6. Appendix

6.1. Pseudocodes

In this subsection, we provide the pseudocodes for split, aggregation and inference in Algorithm 2, 3 and 4.

Algorithm 2: `split`

Inputs: Model M with weights θ and N layers
Parameters: Split ratio pair (s_d, s_w)
Outputs: Split model M' with weights θ'

$M' \leftarrow M, \theta' \leftarrow \theta$
Remove all layers in M' after $\lfloor s_d N \rfloor$-th layer
for $\mathbf{W} \in \theta'$ **do**
  $\mathbf{Z} \leftarrow \texttt{index}(\texttt{size}(\mathbf{W}, s_w))$
  $\mathbf{W} \leftarrow \mathbf{W}[\mathbf{Z}]$
end for
return M' with θ'

Algorithm 3: `aggregate`

Inputs: Global model weights θ, set of local model weights $\{\theta^{(k)}\}_{k \in S}$, split ratio pairs $\{(s_d^{(l)}, s_w^{(l)})\}_{l=1}^{L}$
Outputs: Aggregated model weights θ'

$\theta' \leftarrow \theta$
for $\mathbf{W}$ in θ' **do**
  $\widetilde{\mathbf{W}} \leftarrow \texttt{zeros\_like}(\mathbf{W})$
  $\mathbf{C} \leftarrow \texttt{zeros\_like}(\mathbf{W})$
  for client $k \in \mathcal{S}$ **do**
    if $\texttt{key}(\mathbf{W}) \in \texttt{key}(\theta^{(k)})$ **then**
      $\mathbf{Z} \leftarrow \texttt{index}(\texttt{size}(\mathbf{W}), s_w^{(l_k)})$
      $\widetilde{\mathbf{W}}[\mathbf{Z}] \leftarrow \widetilde{\mathbf{W}}[\mathbf{Z}] + \mathbf{W}_k$
      $\mathbf{C}[\mathbf{Z}] \leftarrow \mathbf{C}[\mathbf{Z}] + 1$
    end if
  end for
  $\overline{\mathbf{C}} = \mathbf{C} > 0$
  $\mathbf{W}[\overline{\mathbf{C}}] \leftarrow \widetilde{\mathbf{W}}[\overline{\mathbf{C}}]$
  $\mathbf{W}[\overline{\mathbf{C}}] \leftarrow \mathbf{W}[\overline{\mathbf{C}}] / \mathbf{C}[\overline{\mathbf{C}}]$
end for
return θ'

For inference, based on the complexity level of the client l, the global model M is split and inference is performed on the local model M_l. If adaptive inference flag a is enabled, this procedure also enables continuously outputting early exit predictions. We illustrate the results obtained with the early exiting capability of the ScaleFL-trained global model in Appendix 6.3.

6.2. Split Configuration Details and Model Statistics

We report the local model statistics and used split ratios for all model architectures in Table 6 to 9. For instance, Level 2-dw represents the statistics for the level-2 model

Algorithm 4: `Inference`

Inputs: Global model M, test input $\mathbf{X}$, complexity level of the client for inference l, split ratio pairs $\{(s_d^{(l')}, s_w^{(l')})\}_{l'=1}^{L}$, adaptive inference flag $a \in \{0, 1\}$
Outputs: Prediction $\hat{\boldsymbol{y}}$

Split: $M_l \leftarrow \texttt{split}(M; s_d^{(l)}, s_w^{(l)})$
for $l' \in \{1, \dots l\}$ **do**
  Calculate $\mathbf{H}_{l'l}$ using Eq. (3)
  if a and $l' \neq l$ **then**
    Calculate $\hat{\boldsymbol{y}}_{l'l}$ using Eq. (4)
    yield $\hat{\boldsymbol{y}}_{l'l}$
  end if
end for
Calculate $\hat{\boldsymbol{y}}_{ll}$ using Eq. (4)
return $\hat{\boldsymbol{y}}_{ll}$

ResNet110	Split Ratios		Cost		
Level	s_d	s_w	#PARAMS	#FLOPS	Latency (ms)
4	1.00	1.00	1.73 M	253.1 M	47.2
3-w	1.00	0.70	0.82 M	119.7 M	37.7
2-w	1.00	0.45	0.44 M	63.2 M	35.6
1-w	1.00	0.30	0.21 M	28.0 M	28.3
3-dw	0.88	0.75	0.86 M	138.5 M	39.2
2-dw	0.77	0.70	0.46 M	99.7 M	33.1
1-dw	0.66	0.70	0.21 M	83.4 M	27.9

Table 6. Split ratios and resulting local model statistics (#PARAMS, #FLOPS, latency) for ResNet110.

after two-dimensional splitting, whereas the row Level 2-w contains the statistics for the level-2 model after splitting along width only, as in HeteroFL. Depending on the model architecture, splitting along depth and width may have different effects on #PARAMs and #FLOPs. We also observe that the relation between #FLOPs and latency is not linear. Even though the resulting models may have the same #FLOPs, latency in practice also depends on the number of memory access/allocation operations. Since our approach decreases the number of layers during downscaling, local models perform fewer memory-related operations during training and inference, which can provide speed gain depending on the model architecture, implementation and hardware.

6.3. Adaptive Inference

Another advantage that ScaleFL provides is the fact that the global model is capable of performing adaptive inference with early exits. This functionality is particularly attractive in inference scenarios where a batch of samples have to be processed within a time budget. In this situation, the multi-exit model can adaptively exit at earlier/later exits depending on the difficulty of the input samples while

MSDNet24	Split Ratios		Cost		
Level	s_d	s_w	#PARAMS	#FLOPS	Latency (ms)
4	1.00	1.00	2.64 M	101.9 M	46.6
3-w	1.00	0.75	1.33 M	48.1 M	42.1
2-w	1.00	0.50	0.66 M	25.6 M	34.9
1-w	1.00	0.35	0.30 M	12.4 M	31.6
3-dw	0.875	0.85	1.38 M	59.7 M	40.9
2-dw	0.75	0.70	0.64 M	32.4 M	37.8
1-dw	0.625	0.65	0.28 M	15.8 M	28.6

Table 7. Split ratios and resulting local model statistics (#PARAMS, #FLOPS, latency) for at each level for MSDNet24.

EfficientNetB4	Split Ratios		Cost		
Level	s_d	s_w	#PARAMS	#FLOPS	Latency (ms)
4	1.00	1.00	17.2 M	1223.5 M	212.8
3-w	1.00	0.70	8.6 M	613.3 M	120.3
2-w	1.00	0.48	4.2 M	299.2 M	81.3
1-w	1.00	0.33	2.1 M	148.8 M	59.5
3-dw	0.94	0.82	8.6 M	799.6 M	146.9
2-dw	0.81	0.65	4.5 M	641.5 M	110.1
1-dw	0.69	0.65	2.1 M	534.1 M	103.9

Table 8. Split ratios and resulting local model statistics (#PARAMS, #FLOPS, latency) for at each level for EfficientNet-B4 (224x224 input resolution).

BERT	Split Ratios		Cost		
Level	s_d	s_w	#PARAMS	#FLOPS	Latency (ms)
4	1.00	1.00	109.5 M	163.5 M	155.9
3-w	1.00	0.70	57.3 M	106.2 M	84.5
2-w	1.00	0.45	27.1 M	72.0 M	44.8
1-w	1.00	0.30	14.7 M	22.3 M	32.9
3-dw	0.75	0.75	54.8 M	85.3 M	79.7
2-dw	0.50	0.55	26.1 M	42.9 M	26.6
1-dw	0.33	0.40	13.7 M	22.2 M	15.3

Table 9. Split ratios and resulting local model statistics (#PARAMS, #FLOPS, latency) for at each level for BERT. Measurements are performed with a sentence with 64 words.

satisfying the given inference budget. We follow the early exiting approach described in (11), where the difficulty of a sample is defined based on the maximum prediction score.

We illustrate this capability in Figure 5 to 9 for ResNet110 on CIFAR-10. In this setting, the multi-exit model achieves 85.15% accuracy with an inference time of 35 ms per sample by exiting 38.28%, 23.74%, 18.91% and 19.07% of the samples at each exit. We note that the accuracy achieved by the last exit is 85.53% with 47.2 ms/sample. Therefore, employing adaptive inference with early exiting on the model trained with ScaleFL enables preserving the performance while significantly reducing the latency.

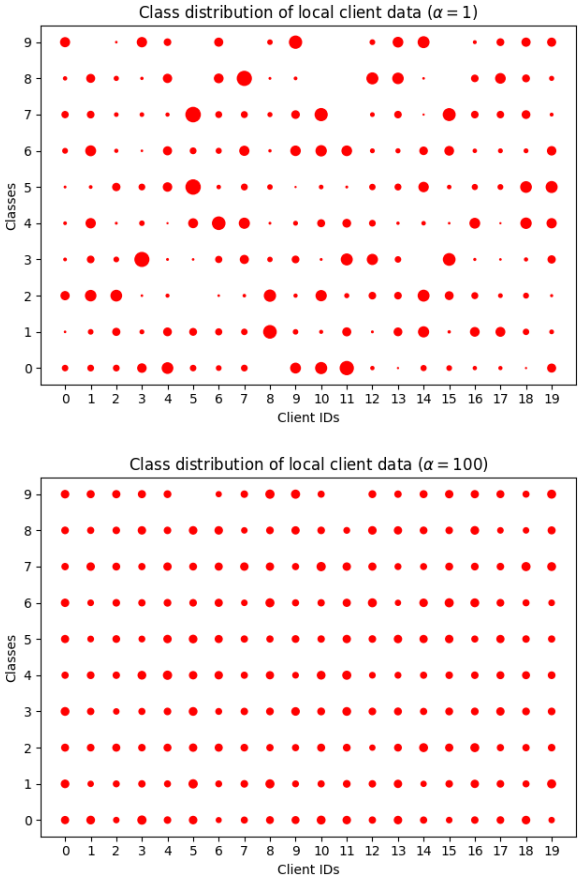
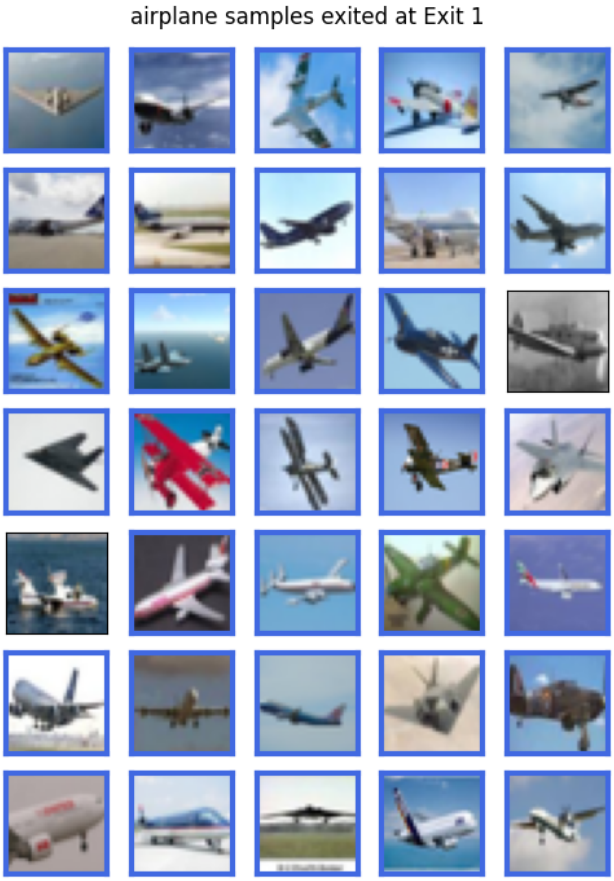


Figure 4. Class distribution of local CIFAR10 training data for the first 20 clients, on two levels of data heterogeneity. Size of the dots represent the number of data samples for each class at each client.



(a) Exit 1

(b) Exit 2

(c) Exit 3

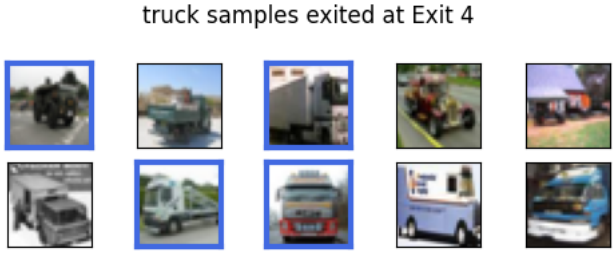
(d) Exit 4

Figure 5. Airplane samples from CIFAR-10 test set. Each subfigure illustrates the samples at the corresponding exit of ResNet110 with four exits under the average inference budget of 35 ms/sample. Blue border indicates correct predictions. Orange border indicates the incorrectly predicted samples but correctly predicted by the base model. No border indicates the samples incorrectly predicted both at that exit and by the base model. 10% of the sample are visualized.

(a) Exit 1

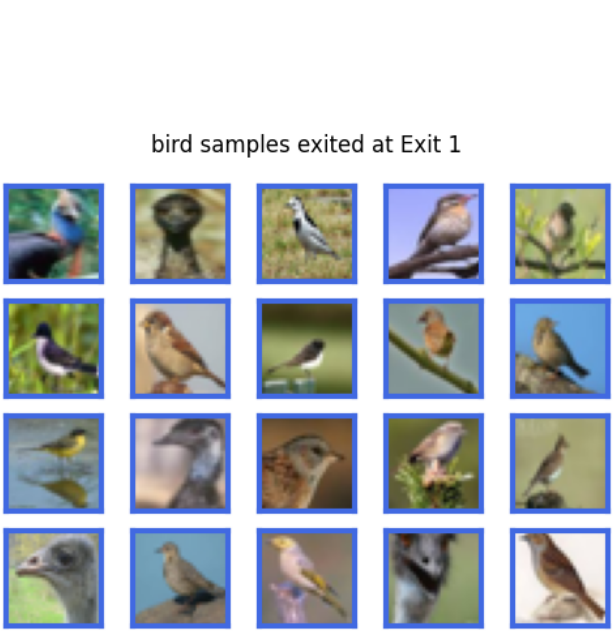
(b) Exit 2

(c) Exit 3



(d) Exit 4

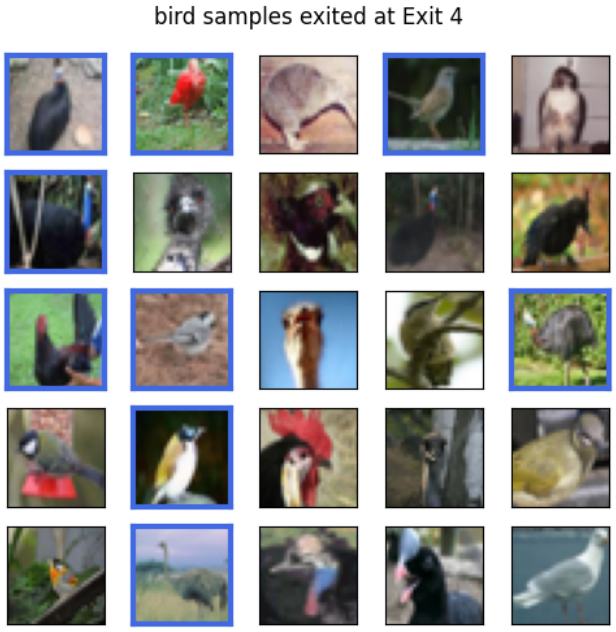
Figure 6. Truck samples from CIFAR-10 test set. Each subfigure illustrates the samples at the corresponding exit of ResNet110 with four exits under the average inference budget of 35 ms/sample. Blue border indicates correct predictions. Orange border indicates the incorrectly predicted samples but correctly predicted by the base model. No border indicates the samples incorrectly predicted both at that exit and by the base model. 10% of the sample are visualized.



(a) Exit 1

(b) Exit 2

(c) Exit 3



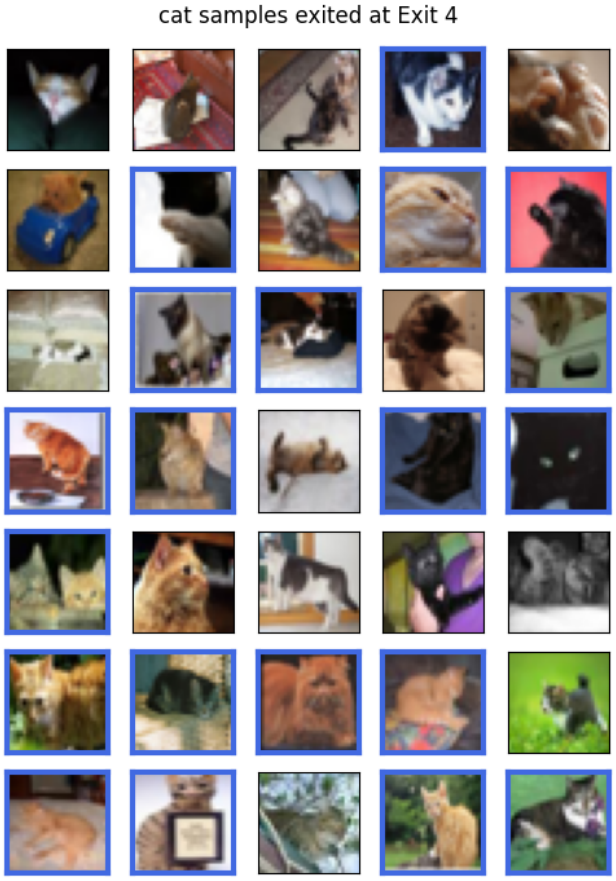
(d) Exit 4

Figure 7. Bird samples from CIFAR-10 test set. Each subfigure illustrates the samples at the corresponding exit of ResNet110 with four exits under the average inference budget of 35 ms/sample. Blue border indicates correct predictions. Orange border indicates the incorrectly predicted samples but correctly predicted by the base model. No border indicates the samples incorrectly predicted both at that exit and by the base model. 10% of the sample are visualized.

(a) Exit 1

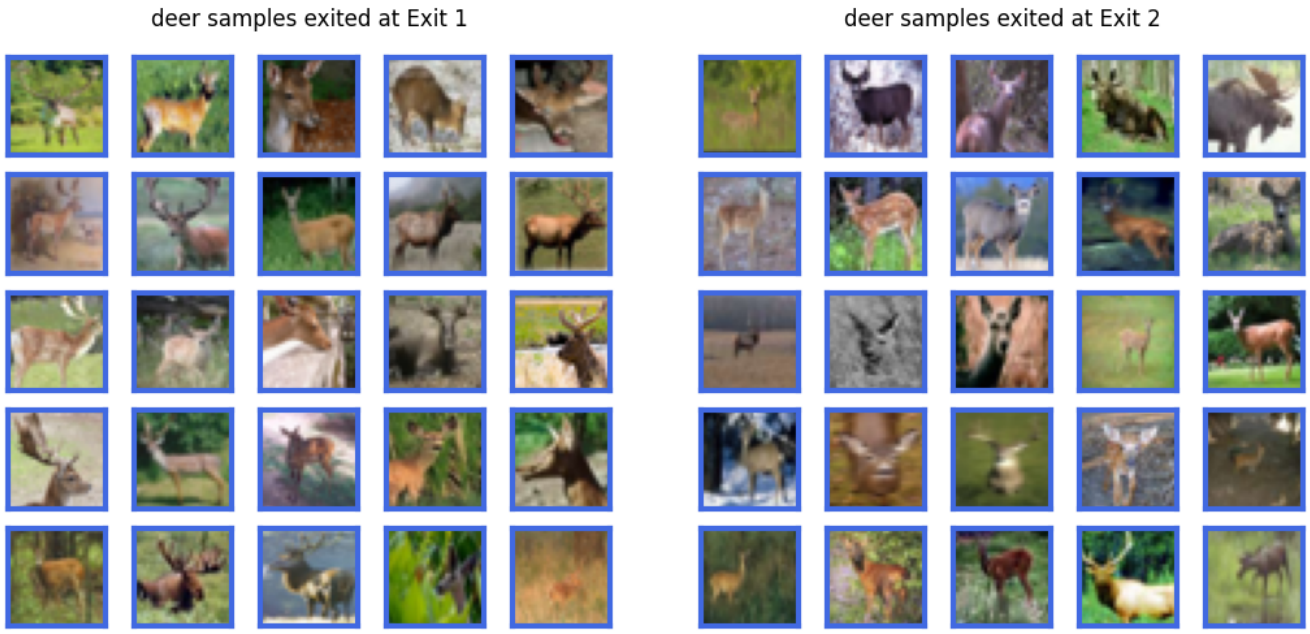
(b) Exit 2

(c) Exit 3



(d) Exit 4

Figure 8. Cat samples from CIFAR-10 test set. Each subfigure illustrates the samples at the corresponding exit of ResNet110 with four exits under the average inference budget of 35 ms/sample. Blue border indicates correct predictions. Orange border indicates the incorrectly predicted samples but correctly predicted by the base model. No border indicates the samples incorrectly predicted both at that exit and by the base model. 10% of the sample are visualized.



(a) Exit 1

(b) Exit 2

(c) Exit 3

(d) Exit 4

Figure 9. Deer samples from CIFAR-10 test set. Each subfigure illustrates the samples at the corresponding exit of ResNet110 with four exits under the average inference budget of 35 ms/sample. Blue border indicates correct predictions. Orange border indicates the incorrectly predicted samples but correctly predicted by the base model. No border indicates the samples incorrectly predicted both at that exit and by the base model. 10% of the sample are visualized.