nesterov momentum set at 0.9 and an L2 weight decay factor of 1e-4. We employ batch sizes of 1024 for CIFAR-10 and SVHN and 128 for STL-10. The training duration spans 400 epochs, starting with initial learning rates of 0.8 for CIFAR-10 / SVHN and 0.1 for STL-10, following a cosine annealing scheduler [9].

In our experiments with ImageNet [12], we adopt different training settings for various architectures. We train VGG13 [36] for 90 epochs with an initial learning rate of 0.025. For ResNet-101 [15], ResNet-152 [15], and ResNeXt-101,32 $\times$ 8d [39], we train them for 90 epochs as well, with initial learning rates of 0.05, 0.05, and 0.025, respectively. We set the batch sizes to 64 for VGG13, 128 for ResNet-101 and ResNet-152, and 64 for ResNeXt-101,32 $\times$ 8d. We maintain consistency with other training configurations as previously described for CIFAR-10 [21].

Based on our multiple experimental results, we select the hyperparameter *momentum* value as 0.995, due to its consistently stable and superior performance. We will provide the experimental results of other *momentum* hyperparameters in the supplementary materials.

4.3 Results on Image Classification Datasets

Results on Image Classification Benchmarks: We start by assessing the accuracy performance of our approach using the CIFAR-10 [21], SVHN [30], and STL-10 [9] datasets. We employ ResNet-32 [15], partitioned into 8 and 16 local blocks, and ResNet-110 [15], divided into 32 and 55 local blocks. As illustrated in Table 1, our MAN significantly bolsters the accuracy of all methods.

Table 1: Perfomance of different networks with varying numbers of local blocks. The average test error is obtained by 5 experiments. The ***** means addition of our MAN.

Dataset	Method	ResNet-32		ResNet-110	
		K = 8 (Test Error)	K = 16 (Test Error)	K = 32 (Test Error)	K = 55 (Test Error)
CIFAR-10 (E2E(ResNet-32)=6.37, E2E(ResNet-110)=5.42)	PredSim [32]	20.62	22.71	22.08	24.74
	PredSim*	**14.29(↓6.33)**	**15.58(↓7.13)**	**17.05(↓5.03)**	**18.02(↓6.72)**
	DGL [3]	11.63	14.08	12.51	14.45
	DGL*	**8.42(↓3.21)**	**9.11(↓4.97)**	**9.65(↓2.86)**	**9.73(↓4.72)**
	InfoPro [37]	11.51	12.93	12.26	13.22
	InfoPro*	**9.32(↓2.19)**	**9.65(↓3.28)**	**9.06(↓3.20)**	**9.77(↓3.45)**
STL-10 (E2E(ResNet-32)=19.35, E2E(ResNet-110)=19.67)	PredSim [32]	31.97	32.90	32.05	33.27
	PredSim*	**29.97(↓2.00)**	**29.99(↓2.91)**	**30.48(↓1.57)**	**31.55(↓1.72)**
	DGL [3]	25.05	27.14	25.67	28.16
	DGL*	**20.74(↓4.31)**	**21.37(↓5.77)**	**22.54(↓3.13)**	**22.69(↓5.47)**
	InfoPro [37]	27.32	29.28	28.58	29.20
	InfoPro*	**23.17(↓4.15)**	**23.54(↓5.74)**	**24.08(↓4.50)**	**24.74(↓4.46)**
SVHN (E2E(ResNet-32)=2.99, E2E(ResNet-110)=2.92)	PredSim [32]	6.91	8.08	9.12	10.47
	PredSim*	**5.54(↓1.37)**	**6.39(↓1.69)**	**7.27(↓1.85)**	**8.24(↓2.23)**
	DGL [3]	4.83	5.05	5.12	5.36
	DGL*	**3.80(↓1.03)**	**4.04(↓1.01)**	**4.08(↓1.04)**	**4.52(↓0.84)**
	InfoPro [37]	5.61	5.97	5.89	6.11
	InfoPro*	**4.49(↓1.12)**	**5.19(↓0.78)**	**4.85(↓1.04)**	**4.99(↓1.12)**

On the CIFAR-10 dataset [21], our method exhibits considerable improvements in diminishing test errors across various methods. In the relatively shallower network of ResNet-32 (K=16), where individual layers function as gradient-isolated local blocks, we record a reduction in test errors for PredSim [32], DGL [3], and InfoPro [37], from 22.71, 14.08, and 12.93 to 15.58, 9.11, and 9.65 respectively. This translates to a performance enhancement exceeding 25% for all methods. Even though the performance across all methods in the comparatively deeper network, ResNet-110 (K=55), is somewhat inferior due to the inherent need for more global information in such networks, our method still delivers exceptional performance. It achieves approximately a 20% improvement across all methods, underscoring the robust effectiveness of MAN in deeper networks.

When applied to other datasets, MAN can also reduce the test error of PredSim [32], DGL [3], and InfoPro [37] by at least 5%, 12%, and 16% on the STL-10 [9] dataset. On the SVHN [30] dataset, our improvements over the three

methods also surpass 20%, 21%, and 13%. As can be seen, the improvement our MAN introduces to all methods is quite remarkable—comparable even to the accuracy of E2E training—and it significantly mitigates the underwhelming performance issue that has continually plagued supervised local learning.

Results on ImageNet: We further validate the effectiveness of our approach on ImageNet [12] using four networks of varying depths (ResNets [15] and VGG13 [36]). As depicted in Table 2, when we employ VGG13 as the backbone and divide the network into 10 blocks, DGL [3] achieves merely a Top1-Error of 35.60 and a Top5-Error of 14.2, representing a substantial gap when compared to the E2E method. However, with the introduction of our MAN, the Top1-Error reduces by 3.61 points, and the Top5-Error decreases by 3.36 points for DGL. This significant enhancement brings the performance closer to the E2E method.

As illustrated in Table 2, when we use ResNet-101 [15], ResNet-152 [15], ResNeXt-101, 32×8d [39] as backbones and divide the network into four blocks, the performance of InfoPro [37] is already below that of E2E. After incorporating our MAN, the Top-1 Error of these three backbone networks can be reduced by approximately 6% compared to the original, surpassing the performance of E2E training. These results underscore the effectiveness of our MAN on the large-scale ImageNet [12] dataset, even when using deeper networks.

Table 2: Results on the validation set of ImageNet

Network	Method	Top1-Error	Top5-Error
ResNet-101	E2E	22.03	5.93
	InfoPro(K=2) [37]	21.85	5.89
	InfoPro*(K=2)	**21.65(↓0.20)**	**5.49(↓0.40)**
	InfoPro(K=4) [37]	22.81	6.54
	InfoPro*(K=4)	**21.73(↓1.08)**	**5.81(↓0.73)**
ResNet-152	E2E	21.60	5.92
	InfoPro(K=2) [37]	21.45	5.84
	InfoPro*(K=2)	**21.23(↓0.22)**	**5.53(↓0.31)**
	InfoPro(K=4) [37]	22.93	6.71
	InfoPro*(K=4)	**21.59(↓1.34)**	**5.89(↓0.82)**
ResNeXt-101, 32 × 8d	E2E	20.64	5.40
	InfoPro(K=2) [37]	20.35	5.28
	InfoPro*(K=2)	**20.11(↓0.24)**	**5.18(↓0.10)**
	InfoPro(K=4) [37]	21.69	6.11
	InfoPro*(K=4)	**20.37(↓1.32)**	**5.34(↓0.77)**
VGG13	E2E	28.41	9.63
	DGL [3]	35.60	14.20
	DGL*(K=10)	**31.99(↓3.61)**	**10.84(↓3.36)**

Training-Accuracy Curve Analysis: As depicted in the accuracy-epoch curve in Fig. 4, our Momentum Auxiliary Network consistently outperforms the original method in terms of accuracy throughout the entire training process. This underscores its reliability and stability during the training process of classification tasks. Moreover, our MAN achieves a higher accuracy earlier in the later stages of training and attains stability sooner, indicating a faster convergence rate—a critical attribute in large-scale and complex tasks.

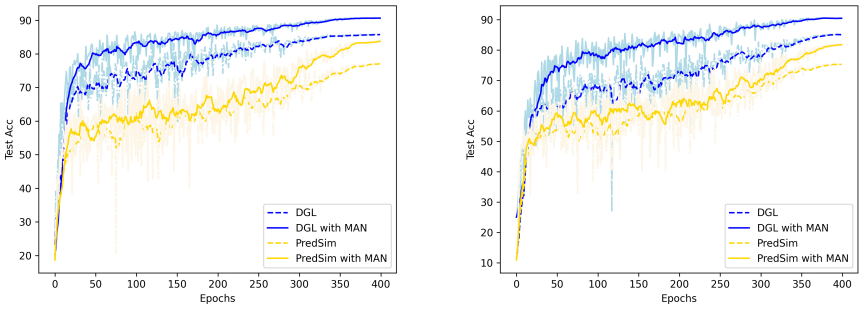


Fig. 4: Training-Accuracy curves, the left uses ResNet-32 (K=16) as the backbone, while the right uses ResNet-110 (K=55). Both are utilizing the CIFAR-10 dataset.

In summary, our Momentum Auxiliary Network significantly enhances the accuracy of traditional supervised local learning methods by promoting information exchange between local blocks. The beneficial information exchange facilitated by Momentum Auxiliary Network significantly mitigates the pervasive issue of shortsightedness within the supervised local learning domain, thereby offering advantages in terms of training stability and convergence speed. This is crucial for training models more effectively and efficiently.

Table 3: Comparison of GPU memory usage using InfoPro as baseline with different backbones on the ImageNet dataset.

Method	ResNet-101 GPU Memory(GB)	ResNet-152 GPU Memory(GB)	ResNeXt-101, 32×8d GPU Memory(GB)
E2E	19.71	26.29	19.22
InfoPro(K=2) [37]	12.06(↓ 38.8%)	15.53(↓ 40.9%)	11.55(↓ 39.9%)
InfoPro*(K=2)	**12.32**(↓ **37.5**%)	**15.93**(↓ **39.4**%)	**11.73**(↓ **38.9**%)
InfoPro(K=4) [37]	10.37(↓ 47.3%)	13.48(↓ 48.7%)	10.24(↓ 46.7%)
InfoPro*(K=4)	**10.69**(↓ **45.8**%)	**13.91**(↓ **47.1**%)	**10.49**(↓ **45.5**%)

Results on GPU memory requirement: Supervised local learning markedly conserves GPU memory by limiting gradient propagation to within local blocks, thereby ensuring that backpropagation transpires exclusively within each respective local block and its corresponding auxiliary network. This strategy significantly diminishes the storage demands for activation parameters and gradient information, which would typically disseminate across local blocks. Consequently, this facilitates substantial savings in GPU memory utilization. Our analysis, featuring a comprehensive comparison of GPU memory usage on the ImageNet dataset, illustrates that the implementation of a Momentum Auxiliary Network

Table 4: Abalation study of MAN. (a) Using DGL as baseline and ResNet-32 (K=16) as backbone on the CIFAR-10 dataset. (b) Using InfoPro as baseline and ResNet-101 (K=4) as backbone on the ImageNet dataset. LB stands for Learnable Bias.

EMA	LB	Test Error
×	×	14. 08
✓	×	**11.07(↓ 3.01)**
✓	✓	**9.11(↓ 4.97)**

(a)

EMA	LB	Top1-Error	Top5-Error
×	×	22.81	6.54
✓	×	**22.09(↓ 0.72)**	**6.07(↓ 0.47)**
✓	✓	**21.73(↓ 1.08)**	**5.81(↓ 0.73)**

(b)

can elicit notable performance enhancements while incurring only a minimal uptick in GPU memory consumption.

As shown in Table 3, it indicates the substantial GPU memory reduction on the ImageNet [12]. When applying our method to InfoPro [37] on ResNet-101 [15], ResNet-152 [15], and ResNeXt-101, 32×8d [39], dividing the backbone into two local blocks results in a GPU memory saving of 37.5% to 39.4%. When the network is divided into four local blocks, the degree of GPU memory savings is further enhanced, exceeding 45%. Combined with Table 2, it can be seen that we achieve better performance with almost half the GPU memory required for end-to-end training. Further analysis of the memory usage comparison with the original method shows that our approach only increases the GPU memory by about 1% over the original method, while achieving a performance improvement of over 5%. These results highlight the excellent balance that the MAN method achieves in terms of GPU memory usage and performance enhancement.

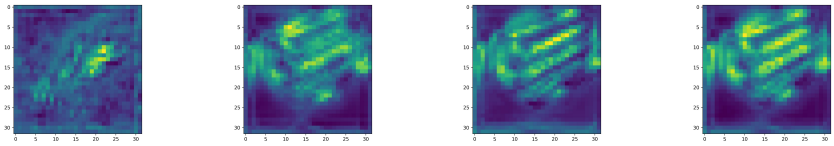
4.4 Ablation Studies

We conduct an ablation study on the CIFAR-10 [21] dataset and ImageNet dataset to assess the impact of the EMA method [14] and learnable bias in the MAN on performance. For this analysis, we use ResNet-32 (K=16) [15] as the backbone and the original DGL [3] method as a comparison baseline.

As shown in Table 4(a), when we only use the EMA [14] in MAN to promote information exchange with subsequent blocks, without adding a learnable bias, the test error decreases from 14.08 to 11.07. When we further add a learnable bias, the test error decreases from 11.07 to 9.11, it is evident that both the EMA method and learnable bias contribute to performance improvement. We hypothesize that the EMA and learnable bias are somewhat complementary in terms of the features they learn. To validate this, we conduct feature visualization.

As depicted in Fig. 5(a), we can observe that the original method without the addition of MAN learns very limited and chaotic features. After adding the EMA method alone, the features it learns are clearly more characteristic and defined, indicating that it indeed receives some global features from the information in the subsequent block. However, there are still many blurry features, which may be due to the EMA parameter update method [14], causing information imbalance due to feature differences between different local blocks. When we

add learnable bias to the original method alone, the features it learns are more concentrated and clear, indicating that the learnable bias significantly enhances the learning ability of the current local block's hidden layer, but it still lacks some details because it does not receive more global information from the subsequent block. When we add the EMA method and learnable bias simultaneously, the learned features are not only clear and specific, but also more comprehensive in detail, proving that their learning abilities are complementary, and greatly improving the performance of the original method. The feature visualization in Fig. 5 verifies our previous thoughts.



(a) (b) (c) (d)

Fig. 5: Feature maps comparison of (a) Original Method without MAN, (b) MAN with only EMA, (c) MAN with only learnable bias and (d) MAN with both EMA and learnable bias. The feature map are obtained using ResNet-32 (K=16) on CIFAR-10.

4.5 The Effectiveness of EMA

To verify the effectiveness of the EMA update method in MAN, we conduct further ablation experiments. As shown in Table 5, using only the parameters of the next layer in MAN results in very limited improvement. In contrast, employing the EMA method to utilize the parameters of the next layer leads to more significant enhancements. This is because the subsequent local block and the current local block learn different features and have different learning objectives, leading to minimal improvement. However, using the EMA method can gradually promote information exchange between local blocks and reduce parameter fluctuations during training, leading to smoother parameter updates. This beneficial interaction brings significant improvements to MAN.

Table 5: Abalation study of EMA. (a) Using DGL as baseline and ResNet-32 (K=16) as backbone on the CIFAR-10 dataset. (b) Using InfoPro as baseline and ResNet-101 (K=4) as backbone on the ImageNet dataset. Parameter signifies whether each local block utilizes the parameters of the adjacent next layer connected to it.

Parameter	EMA	Test Error
×	×	14. 08
✓	×	**12.93**
✓	✓	**11.07**

(a)

Parameter	EMA	Top1-Error	Top5-Error
×	×	22.81	6.54
✓	×	**22.64**	**6.41**
✓	✓	**22.09**	**6.07**

(b)

4.6 Linear Separability Analysis

To demonstrate the effectiveness of our Momentum Auxiliary Network, we freeze the parameters of the main network and train a classifier for each local block to obtain the classification accuracy of each local block. As depicted in Fig. 6, we use DGL [3] as a baseline, and after integrating our method, the accuracy of the earlier layers decrease, while the accuracy of the middle and later layers significantly improve. The decrease in accuracy of the earlier layers suggests that they have learned more generalized features. Although these features are not beneficial for optimizing local objectives, they are beneficial from a global perspective. The middle and later layers, receiving these globally beneficial features, have seen a significant increase in accuracy. These results illustrate that our proposed method has facilitated information interaction between gradient-isolated local blocks. This addresses the shortsightedness issue existing in the current supervised local learning field and significantly enhances their performance.

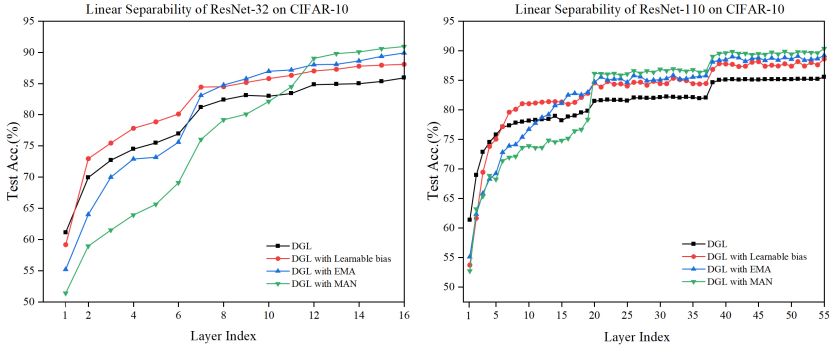


Fig. 6: Comparison of layer-wise linear separability across different learning rules on ResNet-32 and ResNet-110.

4.7 Representation Similarity Analysis

To further validate the efficacy of MAN, we conduct a Centered Kernel Alignment (CKA) experiment [20]. CKA serves as a metric to assess the similarity between feature representations. If a method's CKA score is closer to 1 in relation to the E2E training method, it indicates that the method's feature learning process is more aligned with that of E2E training. As depicted in Fig. 7, we use DGL [3] as a baseline and incorporate our method. It can be observed that whether adding the EMA method or learnable bias alone, the CKA scores significantly improve compared to the original method. When both the EMA and learnable bias are used in MAN, the CKA score further improves and performs more stably. Notably, the most significant increase in the CKA score occurs in the early and late layers. In conjunction with our previous analysis of the linear separability experiment, this is because if the early layers' learning method is closer

to E2E training, they will focus more on learning general features to optimize the global objective, rather than focusing narrowly on local objectives. While this may result in poorer classification capabilities in the early layers, it greatly contributes to the overall performance improvement of the network. Through the analysis of images and experimental results, it can be demonstrated that MAN enables information interaction between gradient-isolated local blocks, solving the myopia problem present in current supervised local learning methods.

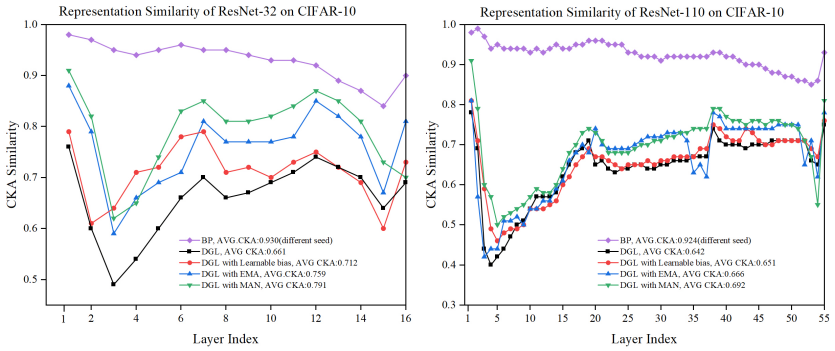


Fig. 7: Assessment of Similarity in Layer-wise Representations. We use Centered Kernel Alignment (CKA) [20] to quantify the degree of similarity in the layer-wise representations between the E2E backpropagation (BP) and our proposed MAN.

5 Conclusion

This study addresses the performance disparities between traditional supervised local learning and end-to-end (E2E) methods in deep learning. We introduce a versatile Momentum Auxiliary Network to tackle the short-sighted problem in the early optimization work related to supervised local learning, facilitating information exchange between gradient-isolated local blocks. We integrate our Momentum Auxiliary Network into three advanced supervised local learning approaches and evaluate their performance across network architectures with varying depths on four widely adopted datasets. The results demonstrate our method's ability to significantly enhance the ultimate output performance of original supervised local learning methods. Particularly when combined with InfoPro [37], our method significantly reduces GPU memory usage while consistently maintaining performance levels closely aligned with E2E approaches.

Limitations and future works: Despite the superior performance on large-scale problems like ImageNet [12], our method still performs less accurately than E2E on some conventional image classification datasets. This may be due to the MAN using too few information interaction layers when the network is divided into a larger number of local blocks. In future work, we could explore deepening these information interaction layers to achieve better precision performance.

Acknowledgement

This work was supported by the National Natural Science Foundation of China (52278154), the Natural Science Foundation of Jiangsu (BK20231429), the Fundamental Research Funds for the Central Universities (2242024RCB0008), and assupport from the program of Zhishan Young Scholar of Southeast University.

References

1. Akrout, M., Wilson, C., Humphreys, P., Lillicrap, T., Tweed, D.B.: Deep learning without weight transport. Advances in neural information processing systems **32** (2019)
2. Belilovsky, E., Eickenberg, M., Oyallon, E.: Greedy layerwise learning can scale to imagenet. In: International conference on machine learning. pp. 583–593. PMLR (2019)
3. Belilovsky, E., Eickenberg, M., Oyallon, E.: Decoupled greedy learning of cnns. In: International Conference on Machine Learning. pp. 736–745. PMLR (2020)
4. Bengio, Y.: How auto-encoders could provide credit assignment in deep networks via target propagation. arXiv preprint arXiv:1407.7906 (2014)
5. Bengio, Y., Lamblin, P., Popovici, D., Larochelle, H.: Greedy layer-wise training of deep networks. Advances in neural information processing systems **19** (2006)
6. Bengio, Y., Lee, D.H., Bornschein, J., Mesnard, T., Lin, Z.: Towards biologically plausible deep learning. arXiv preprint arXiv:1502.04156 (2015)
7. Caporale, N., Dan, Y.: Spike timing–dependent plasticity: a hebbian learning rule. Annu. Rev. Neurosci. **31**, 25–46 (2008)
8. Clark, D., Abbott, L., Chung, S.: Credit assignment through broadcasting a global error vector. Advances in Neural Information Processing Systems **34**, 10053–10066 (2021)
9. Coates, A., Ng, A., Lee, H.: An analysis of single-layer networks in unsupervised feature learning. In: Proceedings of the fourteenth international conference on artificial intelligence and statistics. pp. 215–223. JMLR Workshop and Conference Proceedings (2011)
10. Crick, F.: The recent excitement about neural networks. Nature **337**(6203), 129–132 (1989)
11. Dellaferrera, G., Kreiman, G.: Error-driven input modulation: solving the credit assignment problem without a backward pass. In: International Conference on Machine Learning. pp. 4937–4955. PMLR (2022)
12. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
13. Gomez, A.N., Key, O., Perlin, K., Gou, S., Frosst, N., Dean, J., Gal, Y.: Interlocking backpropagation: Improving depthwise model-parallelism. The Journal of Machine Learning Research **23**(1), 7714–7741 (2022)
14. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9729–9738 (2020)
15. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)

16. Hinton, G.E., Osindero, S., Teh, Y.W.: A fast learning algorithm for deep belief nets. Neural computation **18**(7), 1527–1554 (2006)
17. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4700–4708 (2017)
18. Illing, B., Ventura, J., Bellec, G., Gerstner, W.: Local plasticity rules can learn deep representations using self-supervised contrastive predictions. Advances in Neural Information Processing Systems **34**, 30365–30379 (2021)
19. Jaderberg, M., Czarnecki, W.M., Osindero, S., Vinyals, O., Graves, A., Silver, D., Kavukcuoglu, K.: Decoupled neural interfaces using synthetic gradients. In: International conference on machine learning. pp. 1627–1635. PMLR (2017)
20. Kornblith, S., Norouzi, M., Lee, H., Hinton, G.: Similarity of neural network representations revisited. In: International conference on machine learning. pp. 3519–3529. PMLR (2019)
21. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
22. Laskin, M., Metz, L., Nabarro, S., Saroufim, M., Noune, B., Luschi, C., Sohl-Dickstein, J., Abbeel, P.: Parallel training of deep networks with local updates. arXiv preprint arXiv:2012.03837 (2020)
23. Le Cun, Y.: Learning process in an asymmetric threshold network. In: Disordered systems and biological organization, pp. 233–240. Springer (1986)
24. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. nature **521**(7553), 436–444 (2015)
25. Lee, D.H., Zhang, S., Fischer, A., Bengio, Y.: Difference target propagation. In: Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2015, Porto, Portugal, September 7-11, 2015, Proceedings, Part I 15. pp. 498–515. Springer (2015)
26. Lillicrap, T.P., Cownden, D., Tweed, D.B., Akerman, C.J.: Random synaptic feedback weights support error backpropagation for deep learning. Nature communications **7**(1), 13276 (2016)
27. Lillicrap, T.P., Santoro, A., Marris, L., Akerman, C.J., Hinton, G.: Backpropagation and the brain. Nature Reviews Neuroscience **21**(6), 335–346 (2020)
28. Löwe, S., O'Connor, P., Veeling, B.: Putting an end to end-to-end: Gradient-isolated learning of representations. Advances in neural information processing systems **32** (2019)
29. Mostafa, H., Ramesh, V., Cauwenberghs, G.: Deep supervised learning using local errors. Frontiers in neuroscience **12**, 608 (2018)
30. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning (2011)
31. Nøkland, A.: Direct feedback alignment provides learning in deep neural networks. Advances in neural information processing systems **29** (2016)
32. Nøkland, A., Eidnes, L.H.: Training neural networks with local error signals. In: International conference on machine learning. pp. 4839–4850. PMLR (2019)
33. Pyeon, M., Moon, J., Hahn, T., Kim, G.: Sedona: Search for decoupled neural networks toward greedy block-wise learning. In: International Conference on Learning Representations (2020)
34. Ren, M., Kornblith, S., Liao, R., Hinton, G.: Scaling forward gradient with local losses. arXiv preprint arXiv:2210.03310 (2022)
35. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. nature **323**(6088), 533–536 (1986)

36. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
37. Wang, Y., Ni, Z., Song, S., Yang, L., Huang, G.: Revisiting locally supervised learning: an alternative to end-to-end training. arXiv preprint arXiv:2101.10832 (2021)
38. Wu, B., Nair, S., Martin-Martin, R., Fei-Fei, L., Finn, C.: Greedy hierarchical variational autoencoders for large-scale video prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2318–2328 (2021)
39. Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1492–1500 (2017)
40. Xie, X., Seung, H.S.: Equivalence of backpropagation and contrastive hebbian learning in a layered network. Neural computation **15**(2), 441–454 (2003)
41. Xiong, Y., Ren, M., Urtasun, R.: Loco: Local contrastive representation learning. Advances in neural information processing systems **33**, 11142–11153 (2020)