Multi-Level Branched Regularization for Federated Learning

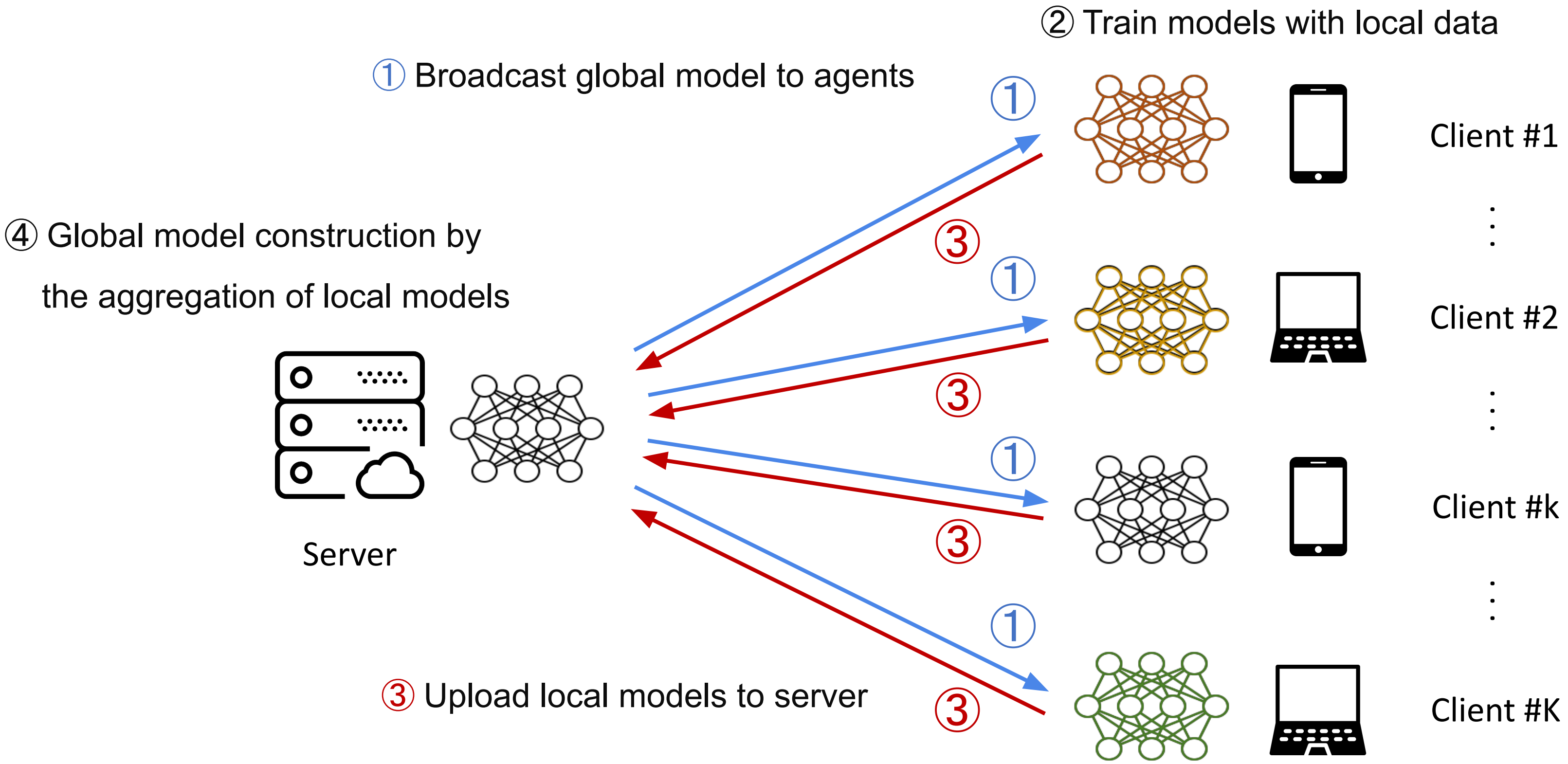
Jinkyu Kim*, Geeho Kim*, Bohyung Han (* equal contribution)

Computer Vision Lab
Seoul National University

ICML 2022, Baltimore, USA

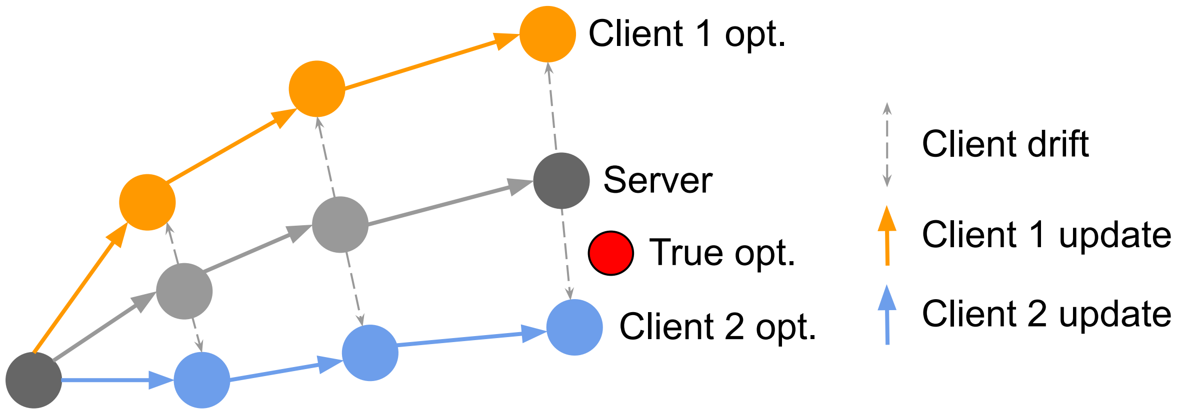
Federated Learning

- Unique distributed learning framework with decentralized data
  - Server learns a shared model through collaboration with a large number of remote clients
  - Achieve **the basic level of privacy** since the server does **not observe** training data directly
  - Each client runs a number of iterations to minimize communication costs with the central server
  - The server constructs a shared model via model averaging



Main Challenge: Data Heterogeneity

- Data distributions of individual clients are **different** from the global distribution
  - **Multiple** local updates on **non-iid** data distributions lead to **client-drift**
  - Individual client updates are prone to **diverge** and **inconsistent**
  - **Overfit** on local skewed data
- This challenge is exacerbated when the client participation rate per round is **low**
  - Unstable client device operations and limited communication channels
  - Hampers the convergence to the **optimal average loss** over all clients



[Karimireddy20] S. P. Karimireddy, et al.: **SCAFFOLD: stochastic controlled averaging for on-device federated learning.** ICML 2020

Existing Frameworks

- Regularize local model updates to prevent a large deviation from the global model
  - Variance reduction techniques
  - Dynamic regularization based on local gradient
  - Ensure the similarity of the representations between the global model and local networks

☹ Need additional communication cost per round

☹ Extra memory requirements in clients to store local historical states

Existing Frameworks

- Knowledge distillation based methods
  - Use the global model as a teacher
  - Matches the representations of the local model to those of the teacher at the logit level

☺ Utilize global knowledge on whole data distribution

☹ The parameters in the lower layers are less affected

☹ Merely simulating the fixed output of the global model is sub-optimal

Existing Frameworks

- Knowledge distillation based methods
  - Use the global model as a teacher of local model
  - Matches the representations at the logit level

☺ Utilize global knowledge on whole data distribution

☹ The parameters in the lower layers are less affected

☹ Merely simulating the fixed output of the global model is sub-optimal

- Layer-wise KD techniques
  - Minimize the L2-distance between activations of the local model and those of the global model

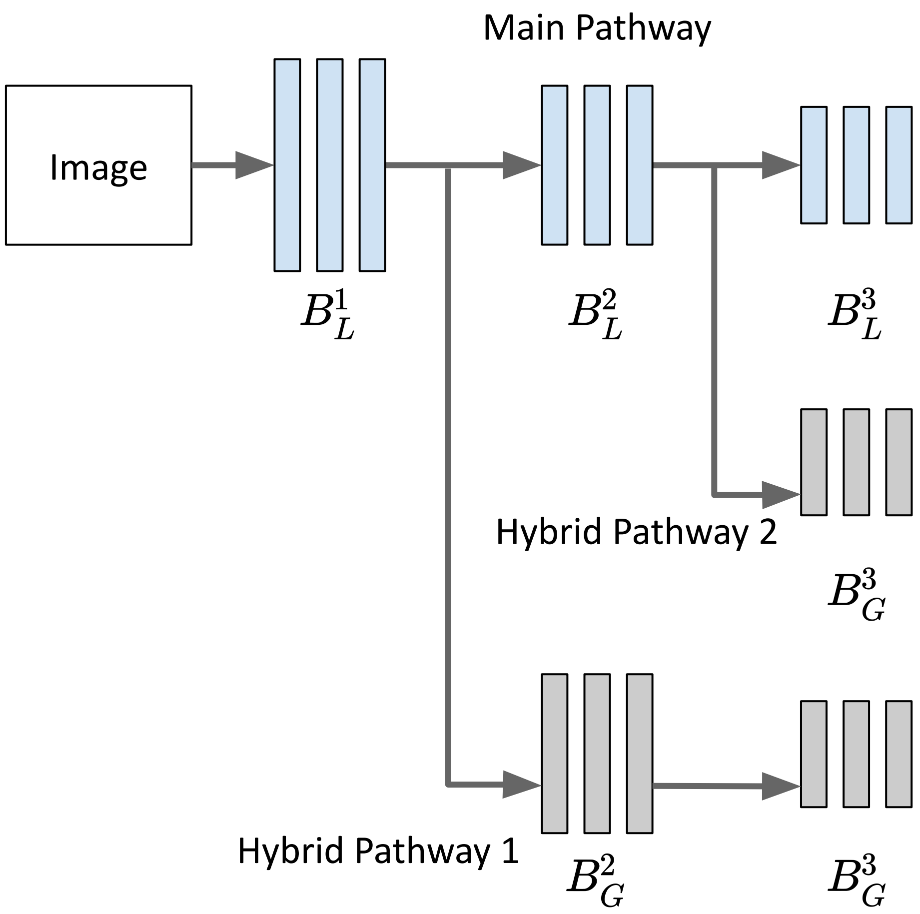
☺ All intermediate layers are affected

☹ Independent supervisions at multiple layers may lead to inconsistent and restrictive updates of model parameters

☹ Still merely simulating the fixed output of the global model

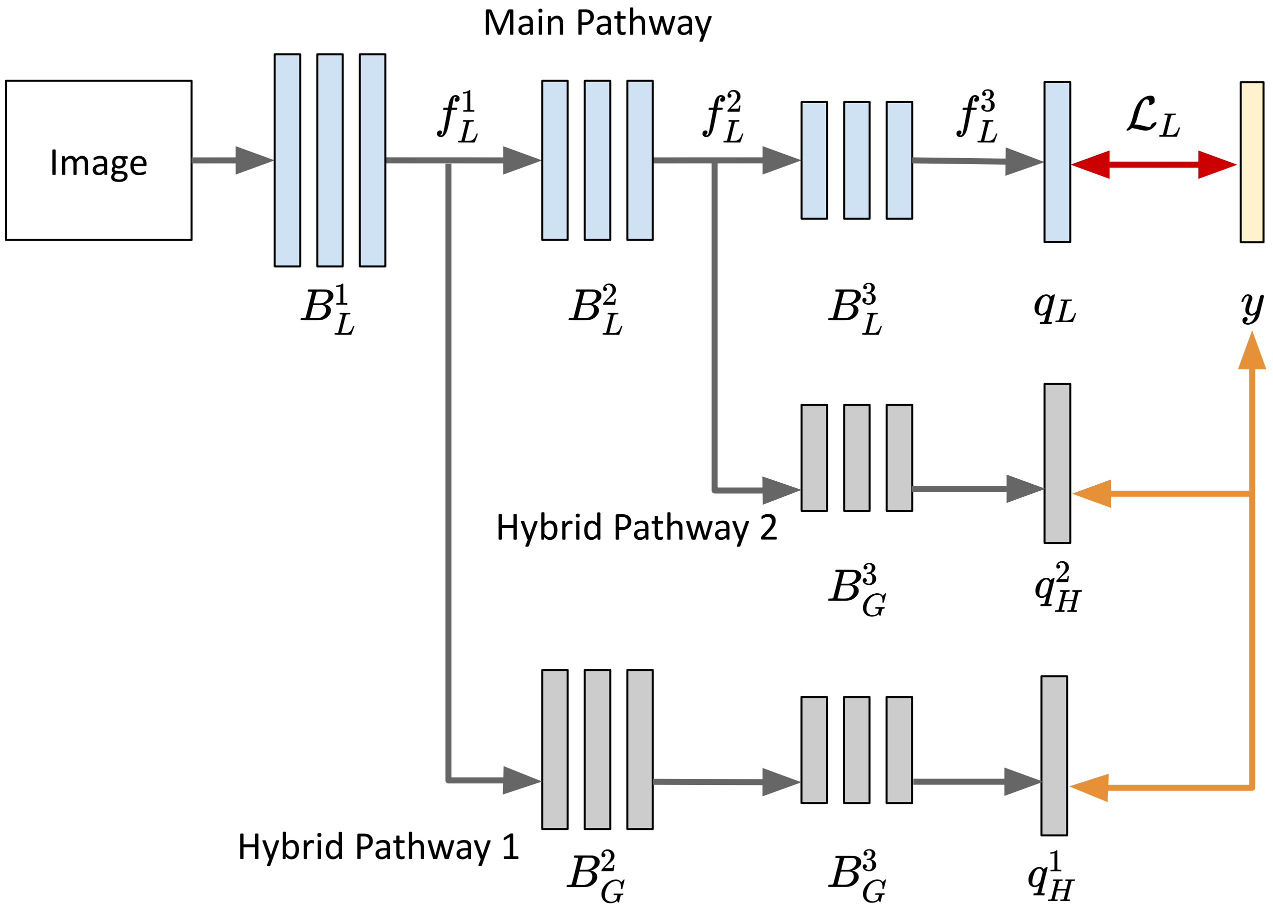
Multi-Level Branched Regularization (FedMLB)

- Multi-level hybrid branching
  - Augment a subnetwork in the global network $B_G^{m+1:M}$ to a local subnetwork $B_L^{1:m}$
  - Construct different **hybrid pathways** depending on branching locations



Multi-Level Branched Regularization (FedMLB)

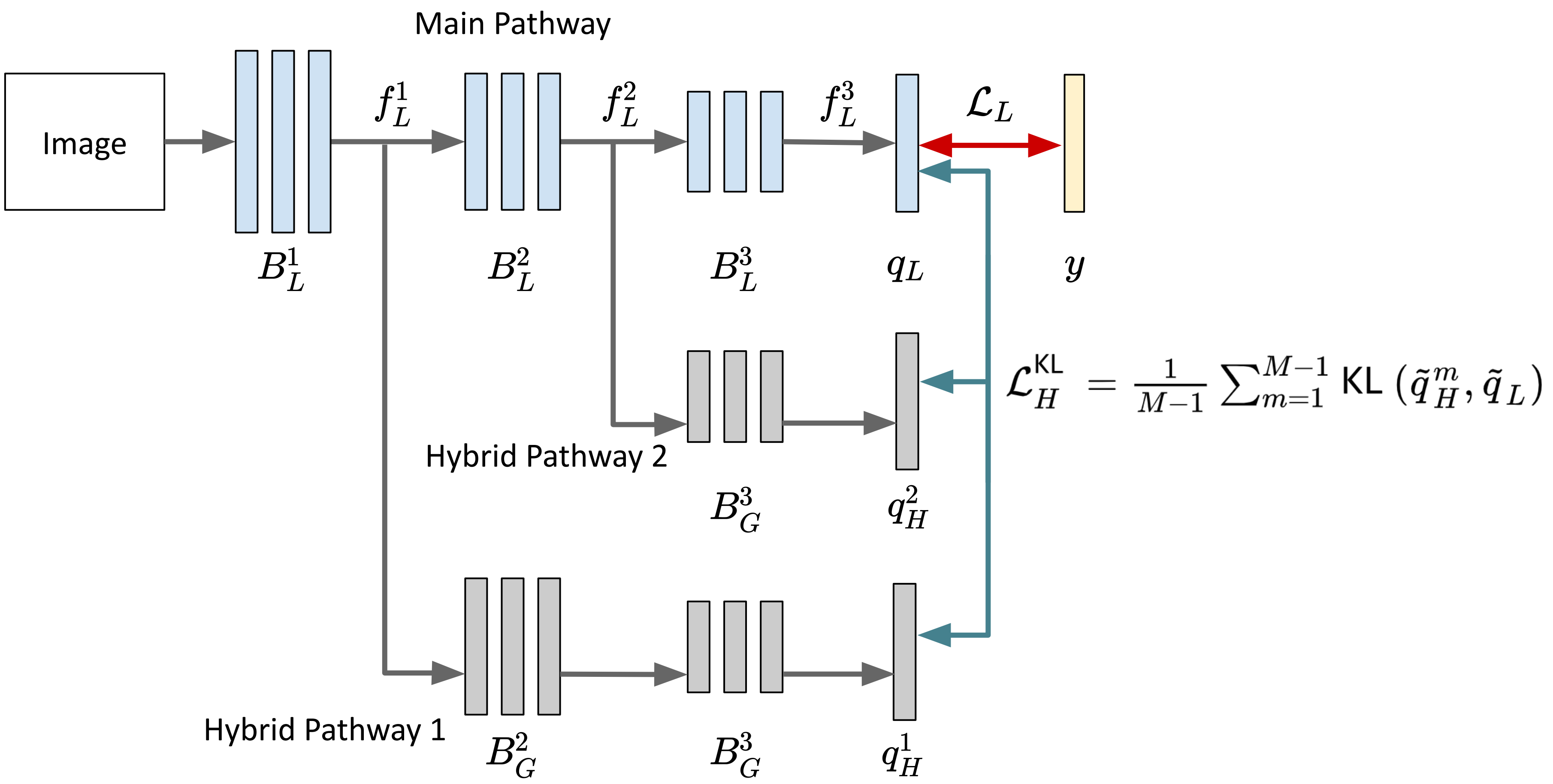
- Online knowledge distillation
  - Encourage the representation of individual hybrid pathways to be **similar** to the main branch
  - Use two different loss terms; **the cross-entropy loss** $\mathcal{L}_H^{\text{CE}}$ and **the knowledge distillation loss** $\mathcal{L}_H^{\text{KL}}$



$$\mathcal{L}_H^{\text{CE}} = \frac{1}{M-1}\sum_{m=1}^{M-1} \text{CrossEntropy}(q_H^m, y)$$

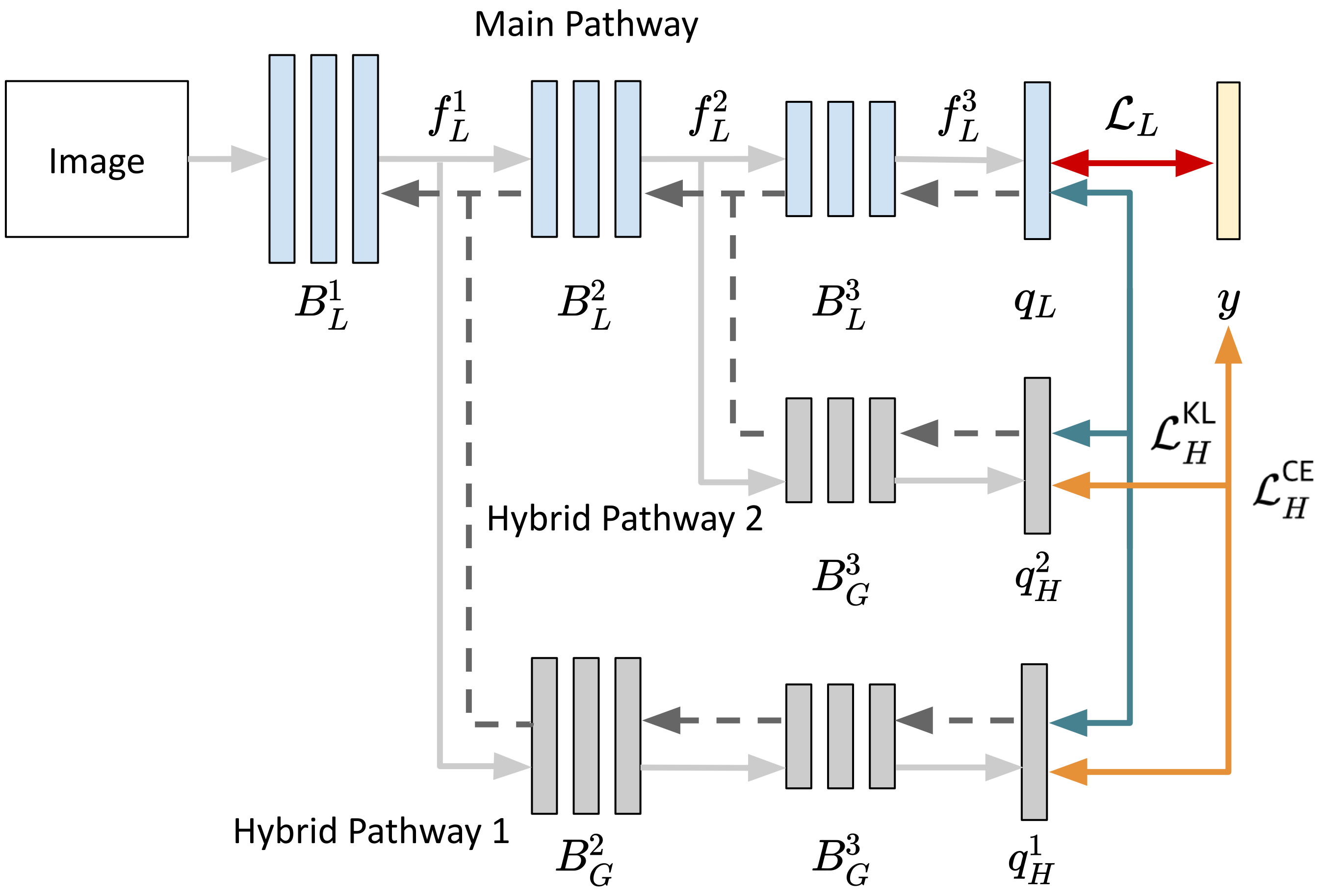
Multi-Level Branched Regularization (FedMLB)

- Online knowledge distillation
  - Encourage the representation of individual hybrid pathways to be **similar** to the main branch
  - Use two different loss terms; **the cross-entropy loss** $\mathcal{L}_H^{\text{CE}}$ and **the knowledge distillation loss** $\mathcal{L}_H^{\text{KL}}$



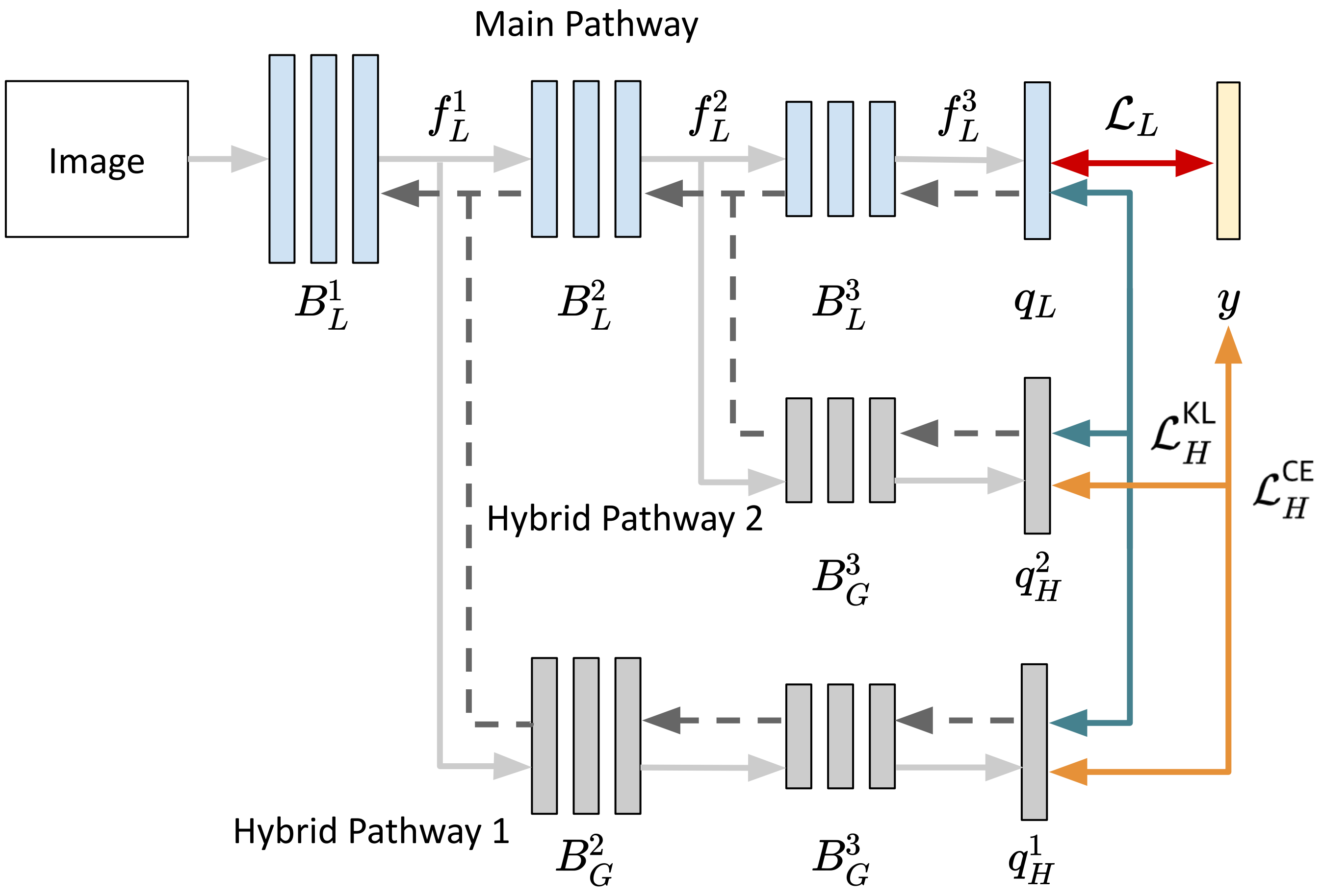
Multi-Level Branched Regularization (FedMLB)

- Final loss function
  - $\mathcal{L} = \mathcal{L}_L + \lambda_1 \cdot \mathcal{L}_H^{\text{CE}} + \lambda_2 \cdot \mathcal{L}_H^{\text{KL}}$
  - Update the model parameters in **the main pathway** of the local network while the blocks from the global network in the hybrid pathways remain **unchanged**



Multi-Level Branched Regularization (FedMLB)

- Final loss function
  - $\mathcal{L} = \mathcal{L}_L + \lambda_1 \cdot \mathcal{L}_H^{\text{CE}} + \lambda_2 \cdot \mathcal{L}_H^{\text{KL}}$
  - Update the model parameters in **the main pathway** of the local network while the blocks from the global network in the hybrid pathways remain **unchanged**



☺ Constrains the representation of the main pathway using **on-the-fly** outputs of the hybrid pathways

☺ Distributes the workload of the network **across multiple blocks** for adapting the local data

Multi-Level Branched Regularization (FedMLB)

- Learning procedure

Algorithm 1 FedMLB

Input: # of clients N, # of communication rounds T,
# of local iterations K, initial server model θ^0

for each round $t = 1, \dots, T$ **do**

  Sample a subset of clients $S_t \subseteq \{1, \dots, N\}$.

  Server sends θ^{t-1} to each of all clients $i \in S_t$.

  for each $i \in S_t$, **in parallel do**

    $\theta^t_{i,0} \leftarrow \theta^{t-1}$

    for $k = 1, \dots, K$ **do**

      for each (x, y) in a batch **do**

        $q_L(x;\tau) \leftarrow \text{softmax}\left(\frac{f_L(x;\theta^t_{i,k-1})}{\tau}\right)$

        $q^m_H(x;\tau) \leftarrow \text{softmax}\left(\frac{f^m_H(x;\theta^t_{i,m,k-1})}{\tau}\right)$, $m = 1, \dots, M-1$

      end for

      $\mathcal{L}(\theta^t_{i,k-1}) \leftarrow \mathcal{L}_L + \lambda_1 \cdot \mathcal{L}^{\text{CE}}_H + \lambda_2 \cdot \mathcal{L}^{\text{KL}}_H$

      $\theta^t_{i,k} \leftarrow \theta^t_{i,k-1} - \eta \nabla \mathcal{L}$

    end for

    Client sends $\theta^t_{i,K}$ back to the server

  end for

  In server:

    $\theta^t = \frac{1}{|S_t|} \sum_{i \in S_t} \theta^t_{i,K}$

end for

☺ No requirement of additional communication overhead

Multi-Level Branched Regularization (FedMLB)

- Learning procedure

Algorithm 1 FedMLB

Input: # of clients N, # of communication rounds T,
# of local iterations K, initial server model θ^0

for each round $t = 1, \ldots, T$ **do**
Sample a subset of clients $S_t \subseteq \{1, \ldots, N\}$.
Server sends θ^{t-1} to each of all clients $i \in S_t$.
for each $i \in S_t$, **in parallel do**
$\theta^t_{i,0} \leftarrow \theta^{t-1}$
for $k = 1, \ldots, K$ **do**
for each (x, y) in a batch **do**
$q_L(x; \tau) \leftarrow \text{softmax}\left(\frac{f_L(x; \theta^t_{i,k-1})}{\tau}\right)$
$q^m_H(x; \tau) \leftarrow \text{softmax}\left(\frac{f^m_H(x; \theta^t_{i,m,k-1})}{\tau}\right)$,
$m = 1, \ldots, M-1$
end for
$\mathcal{L}(\theta^t_{i,k-1}) \leftarrow \mathcal{L}_L + \lambda_1 \cdot \mathcal{L}^{\text{CE}}_H + \lambda_2 \cdot \mathcal{L}^{\text{KL}}_H$
$\theta^t_{i,k} \leftarrow \theta^t_{i,k-1} - \eta \nabla \mathcal{L}$
end for
Client sends $\theta^t_{i,K}$ back to the server
end for
In server:
$\theta^t = \frac{1}{|S_t|} \sum_{i \in S_t} \theta^t_{i,K}$
end for

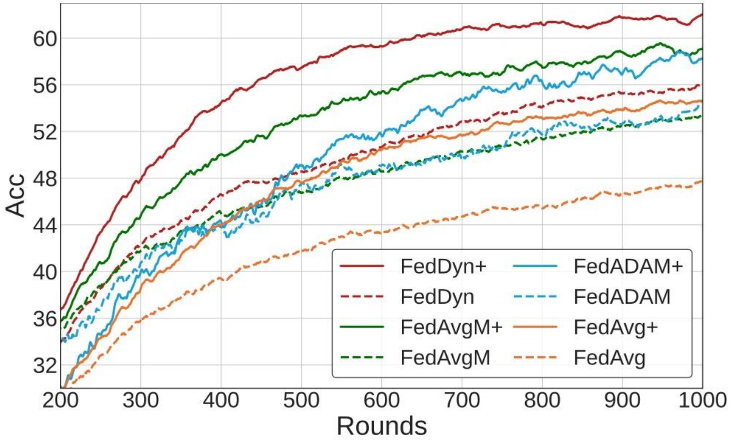
☺ No requirement of additional communication overhead

☺ Clients are not supposed to store historical information of the model

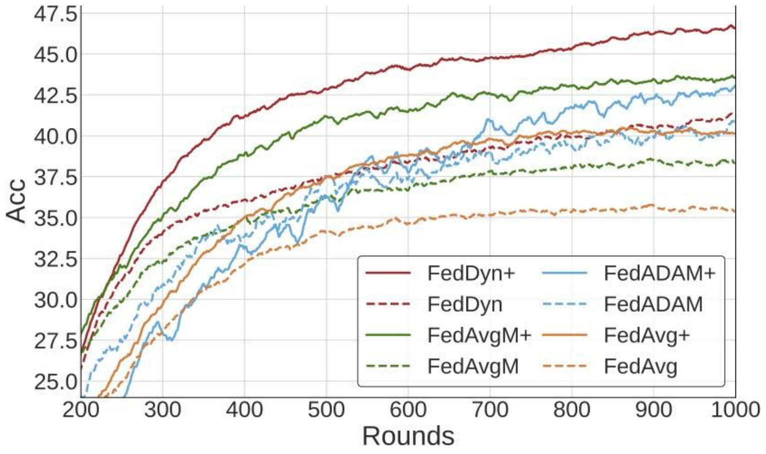
Experimental Results

- FedMLB with server-side optimization techniques
  - Moderate-scale with Dir(0.3): 100 clients, 5% participation

Method	CIFAR-100				Tiny-ImageNet			
	Accuracy (%, ↑)		Rounds (#, ↓)		Accuracy (%, ↑)		Rounds (#, ↓)	
	500R	1000R	47%	53%	500R	1000R	38%	42%
FedAvg (McMahan et al., 2017)	41.88	47.83	924	1000+	33.94	35.42	1000+	1000+
FedMLB	**47.39**	**54.58**	**488**	**783**	**37.20**	**40.16**	**539**	1000+
FedAvgM (Hsu et al., 2019)	46.98	53.24	515	936	36.10	38.36	794	1000+
FedAvgM + FedMLB	**53.02**	**58.97**	**349**	**499**	**40.93**	**43.52**	**380**	**642**
FedADAM (Reddi et al., 2021)	47.07	54.19	499	947	**36.98**	40.60	647	1000+
FedADAM + FedMLB	**48.59**	**58.23**	**472**	**645**	35.81	**42.90**	**552**	**873**
FedDyn (Acar et al., 2021)	48.38	55.78	425	735	37.35	41.17	573	1000+
FedDyn + FedMLB	**57.33**	**61.81**	**299**	**377**	**43.05**	**46.55**	**324**	**446**



(a) CIFAR-100

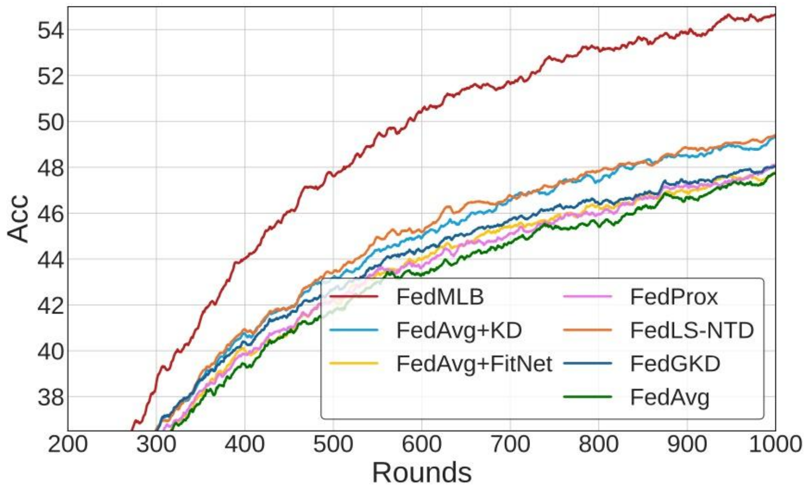


(b) Tiny-ImageNet

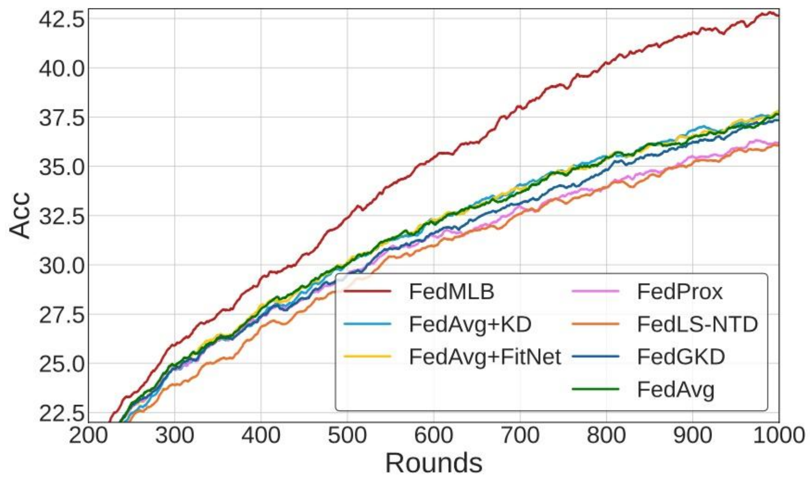
Experimental Results

- Comparison with other local objectives on CIFAR-100

Method	Dir(0.3), 100 clients, 5%				Dir(0.3), 500 clients , 2%			
	Accuracy (%, ↑)		Rounds (#, ↓)		Accuracy (%, ↑)		Rounds (#, ↓)	
	500R	1000R	40%	48%	500R	1000R	30%	36%
FedAvg (McMahan et al., 2017)	41.88	47.83	428	1000+	29.87	37.48	504	858
FedAvg + KD (Hinton et al., 2014)	42.99	49.17	389	842	29.83	37.65	505	859
FedAvg + FitNet (Romero et al., 2015)	42.04	47.67	419	1000+	29.92	37.63	503	860
FedProx (Li et al., 2020a)	42.03	47.93	419	1000+	29.28	36.16	533	966
FedLS-NTD (Lee et al., 2021)	43.22	49.29	386	825	28.66	35.99	546	1000+
FedGKD (Yao et al., 2021)	42.28	47.96	397	1000+	29.27	37.25	530	896
FedMLB (ours)	**47.39**	**54.58**	**339**	**523**	**32.03**	**42.61**	**446**	**642**



(a) Dir(0.3), 100 clients, 5%



(a) Dir(0.3), 500 clients, 2%

Conclusion

- A simple but effective **architectural regularization** technique to handle **heterogeneous** data distribution involved in federated learning
  - Online distillation between the main pathway and multiple hybrid pathways
    - Reduce the drift of the representations in the local models from the feature space of the global model
  - Two desired properties
    - No additional communication cost
    - No requirement to store the history of local states
  - Demonstrate remarkable performance gains in terms of **accuracy** and **efficiency** compared to existing methods.