University of St Andrews

PipeLearn: Pipeline Parallelism for Collaborative Machine Learning

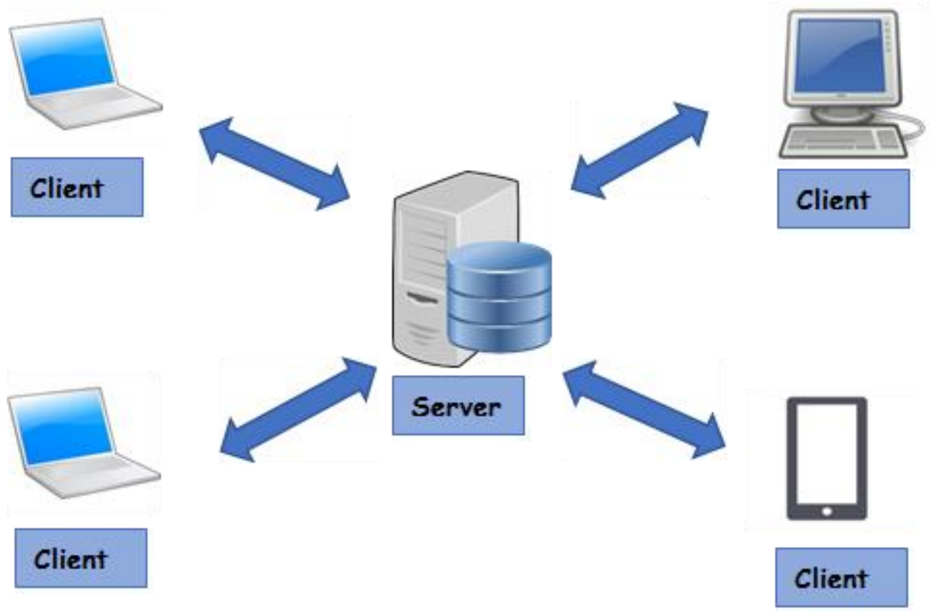
Zihan Zhang

zz66@st-Andrews.ac.uk

Contents

Collaborative Machine Learning

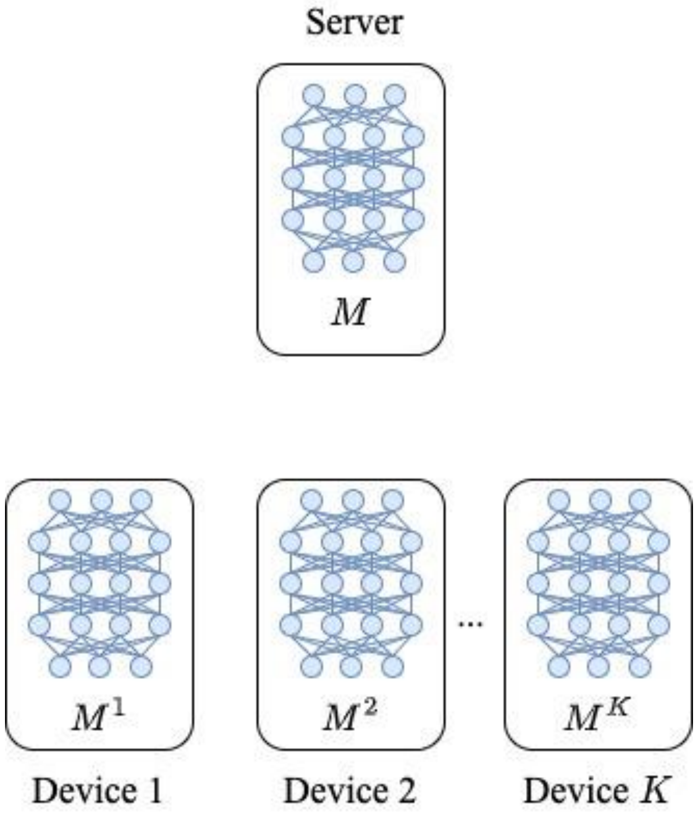
- Collaborative machine learning (CML) techniques were proposed to collaboratively train deep learning models using multiple devices and a server.
- CML techniques preserve the privacy of end-users as it does not require user data to be transferred to the server.



Three popular techniques:

- Federated Learning
- Split Learning
- Split Federated Learning

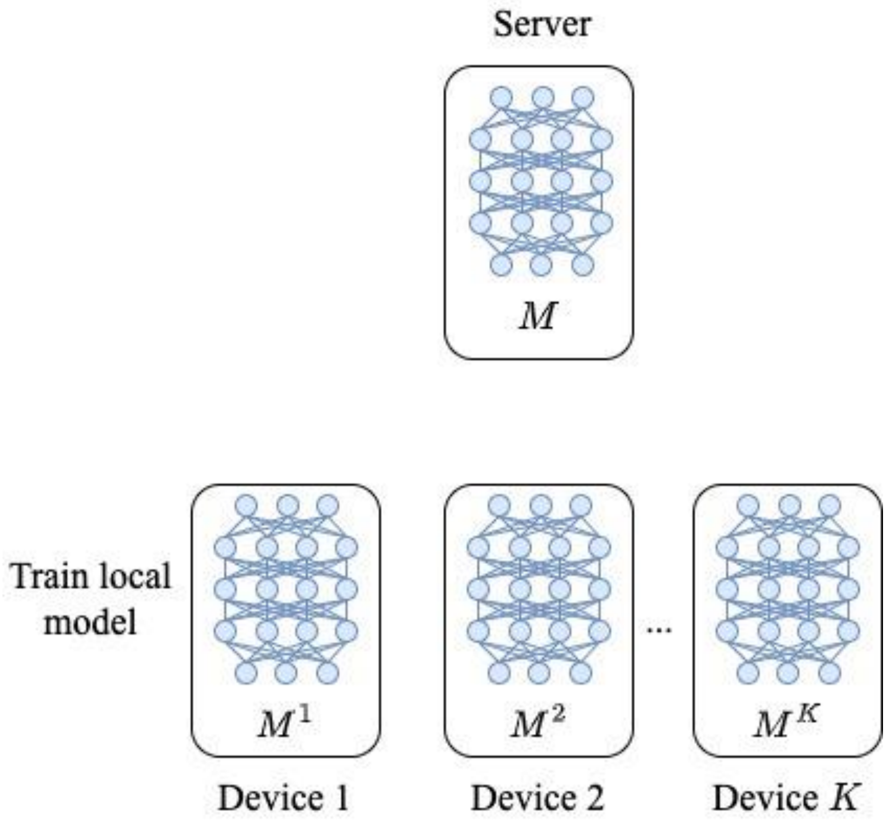
Federated Learning (FL)



- Problem:

The server resources are only employed when the local models are aggregated and remains idle for the remaining time.

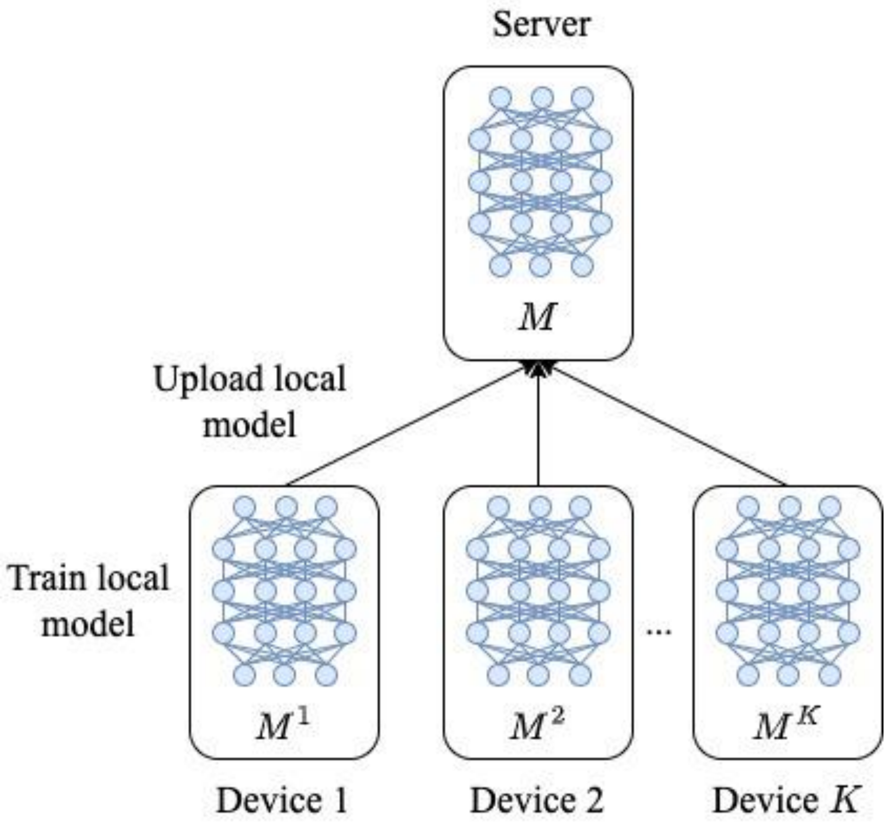
Federated Learning (FL)



- Problem:

The server resources are only employed when the local models are aggregated and remains idle for the remaining time.

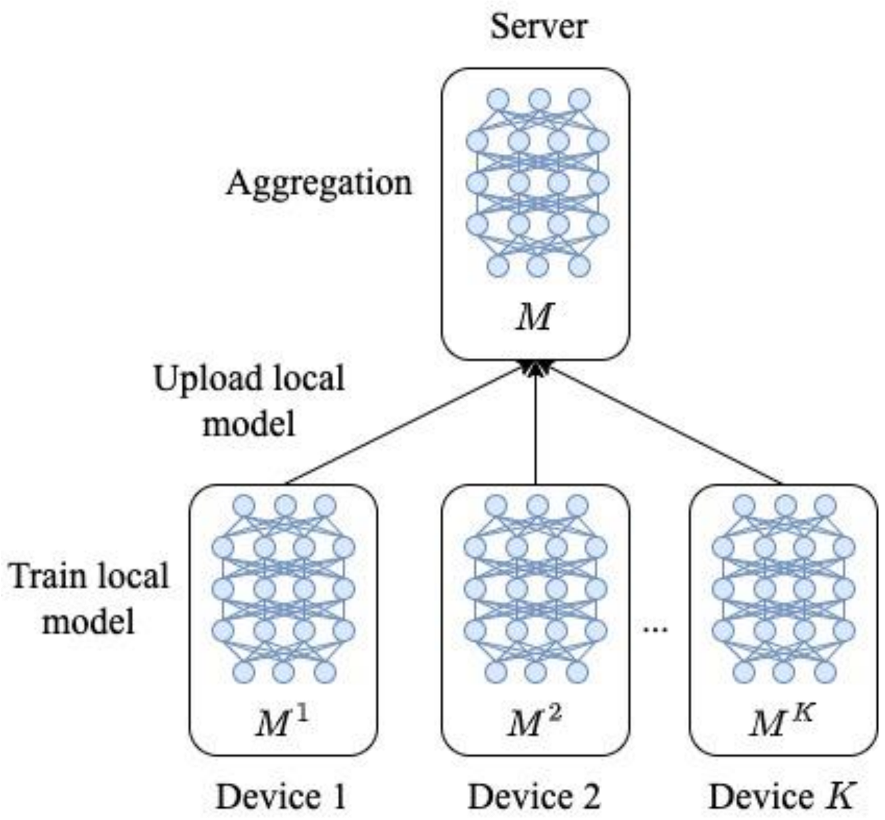
Federated Learning (FL)



- Problem:

The server resources are only employed when the local models are aggregated and remains idle for the remaining time.

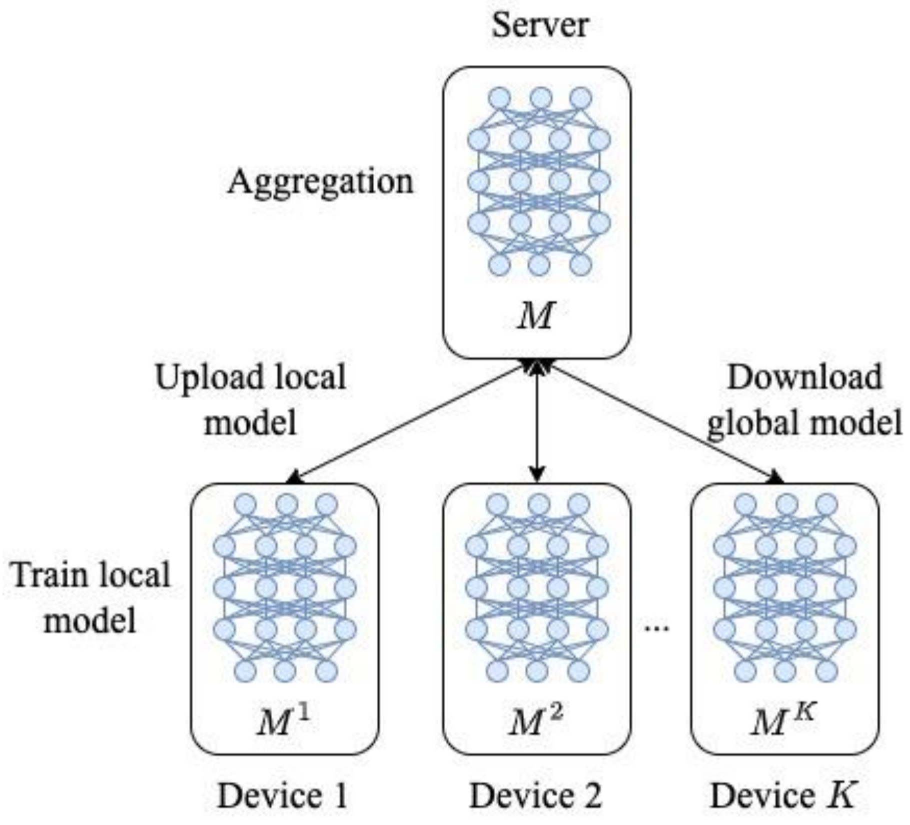
Federated Learning (FL)



- Problem:

The server resources are only employed when the local models are aggregated and remains idle for the remaining time.

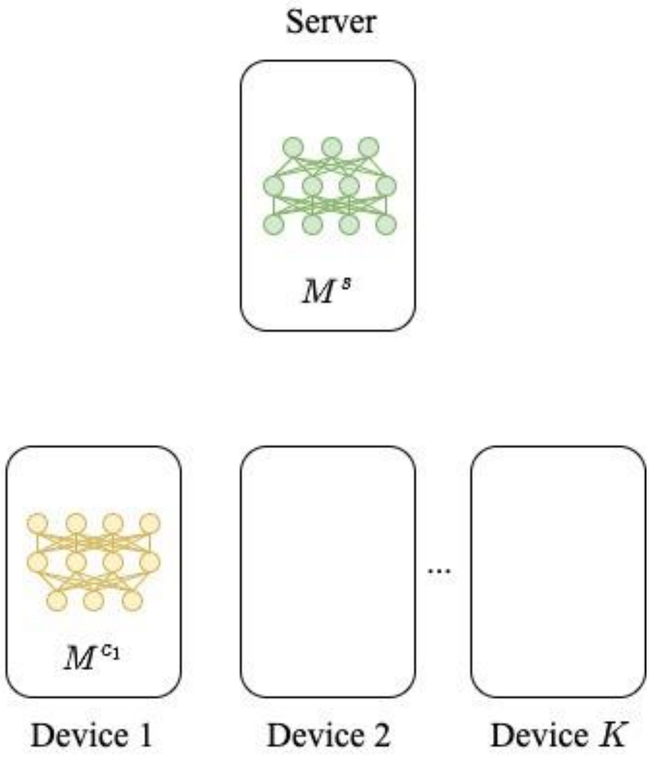
Federated Learning (FL)



- Problem:

The server resources are only employed when the local models are aggregated and remains idle for the remaining time.

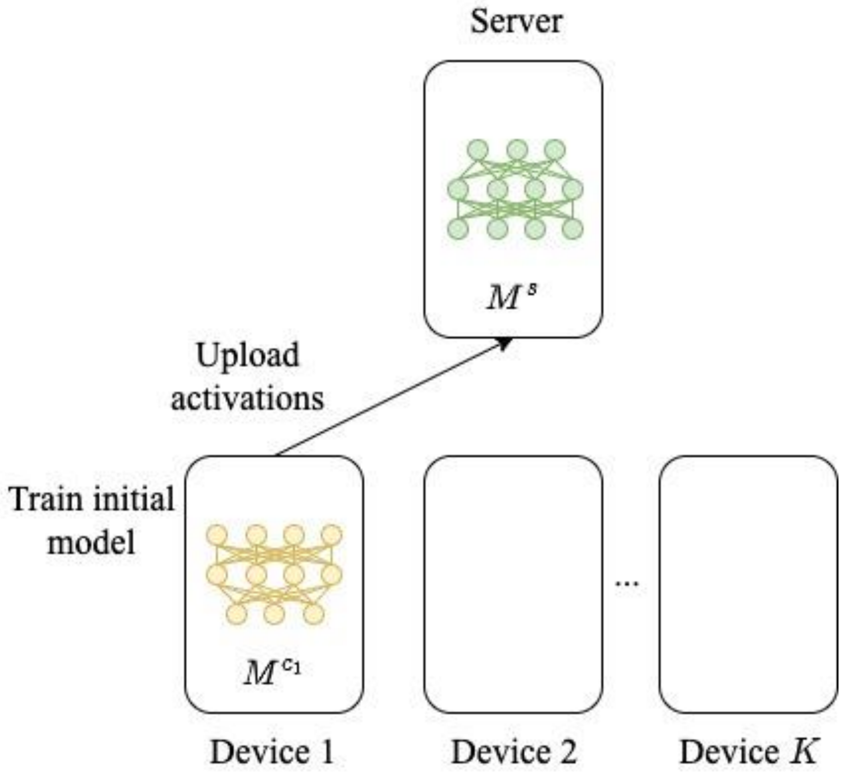
Split Learning (SL)



- Problem:

Only one device or the server will utilise its resources while the other devices or server are idle.

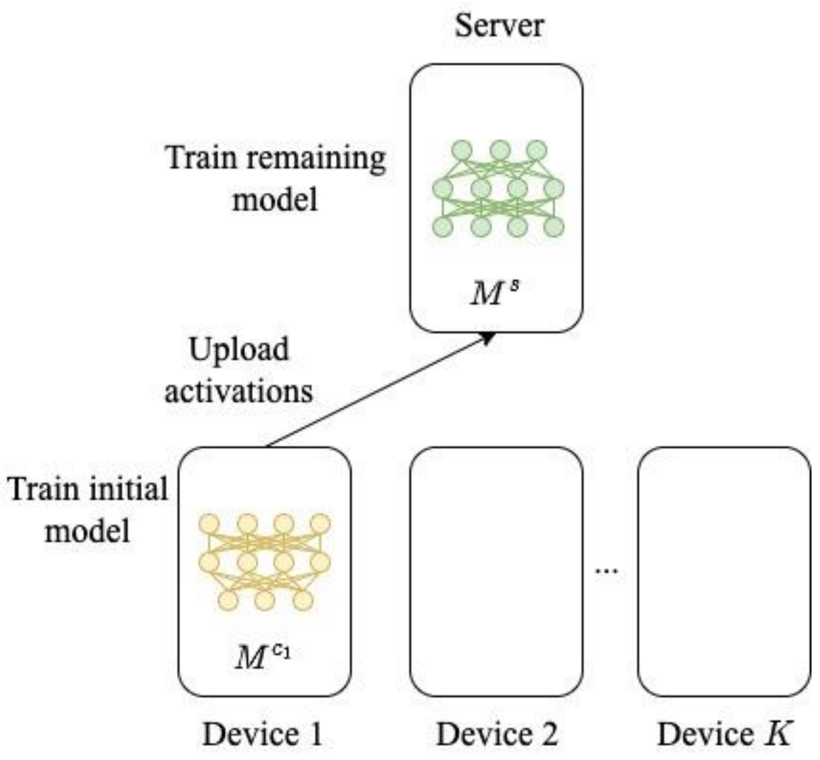
Split Learning (SL)



- Problem:

Only one device or the server will utilise its resources while the other devices or server are idle.

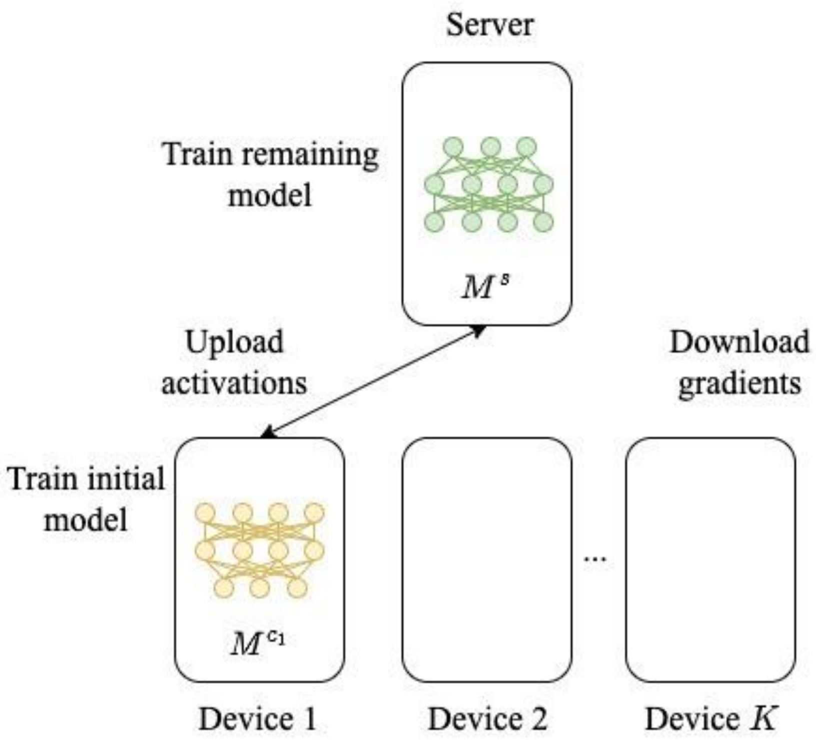
Split Learning (SL)



- Problem:

Only one device or the server will utilise its resources while the other devices or server are idle.

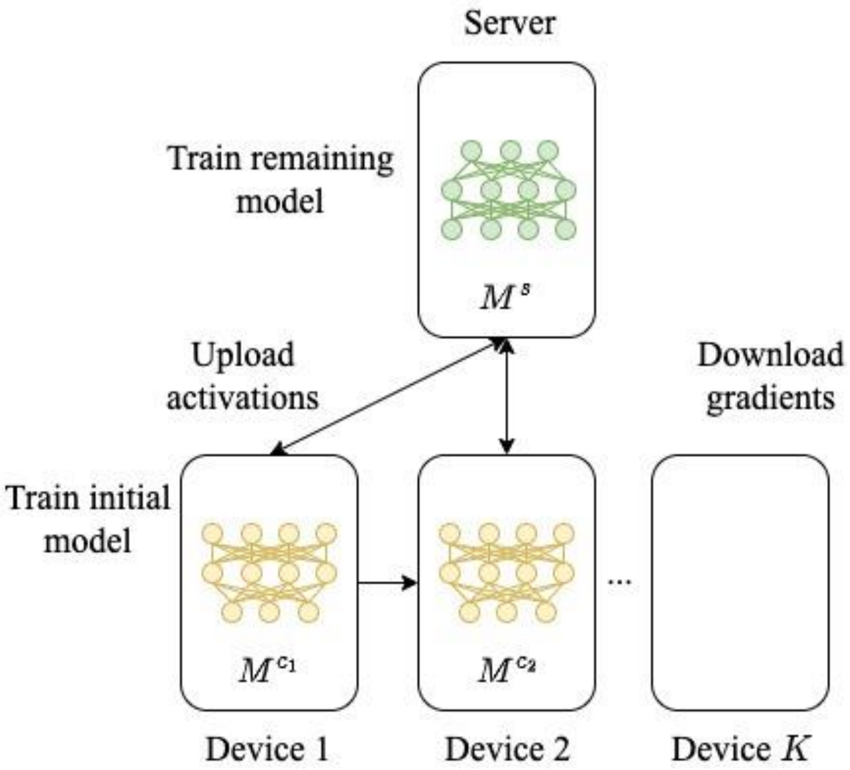
Split Learning (SL)



- Problem:

Only one device or the server will utilise its resources while the other devices or server are idle.

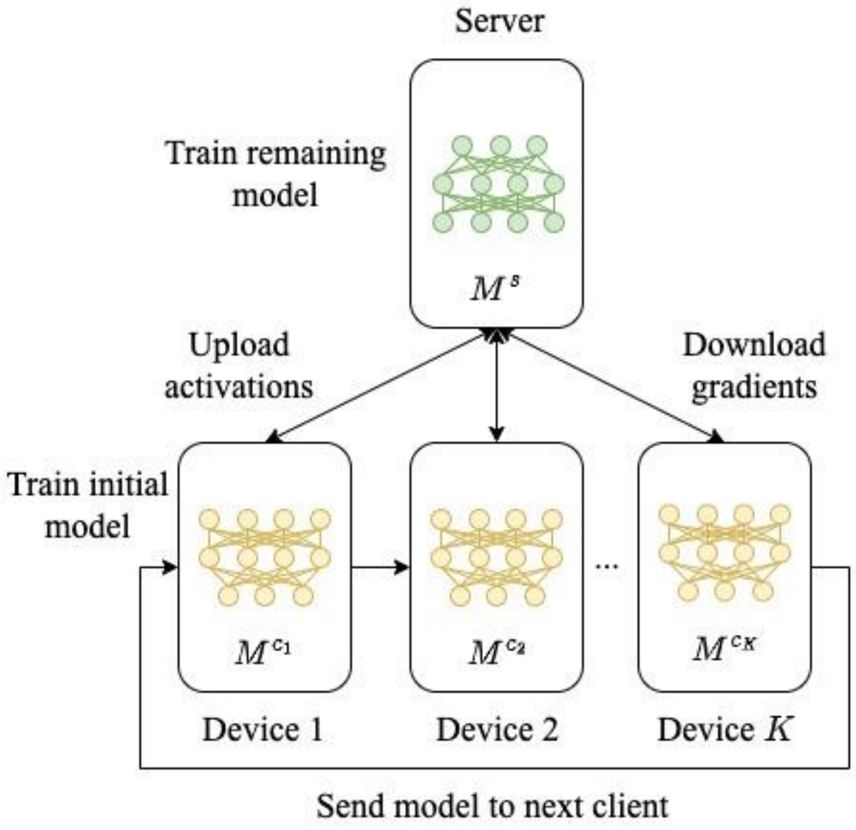
Split Learning (SL)



- Problem:

Only one device or the server will utilise its resources while the other devices or server are idle.

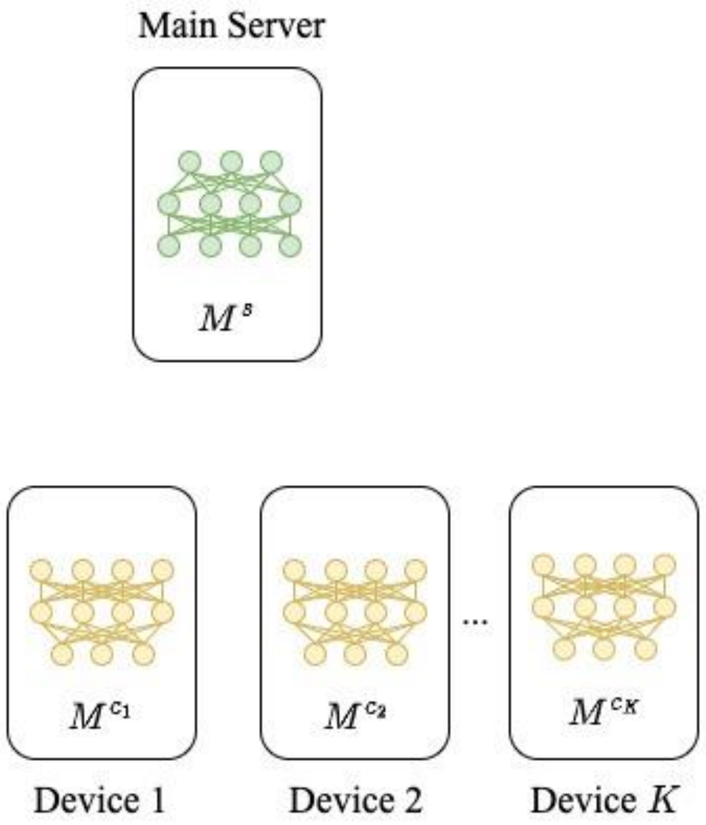
Split Learning (SL)



- Problem:

Only one device or the server will utilise its resources while the other devices or server are idle.

Split Federated Learning (SFL)

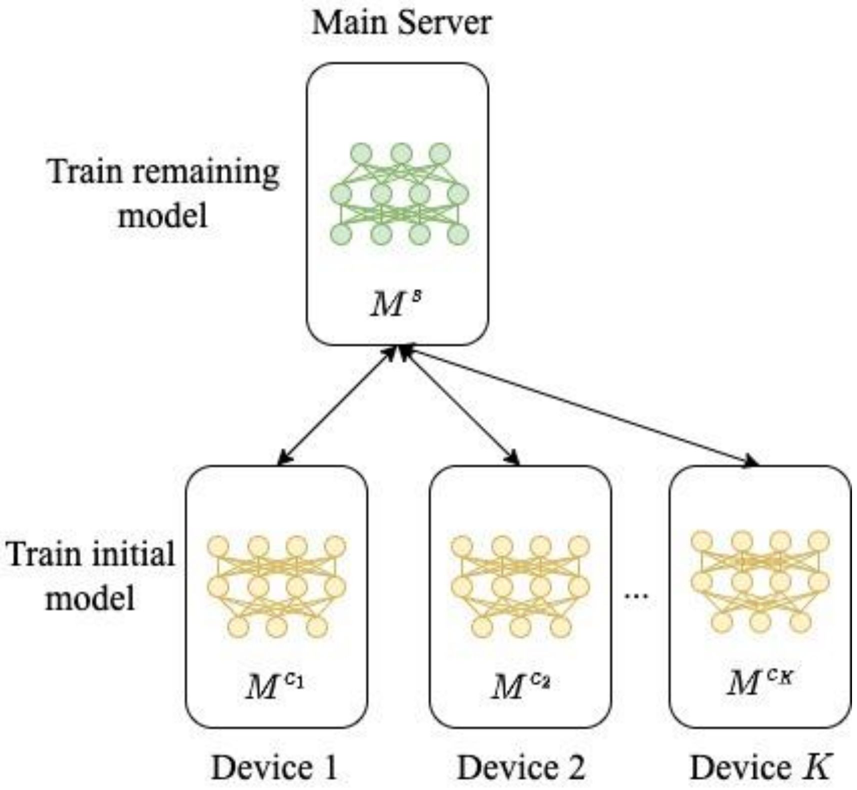


A hybrid of FL and SL.

- Problem:

The server is required to wait while the devices train the model and transfer data, and vice versa.

Split Federated Learning (SFL)

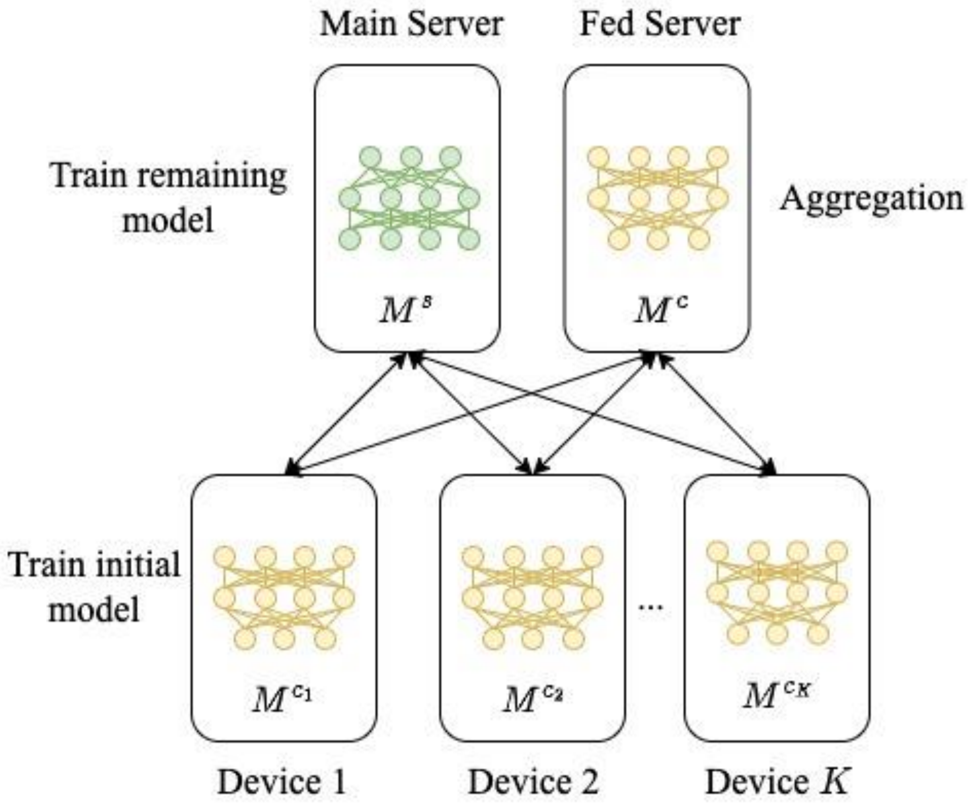


A hybrid of FL and SL.

- Problem:

The server is required to wait while the devices train the model and transfer data, and vice versa.

Split Federated Learning (SFL)



A hybrid of FL and SL.

- Problem:

The server is required to wait while the devices train the model and transfer data, and vice versa.

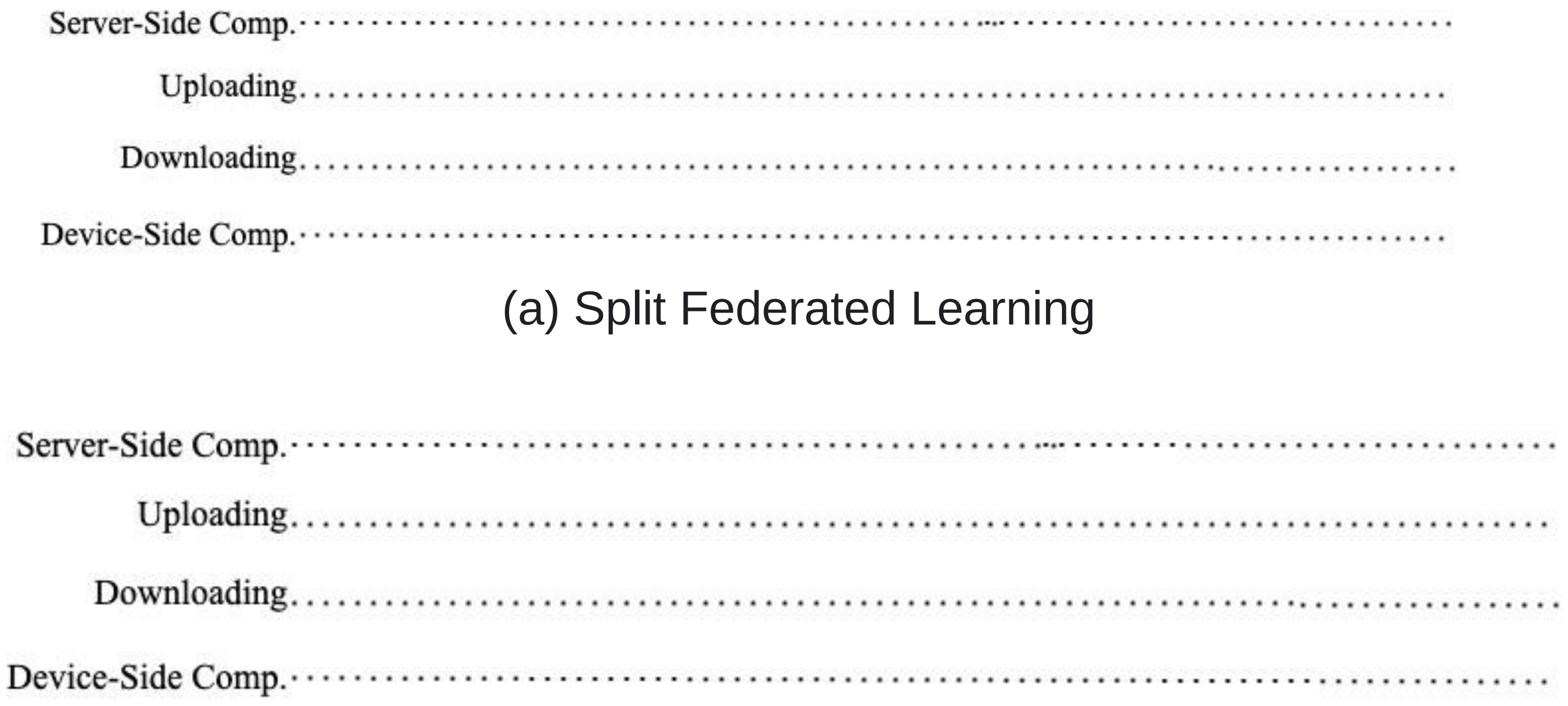
Resources Under-Utilisation Challenges

1. Device and server computations in CML occur in sequence - causes long idle times on both side waiting for the other.

2. Data transfer in CML techniques is time consuming - no training occurs during this time.

PipeLearn

- Split the model across server and devices.
- Split each mini-batch of data to several micro-batches.
- Parallelise device-side computation, server-side computation and communication.
- All devices are training in parallel.



(a) Split Federated Learning

(b) PipeLearn

Figure 1. A training iteration for split federated learning and PipeLearn, where f, b, u and d represent forward pass, backward pass, upload and download, respectively.

PipeLearn

- Split the model across server and devices.
- Split each mini-batch of data to several micro-batches.
- Parallelise device-side computation, server-side computation and communication.
- All devices are training in parallel.

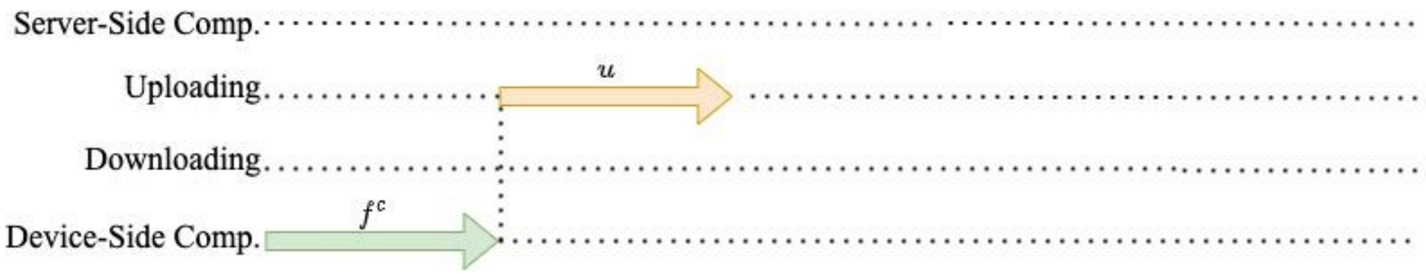
(a) Split Federated Learning

(b) PipeLearn

Figure 1. A training iteration for split federated learning and PipeLearn, where f, b, u and d represent forward pass, backward pass, upload and download, respectively.

PipeLearn

- Split the model across server and devices.
- Split each mini-batch of data to several micro-batches.
- Parallelise device-side computation, server-side computation and communication.
- All devices are training in parallel.



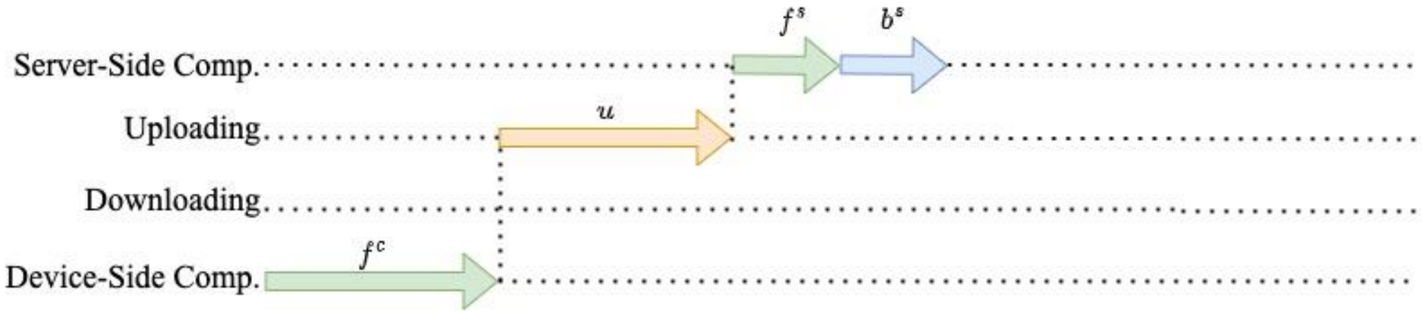
(a) Split Federated Learning

(b) PipeLearn

Figure 1. A training iteration for split federated learning and PipeLearn, where f, b, u and d represent forward pass, backward pass, upload and download, respectively.

PipeLearn

- Split the model across server and devices.
- Split each mini-batch of data to several micro-batches.
- Parallelise device-side computation, server-side computation and communication.
- All devices are training in parallel.



(a) Split Federated Learning

(b) PipeLearn

Figure 1. A training iteration for split federated learning and PipeLearn, where f, b, u and d represent forward pass, backward pass, upload and download, respectively.

PipeLearn

- Split the model across server and devices.
- Split each mini-batch of data to several micro-batches.
- Parallelise device-side computation, server-side computation and communication.
- All devices are training in parallel.

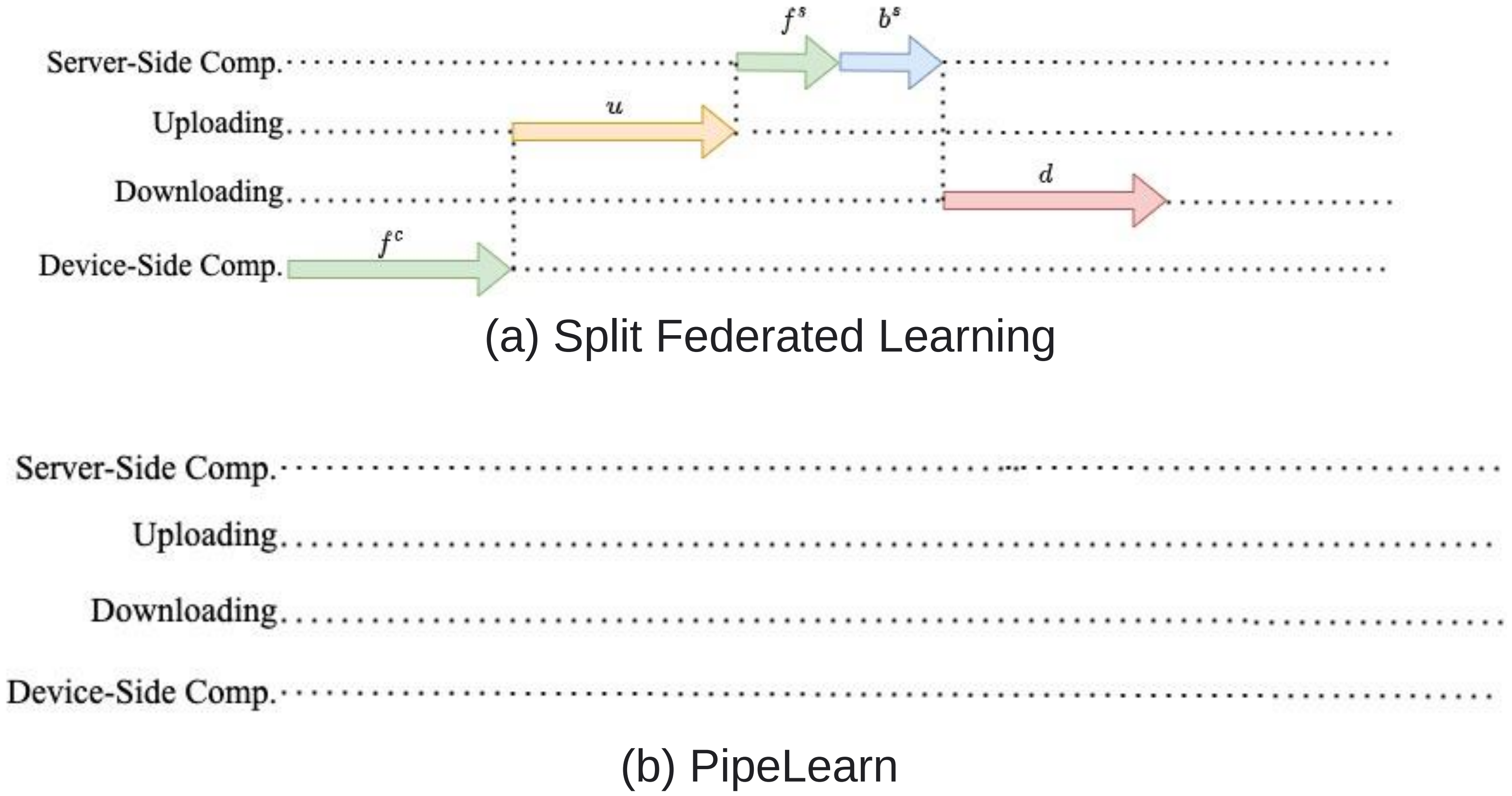
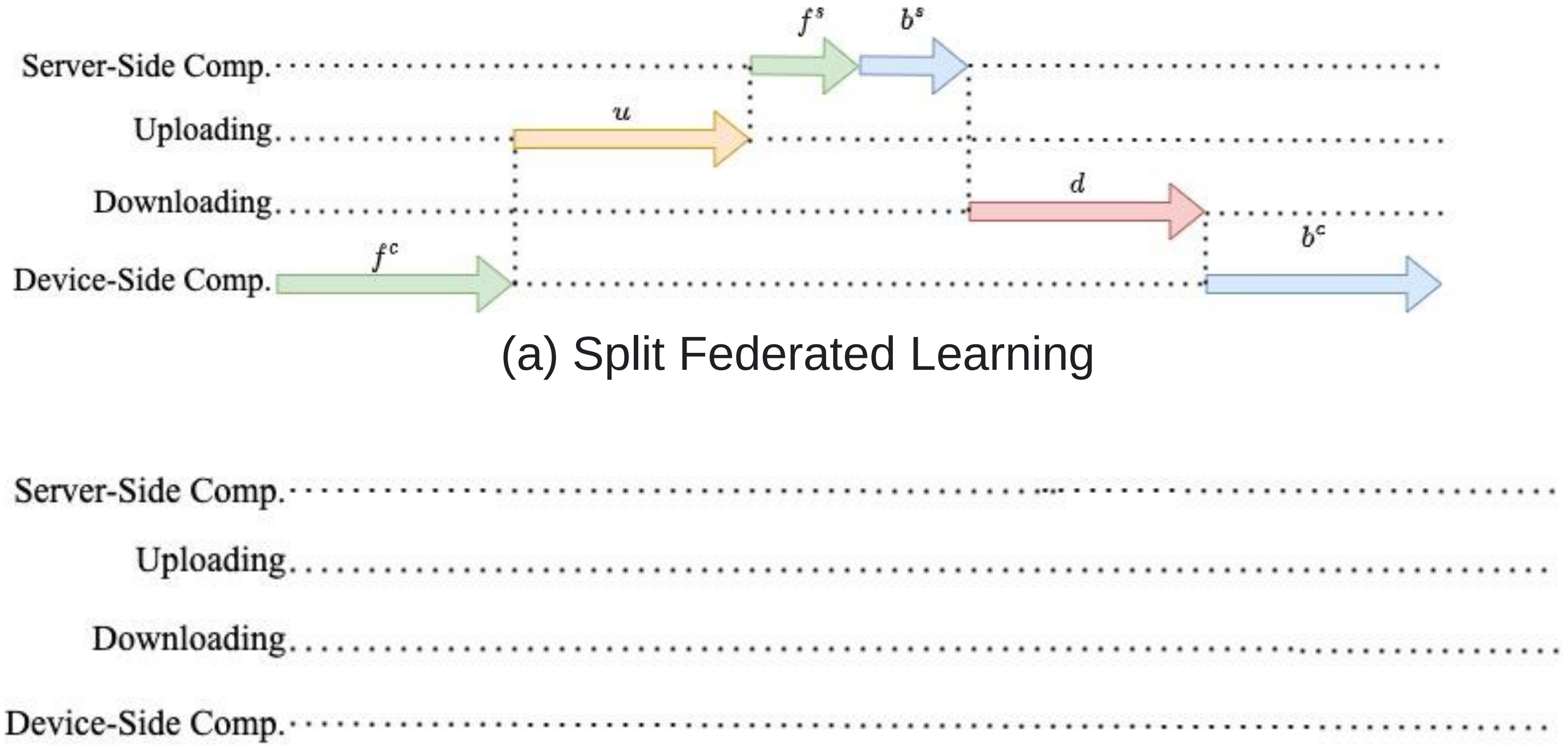


Figure 1. A training iteration for split federated learning and PipeLearn, where f, b, u and d represent forward pass, backward pass, upload and download, respectively.

PipeLearn

- Split the model across server and devices.
- Split each mini-batch of data to several micro-batches.
- Parallelise device-side computation, server-side computation and communication.
- All devices are training in parallel.

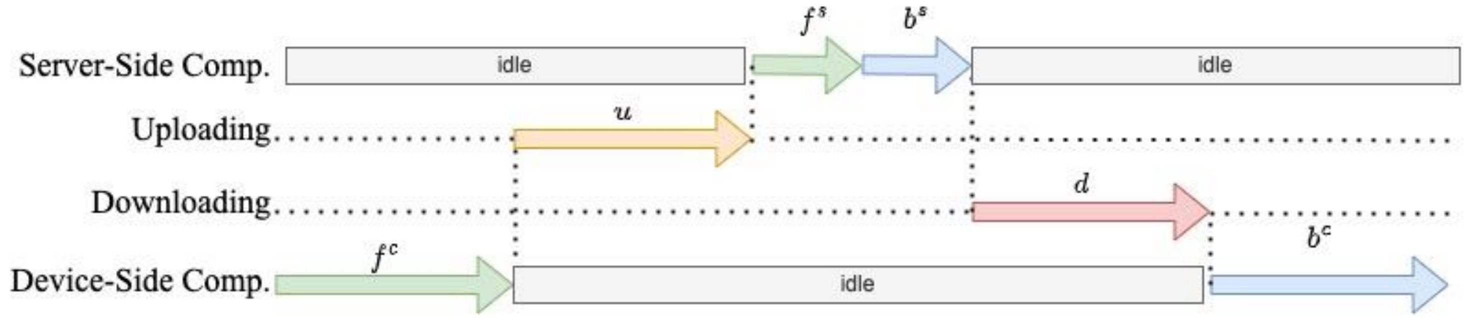


(b) PipeLearn

Figure 1. A training iteration for split federated learning and PipeLearn, where f, b, u and d represent forward pass, backward pass, upload and download, respectively.

PipeLearn

- Split the model across server and devices.
- Split each mini-batch of data to several micro-batches.
- Parallelise device-side computation, server-side computation and communication.
- All devices are training in parallel.



(a) Split Federated Learning

(b) PipeLearn

Figure 1. A training iteration for split federated learning and PipeLearn, where f, b, u and d represent forward pass, backward pass, upload and download, respectively.

PipeLearn

- Split the model across server and devices.
- Split each mini-batch of data to several micro-batches.
- Parallelise device-side computation, server-side computation and communication.
- All devices are training in parallel.

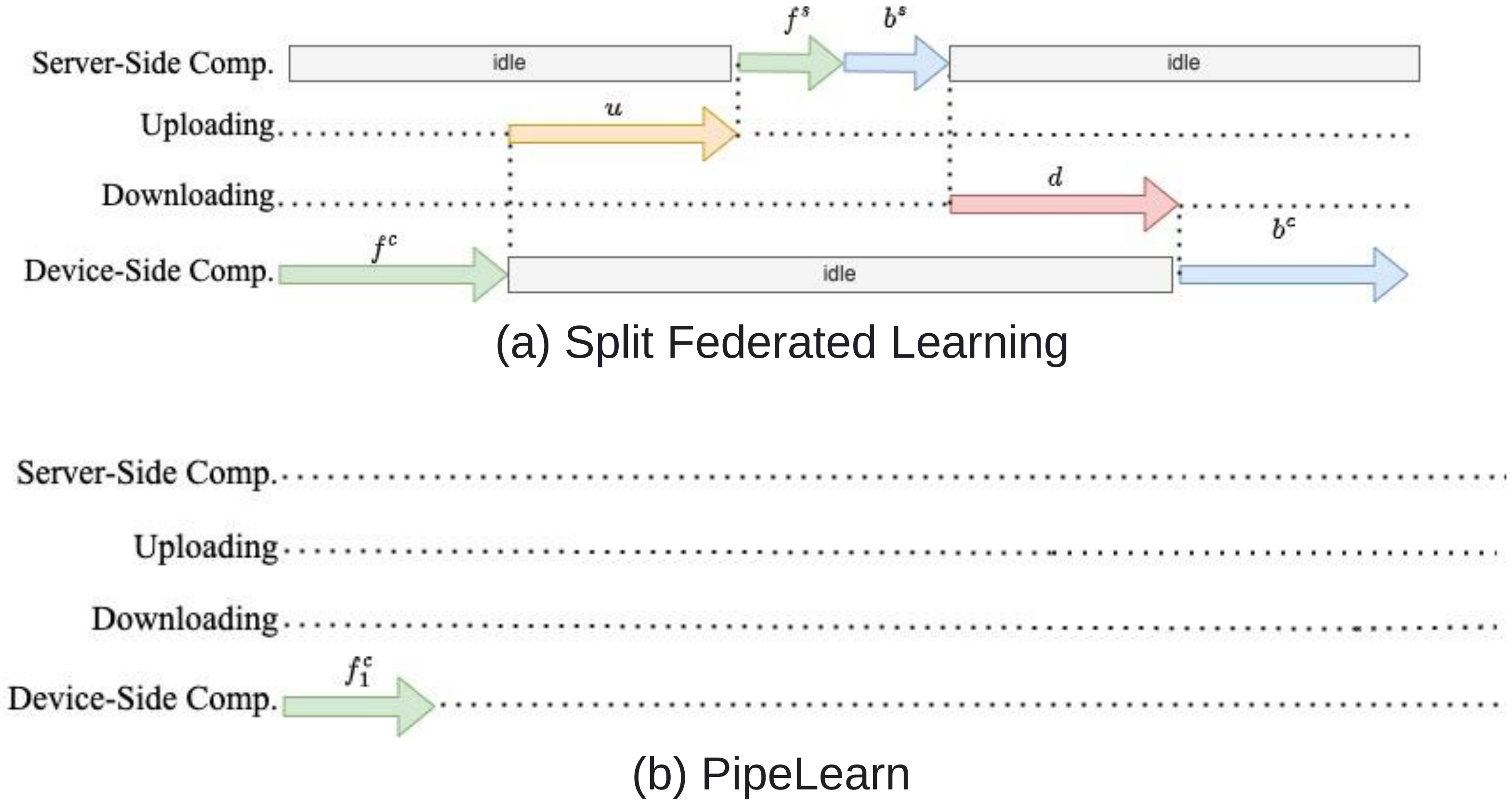


Figure 1. A training iteration for split federated learning and PipeLearn, where f, b, u and d represent forward pass, backward pass, upload and download, respectively.

PipeLearn

- Split the model across server and devices.
- Split each mini-batch of data to several micro-batches.
- Parallelise device-side computation, server-side computation and communication.
- All devices are training in parallel.

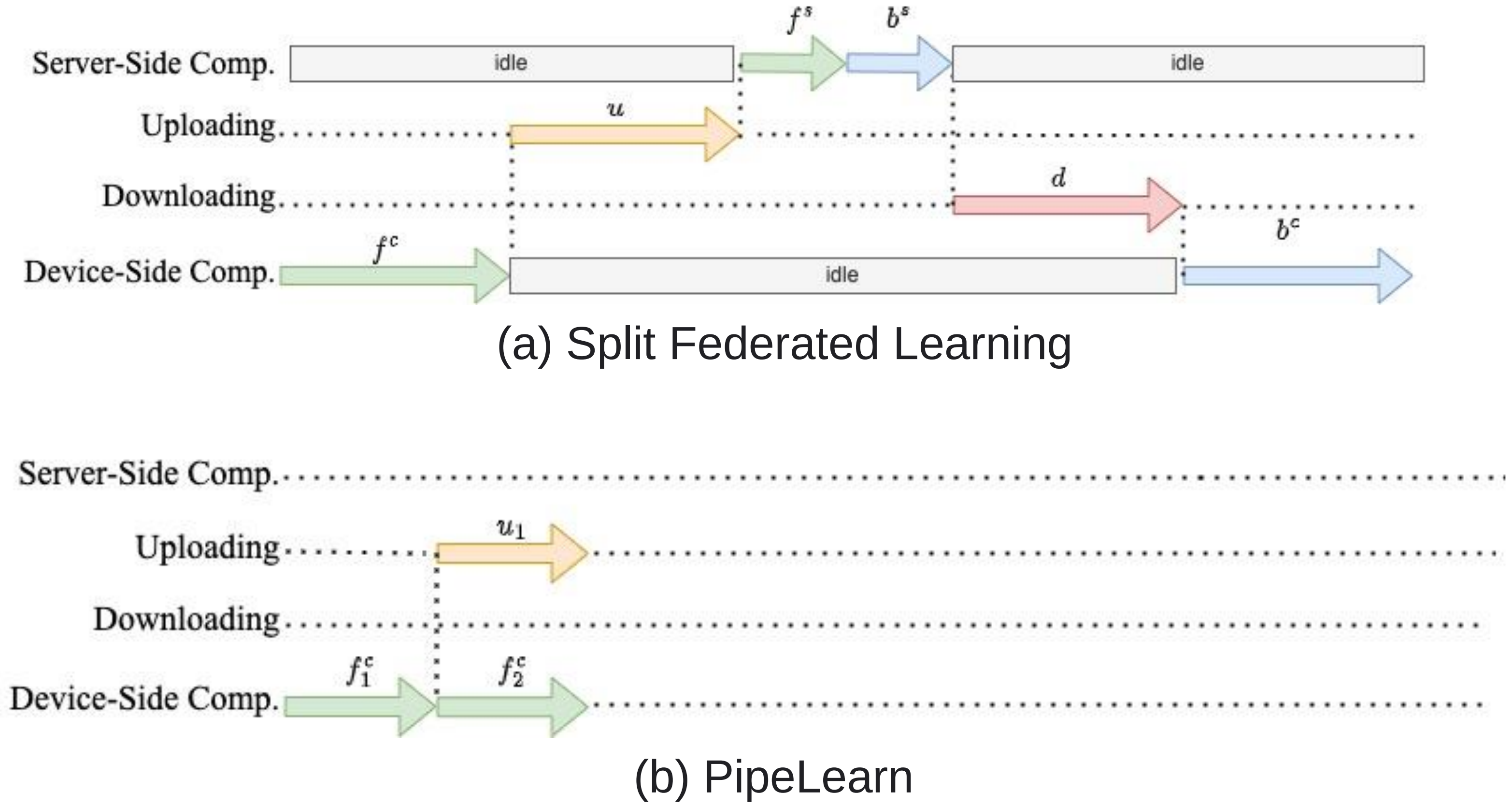
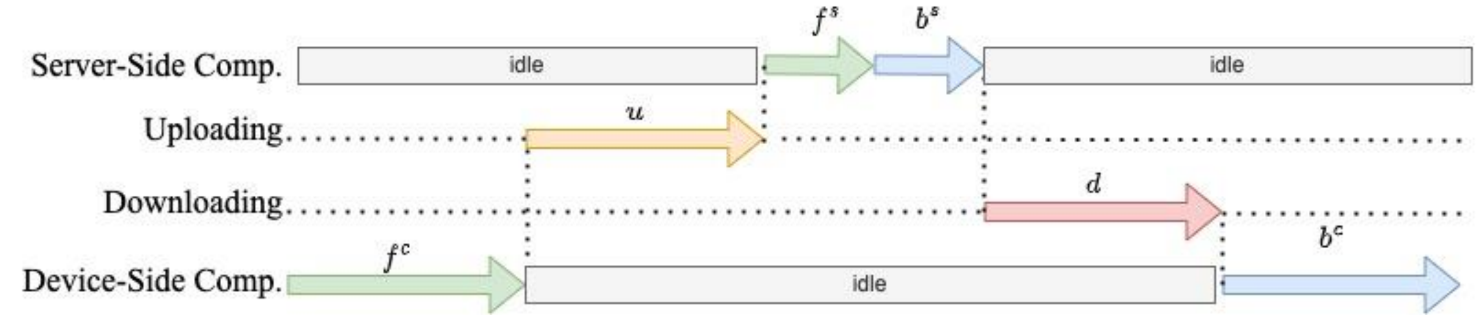


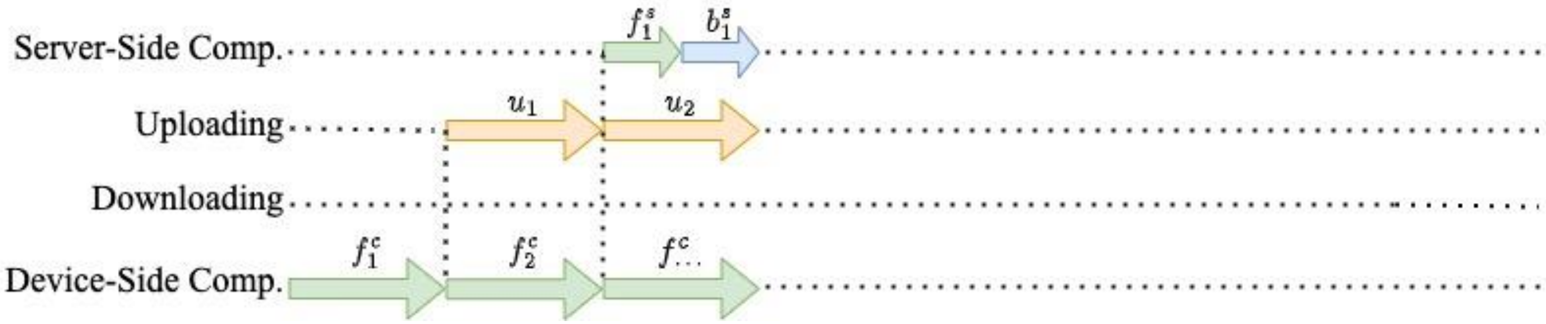
Figure 1. A training iteration for split federated learning and PipeLearn, where f, b, u and d represent forward pass, backward pass, upload and download, respectively.

PipeLearn

- Split the model across server and devices.
- Split each mini-batch of data to several micro-batches.
- Parallelise device-side computation, server-side computation and communication.
- All devices are training in parallel.



(a) Split Federated Learning

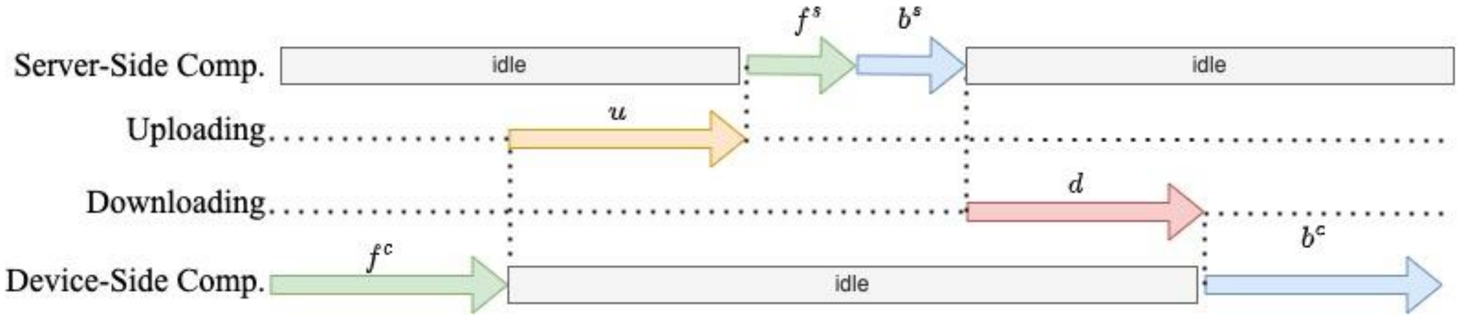


(b) PipeLearn

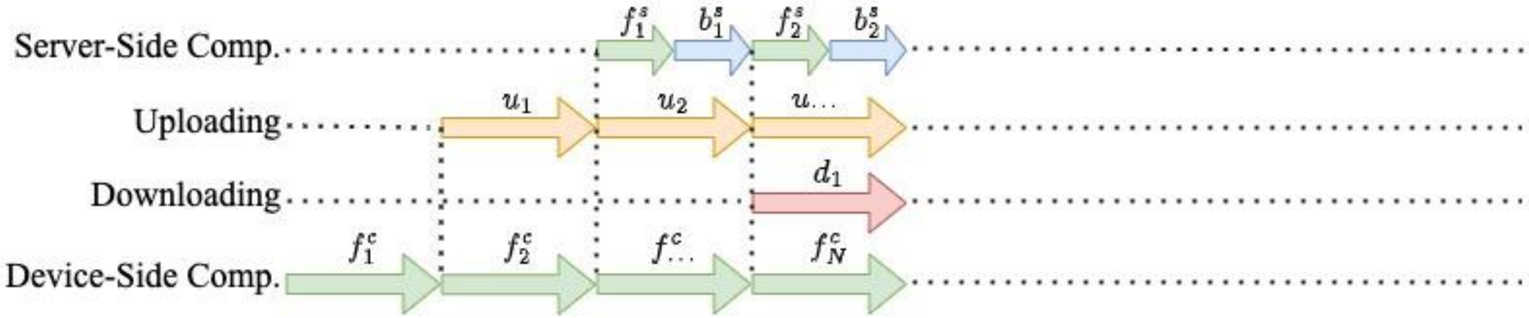
Figure 1. A training iteration for split federated learning and PipeLearn, where f, b, u and d represent forward pass, backward pass, upload and download, respectively.

PipeLearn

- Split the model across server and devices.
- Split each mini-batch of data to several micro-batches.
- Parallelise device-side computation, server-side computation and communication.
- All devices are training in parallel.



(a) Split Federated Learning

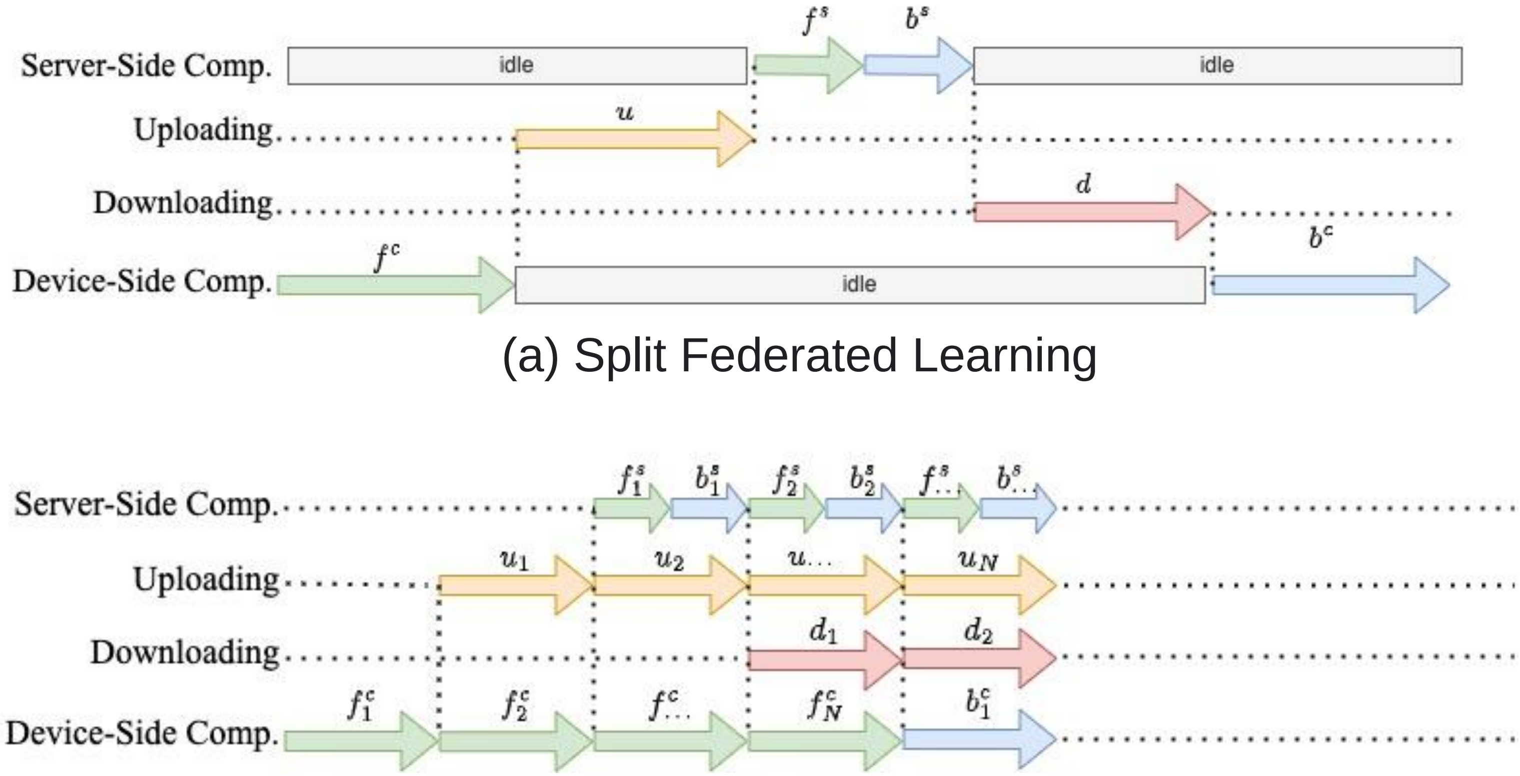


(b) PipeLearn

Figure 1. A training iteration for split federated learning and PipeLearn, where f, b, u and d represent forward pass, backward pass, upload and download, respectively.

PipeLearn

- Split the model across server and devices.
- Split each mini-batch of data to several micro-batches.
- Parallelise device-side computation, server-side computation and communication.
- All devices are training in parallel.



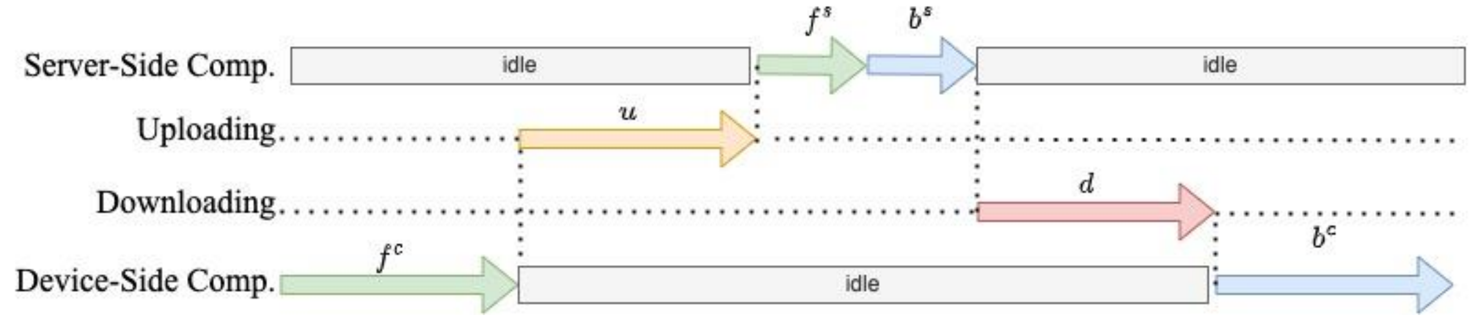
(a) Split Federated Learning

(b) PipeLearn

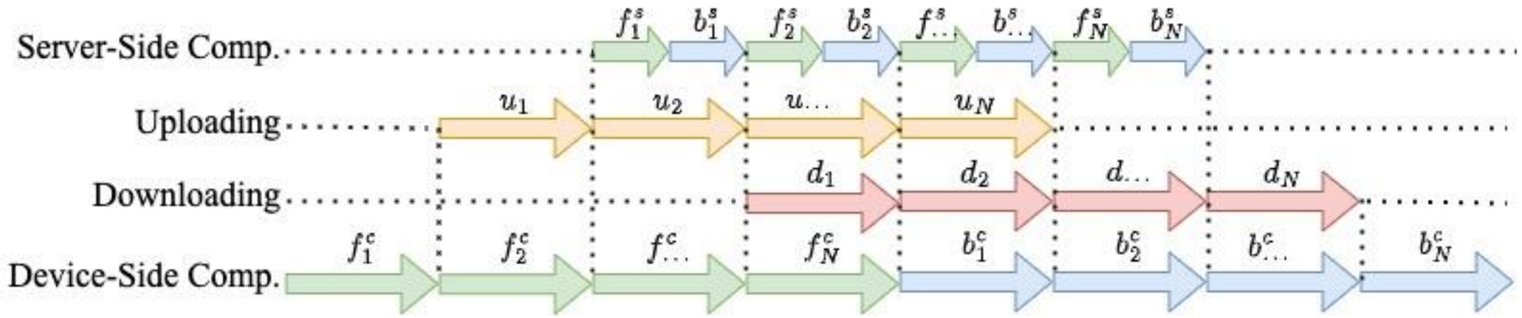
Figure 1. A training iteration for split federated learning and PipeLearn, where f, b, u and d represent forward pass, backward pass, upload and download, respectively.

PipeLearn

- Split the model across server and devices.
- Split each mini-batch of data to several micro-batches.
- Parallelise device-side computation, server-side computation and communication.
- All devices are training in parallel.



(a) Split Federated Learning

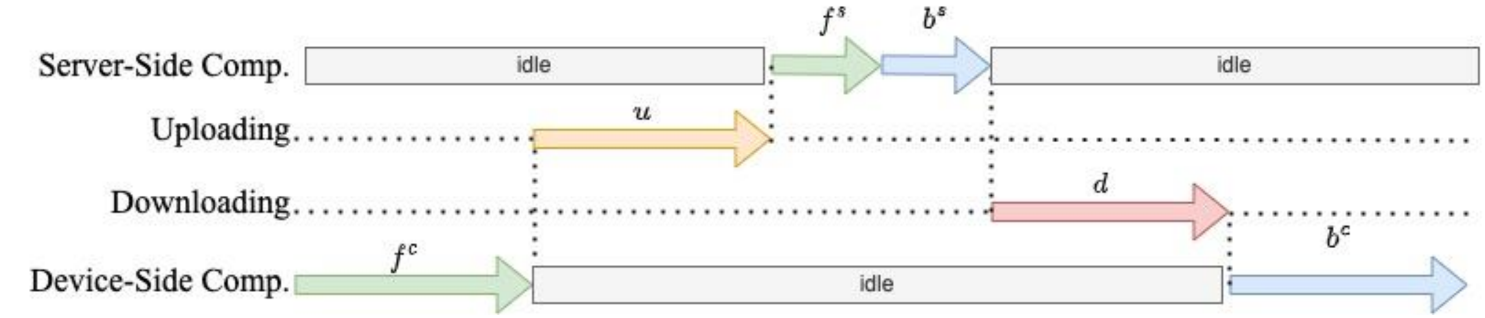


(b) PipeLearn

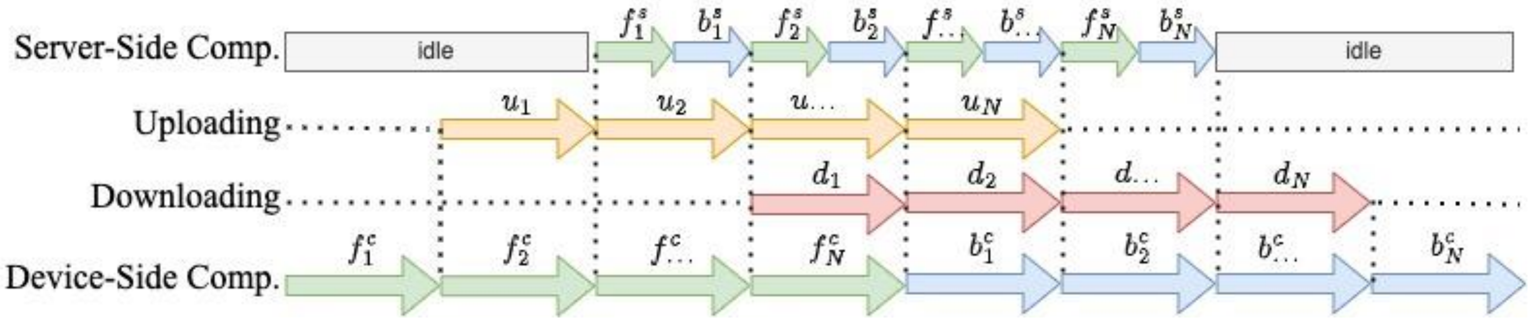
Figure 1. A training iteration for split federated learning and PipeLearn, where f, b, u and d represent forward pass, backward pass, upload and download, respectively.

PipeLearn

- Split the model across server and devices.
- Split each mini-batch of data to several micro-batches.
- Parallelise device-side computation, server-side computation and communication.
- All devices are training in parallel.



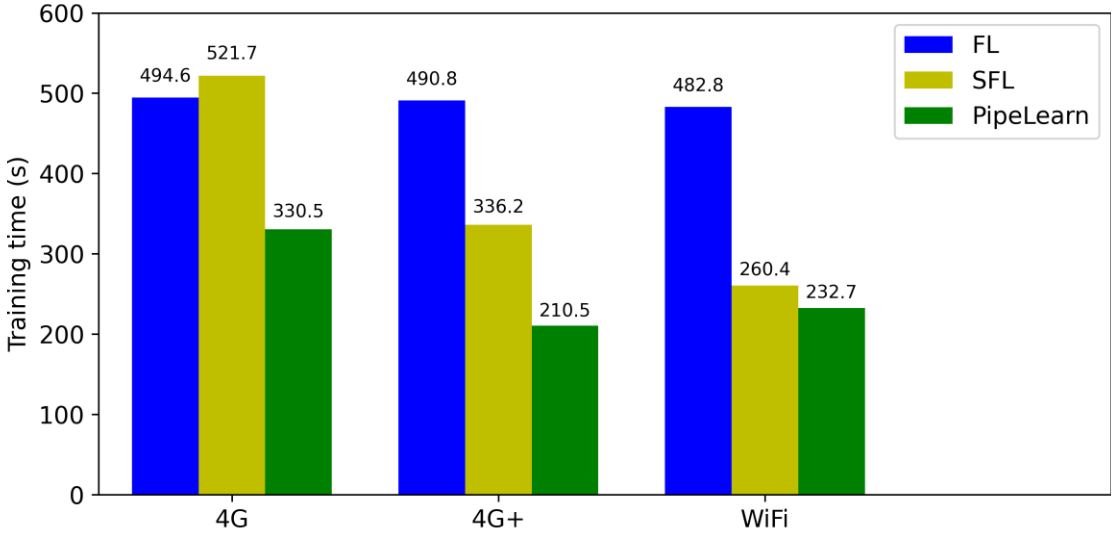
(a) Split Federated Learning



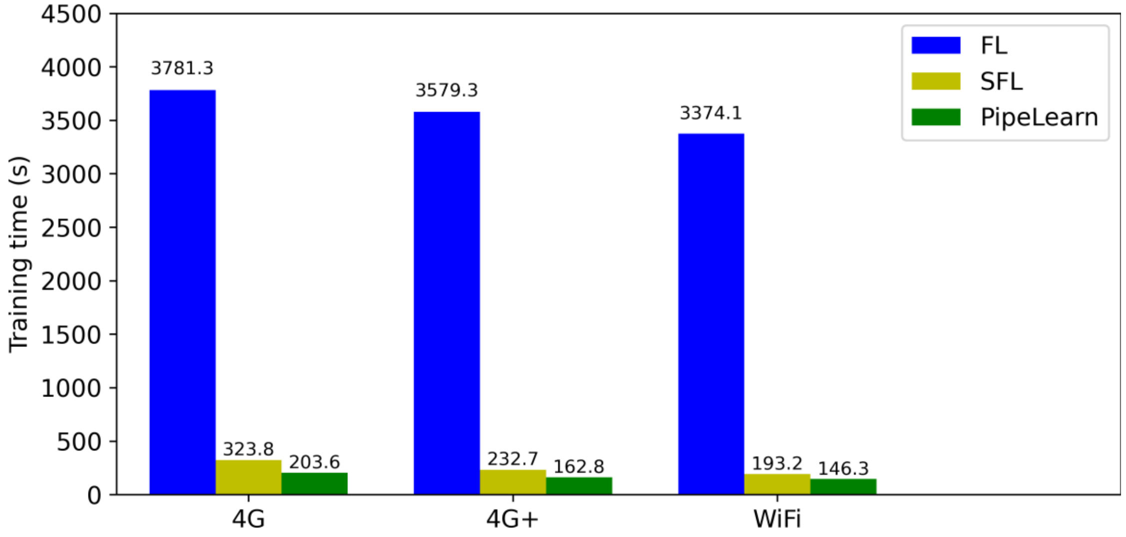
(b) PipeLearn

Figure 1. A training iteration for split federated learning and PipeLearn, where f, b, u and d represent forward pass, backward pass, upload and download, respectively.

Experiment: Training Efficiency



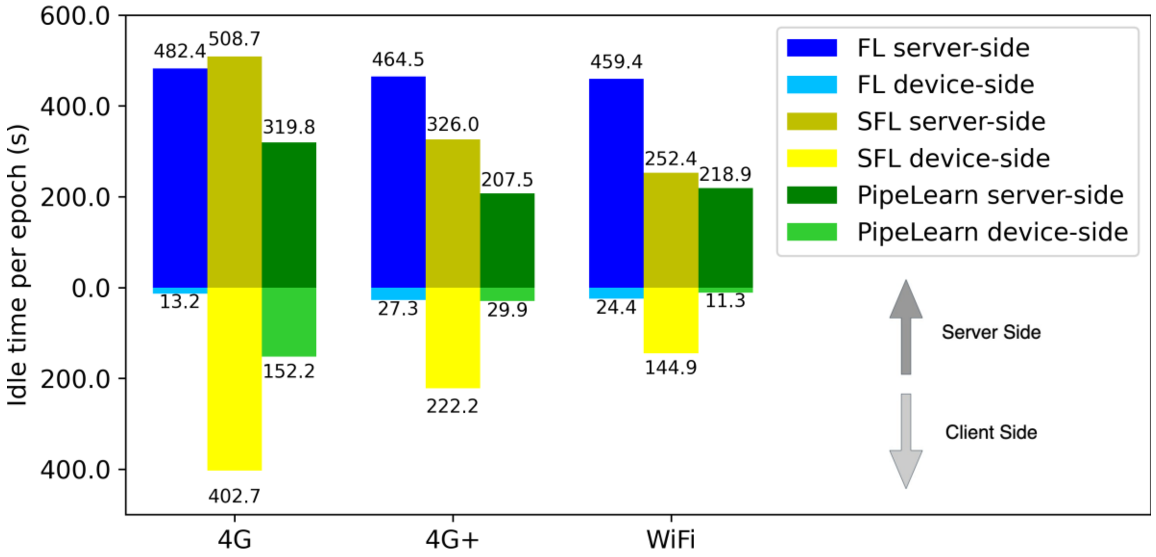
(a) VGG5



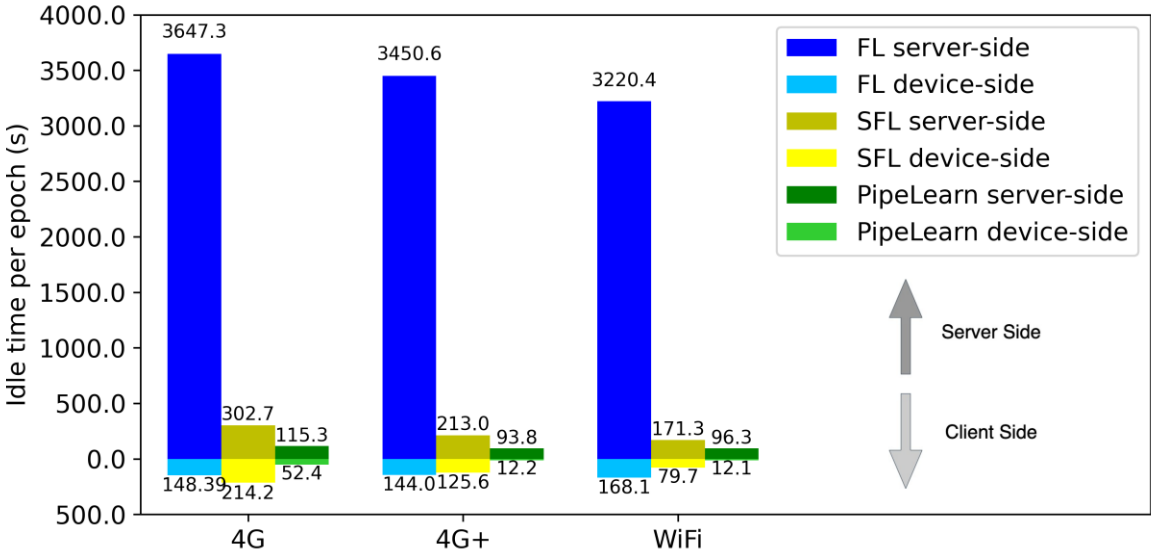
(a) ResNet18

Figure 2. Training time per epoch for FL, SFL and PipeLearn under different network conditions.

Experiment: Idle Time



(a) VGG5



(a) ResNet18

Figure 3. Idle time per epoch on the server and devices in FL, SFL and PipeLearn under different network conditions.

Experiment: Model Accuracy

Model	Technique	Test Accuracy
VGG5	FL	79.95
	SFL	79.55
	PipeLearn under 4G	79.15
	PipeLearn under 4G+	78.4
	PipeLearn under WiFi	78.65
ResNet18	FL	80.4
	SFL	81.55
	PipeLearn under 4G	79.5
	PipeLearn under 4G+	81.35
	PipeLearn under WiFi	80.2

Table 1. Model accuracy of VGG5 and ResNet18 on the test dataset using FL, SFL and PipeLearn, under different network conditions.

Conclusion

Compared to federated learning:

- PipeLearn accelerates the training process by up to 21.6x.
- PipeLearn reduces idle time by up to 28.5x.
- PipeLearn achieves (near) similar model accuracy.

Z. Zhang, P. Rodgers, P. Kilpatrick, I. Spence and B. Varghese, "PipeLearn: Pipeline Parallelism for Collaborative Machine Learning," IEEE Transactions on Parallel and Distributed Systems, 2022 [Under Revision].


University of
St Andrews
www.st-andrews.ac.uk