Federated Black-Box Adaptation for Semantic Segmentation

Jay N. Paranjape
Dept. of Electrical and Computer Engineering
The Johns Hopkins University
Baltimore, USA
jparanj1@jhu.edu

Shameema Sikder
Wilmer Eye Institute
The Johns Hopkins University
Baltimore, USA

S. Swaroop Vedula
Malone Center for Engineering in Healthcare
The Johns Hopkins University
Baltimore, USA

Vishal M. Patel
Dept. of Electrical and Computer Engineering
The Johns Hopkins University
Baltimore, USA

Abstract

Federated Learning (FL) is a form of distributed learning that allows multiple institutions or clients to collaboratively learn a global model to solve a task. This allows the model to utilize the information from every institute while preserving data privacy. However, recent studies show that the promise of protecting the privacy of data is not upheld by existing methods and that it is possible to recreate the training data from the different institutions. This is done by utilizing gradients transferred between the clients and the global server during training or by knowing the model architecture at the client end. In this paper, we propose a federated learning framework for semantic segmentation without knowing the model architecture nor transferring gradients between the client and the server, thus enabling better privacy preservation. We propose *BlackFed* - a black-box adaptation of neural networks that utilizes zero order optimization (ZOO) to update the client model weights and first order optimization (FOO) to update the server weights. We evaluate our approach on several computer vision and medical imaging datasets to demonstrate its effectiveness. To the best of our knowledge, this work is one of the first works in employing federated learning for segmentation, devoid of gradients or model information exchange. Code: `https://github.com/JayParanjape/blackfed/tree/master`

1 Introduction

With data-driven methods becoming immensely popular in Artificial Intelligence (AI) research and applications, there has been a surge in data collection and curation across the world. This has, in turn, led to the development of AI models that require substantial amounts of data to train. Federated Learning (FL) [34] was developed as a viable approach towards training such models by effectively harnessing the data collected at different centers across the world. Through FL, it becomes possible for multiple institutions to collaborate and build a joint model that learns from all of them, while reducing the burden of collecting more data individually. However, collaborations between different institutions present a non-trivial challenge due to disparity in data distributions as well as the imperative of safeguarding data privacy. Consequently, FL aims to jointly learn a shared model that performs well on data from all participating institutions without exchanging raw data. Various FL approaches have been proposed in the literature [1, 10, 12, 16, 25, 22, 35, 37, 41, 44, 55, 56, 30, 34, 29, 31]. Among

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

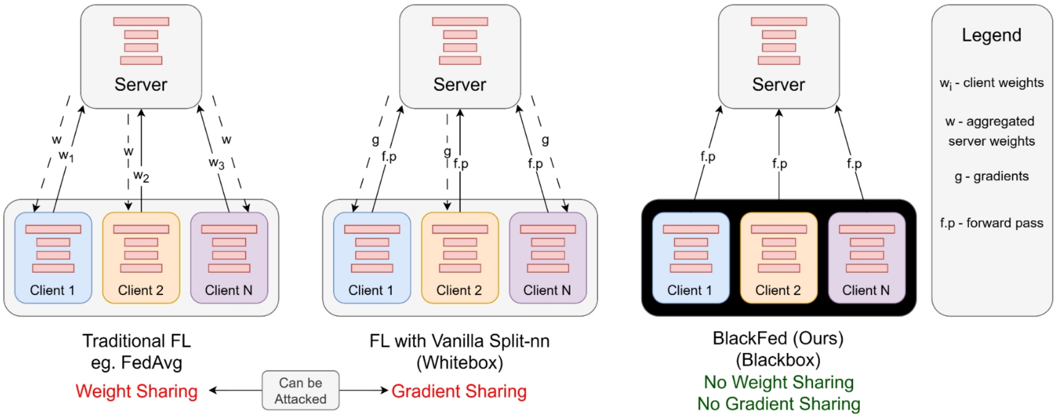

Figure 1: Comparison of our method against traditional FL methods. Existing FL methods are primarily "white-box" as they involve transfer of model weights [34], or gradients[20]. In contrast, our method only utilizes forward passes to update the client and does not require sharing weights or gradients, making it a "black-box" model.

these, FedAvg [34] was one of the pioneering FL methods, which proposes training local models at every center using local data and periodically averaging the local model weights to craft a global server model. Various subsequent works improve FedAvg by improving the local model updates [30, 31] or global updates [24, 29].

While FL was primarily proposed for classification tasks, it is also suited for other computer vision tasks such as segmentation. Generating annotations in segmentation entails creating pixel-level annotations per image, that are more tedious to label than classification tasks. Consequently, it is not always possible for a single institution to collect a large amount of data, underscoring the importance of collaborative efforts. A few approaches have been proposed in the literature for FL-based segmentation [36, 5, 18, 43, 53, 14]. However, all these methods for segmentation and classification, while not involving raw data transfer, employ techniques like model information transfer [34, 30, 36] or gradient transfer [24, 20], as shown in Figure 1. However, recent research has revealed that these techniques are vulnerable to attacks that can recreate training data from the participating centers, thus undermining the privacy preserving characteristic of FL [11, 54, 17, 57, 27, 26, 58, 15, 23]. These attacks employ methods like gradient inversion [54, 17, 57] or adapting model architecture and weights [11]. In this work, we propose a new approach, named BlackFed, for segmentation using FL that does not involve gradient transfer between the server and the client and at the same time, passes no knowledge about the client model architecture to the server, thereby avoiding the necessary conditions for these attacks, as shown in Figure 1. This is done by formulating the FL learning problem as a distributed learning problem using split neural networks (split-nn) [20] and combining first order and zero order optimization techniques for training. Our contributions are as follows:

1. We introduce BlackFed - a black-box algorithm that facilitates distributed learning for semantic segmentation without transferring model information or gradients between the client and the server. For this, we formulate the FL problem using split-nn and use first and zero order optimization for training the server and the clients, respectively.
2. We suggest a method to reduce the effect of catastrophic forgetting in BlackFed by retaining client-wise checkpoints in the server.
3. We evaluate the proposed approach on four segmentation datasets and show its effectiveness as a distributed learning method by showing improvements over individual training.

2 Related Work

Federated Learning for Segmentation: FL for the segmentation tasks was motivated by its immense application in the medical domain. Consequently, various methods were introduced in this field [33, 6, 13, 19, 28, 32, 42, 49, 50, 52, 38]. For instance, [32] attempts to learn a model for brain tumor segmentation by utilizing data from multiple institutions. FedSM [38] attempts to mitigate the effect of non-iid nature of the data from different centers on the global server model. However, most of the approaches for medical segmentation focus on the problem of segmenting out the foreground from the background (one class problem). There are relatively fewer works in the literature for multi-class segmentation [14, 36, 5, 18, 43, 53]. FedSeg [36] deals with the class label inconsistency problem that may be present at the local clients. In other words, it builds a robust system that works well even when the clients have annotations for only a subset of the classes. FedDrive [14], on the other hand, sets up various benchmarks for FL algorithms on multi-class datasets like Cityscapes [9]. However, all the existing algorithms require the global server to know the model architecture used in the clients and thus, these methods are vulnerable to recent attacks [11]. In our work, we develop an algorithm to perform multi-class segmentation which does not require gradients or client model sharing.

Split Neural Networks: Split networks [20] were introduced as an alternative to FedAvg-like techniques which require model sharing. They offer an approach to perform collaborative learning by splitting a larger network into two segments. The latter segment of the network is shared across all centers and placed at the global server, while the former part is distributed such that each center possesses its own sub-network. During training, the clients perform a forward pass using their data and send the encoded features to the server, which further passes the features to the subsequent layers in the network. The server and client models are trained using backpropagation, where the gradients from the first layer of the server model are sent back to each of the clients. Split networks have mainly been used in the literature for the task of classification, mostly in the medical domain [20, 40, 47]. Our work marks one of the initial explorations of split learning in the context of semantic segmentation. Furthermore, split networks are shown to be more robust to reconstruction attacks which arise from sharing model information [48]. Nonetheless, they remain susceptible to gradient leakage attacks since there is gradient transfer between the server and the client. In this work, we formulate the problem of distributed semantic segmentation using split network and introduce a training algorithm that does not require gradient-sharing.

3 Black-box Adaptation

In the following section, we define the problem of semantic segmentation under the federated setting. Then, we describe our proposed formulation for the black-box setting and the algorithm.

3.1 Preliminaries

The setting of FL consists of N clients, denoted by $\mathbb{C}_i$, where $i \in 1, 2, ..., N$ and a global server $\mathbb{G}$. Each of the clients has a local dataset, consisting of images $X_i = \{x^i_j \in \mathbb{R}^{H\times W\times C}; j \in \{1, 2, ..., n^i\}\}$ and their corresponding ground truth segmentation maps $Y_i = \{y^i_j \in \mathbb{R}^{H\times W\times N_c}; j \in \{1, 2, ..., n^i\}\}$. Here, x^i_j and y^i_j represent the j^{th} image and its corresponding segmentation mask, while n^i represents the number of data points in client i and N_c represents the number of classes in the output. Note that the distributions of the input images vary among the clients. Each client uses its dataset to learn a function $f^i : \mathbb{R}^{H\times W\times C} \to \mathbb{R}^{H\times W\times N_c}$, which is parameterized by Θ^i, that minimizes its local loss function as follows:

$$\arg\min_{\Theta^i} \mathbb{L}_i = \frac{1}{n^i}\sum_{j=1}^{n^i} l(f(x^i_j; \Theta^i), y^i_j), \tag{1}$$

where l denotes the loss per data point. The global server, on the other hand, tries to aggregate the knowledge of all the clients by fusing the parameters of the individual clients to optimize a joint loss function as follows:

$$\arg\min_{\Theta^i} \mathbb{L} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{L}_i. \tag{2}$$

A commonly used algorithm, called FedAvg [34], proposes a linear combination of client parameters to optimize the loss function, defined in Eq. 2, as follows:

$$\Theta^i \leftarrow \sum_{i=1}^{N} \frac{n^i}{\sum_{i=1}^{N} n^i}\Theta^i. \tag{3}$$

However, this formulation requires the server to be aware of the exact model architecture or function of the client, which can results in data leakage [11].

3.2 Proposed Algorithm

Proposed Problem Formulation: In this work, we model the problem of distributed learning using a split neural network, which takes away the requirement of the server being aware of the client architecture. In this case, each client learns a function $f^i : \mathbb{R}^{H\times W\times C} \rightarrow \mathbb{R}^{H'\times W'\times C'}$, which is parameterized by Θ^i. Similarly, the global server learns a function $g : \mathbb{R}^{H'\times W'\times C'} \rightarrow \mathbb{R}^{H\times W\times N_c}$, which is parameterized by Φ. Thus, the forward pass for a given center is given as follows:

$$\hat{y}^i_j = g(f(x^i_j; \Theta^i); \Phi), \tag{4}$$

where $\hat{y}^i_j$ denotes the predicted segmentation map. Hence, the objective function of a given client is as follows:

$$\underset{\Theta^i,\Phi}{\arg\min}\, \mathbb{L}^i, i \in \{1, 2, ..., N\} = \frac{1}{n^i}\sum_{j=1}^{n^i} l(\hat{y}^i_j, y^i_j; \Theta^i, \Phi). \tag{5}$$

As in the existing FL literature, the goal of our approach is that after training, all clients should benefit from each other. Hence, during evaluation, given any client, we aim to perform well on data from other clients as well as its own data, thus showing good generalization. More specifically, we want to optimize any combination of data and client as follows:

$$\min \mathbb{L}^{ik}, i \in \{1, 2, ..., N\}, k \in \{1, 2, ..., N\} = \frac{1}{n^i}\sum_{j=1}^{n^i} l(\hat{y}^i_j, y^i_j; \Theta^k, \Phi). \tag{6}$$

One way to optimize this equation is by attending to every client in a round-robin fashion. This involves selecting a client, performing the forward pass, as defined in Eq. 4 and then performing backpropagation to update the server and client using the client loss function defined in Eq. 5. Performing this operation several times in a round-robin fashion enables the server to learn from all client sources, and hence, improves the overall performance. We term this method as "White-box Round-Robin FL", and is similar to the existing methods in the literature [24, 20]. However, recent works have shown that such a method which involves transfer of gradients between the server and client can be utilized to regenerate the training data, thus undermining the privacy preservation principle of FL [54, 23]. Hence, we add one more constraint - i.e. no gradient can flow back from the server to the client in Eq. 5.

BlackFed Algorithm: To optimize the clients without using gradients, we utilize a ZOO method called Simultaneous Perturbation Stochastic Approximation with Gradient Correction (SPSA-GC) [39] which involves perturbing the weights of the client model slightly and approximating a two-sided gradient based on the change in the loss function due to the perturbations. However, this method was developed to perform black-box adaptation of pretrained foundation models and hence, expects the server model to be initialized with good weights, which is not the case in our formulation, making it non-trivial. To overcome this, we propose to iteratively use an alternating optimization technique, which factorizes the optimization problem in Eq. 5 into two optimization problems as follows:

$$\begin{aligned} &\underset{\Theta^i}{\arg\min}\, \mathbb{L}^i, i \in \{1, 2, ..., n\} = \frac{1}{n^i}\sum_{j=1}^{n^i} l(\hat{y}^i_j, y^i_j; \Theta^i|\Phi), \\ &\underset{\Phi}{\arg\min}\, \mathbb{L}^i, i \in \{1, 2, ..., n\} = \frac{1}{n^i}\sum_{j=1}^{n^i} l(\hat{y}^i_j, y^i_j; \Phi|\Theta^i). \end{aligned} \tag{7}$$

During training, we first select a client using the round-robin policy. Next, keeping the server weights fixed, we train the client weights using SPSA-GC for a few iterations. Next, we fix the client weights and use a first order optimizer (i.e. Adam-W) to optimize the server for a few iterations. This process is repeated multiple times. During inference for a client, we simply run the forward pass as described in Eq. 4 using the final weights of the client and server. The training process is described in Algorithm 1. We refer to this approach as BlackFed v1.

Algorithm 1 Proposed Algorithm for BlackFed v1

```
Input: (i) N, number of clients
  (ii) c_e, number of epochs to train one client
  (iii) s_e, number of epochs to train server
  (iv) runs, number of complete round-robin runs  $n \geq 0$
Output: (i) $\Phi$, server model weights
  (ii) $\Theta$, array of client model weights

  BlackFed (N, c_e, s_e, runs) Begin:
  initialize $\Phi, \Theta$
  for r = 1,2,...,runs do
      for i = 1,2,...,N do
          for j = 1,2,...,c_e do
              $\Theta[i] \leftarrow$ SPSA-GC($\Theta[i]$)    //Zero order optimization for client
          end for
          for j = 1,2,...,s_e do
              $\Phi \leftarrow \Phi - \eta_s \frac{\partial \mathbb{L}^i}{\partial \Phi}$      //First order optimization for server, where $\eta_s$ is the learning rate
          end for
      end for
  end for
  Return: $\Phi, \Theta$
  End
```

Reducing the Effect of Catastrophic Forgetting: Since the model at the server is shared among all the clients and is updated in a round-robin fashion, it may happen that training with the data from a given client may cause it to unlearn patterns for the previous client. This phenomenon is often called catastrophic forgetting. This effect is observed in BlackFed v1 especially when the number of clients increases or if there is a significant change in the data distribution among clients. This causes the algorithm to perform well on certain clients and poorly on the rest of the clients. To reduce the effect of catastrophic forgetting, we propose a simple additional step during training with Algorithm 1. After updating the server weights for a given client during training, we store the updated weights of the server model in a hashmap indexed by the index of the client. During inference for a given client, we use the latest weights of the client model and the indexed weights of the server model to perform the forward pass. Note that the server state is loaded from the hashmap only during inference. During training, the server still benefits from the data from all clients and updates its weights as well as the hashmap. This approach is visualized in Figure 2. We refer to this method as BlackFed v2.

4 Experiments and Results

Datasets: For evaluating our method, we consider four publicly available datasets, namely (i) Cityscapes [9] (ii) CAMVID [4], (iii) ISIC [21, 8, 45] and (iv) Polypgen [2]. Cityscapes and CAMVID are two road-view semantic segmentation datasets with 19 and 32 classes of interest respectively, collected from multiple cities. In the FL setup, we consider each of the cities as separate clients. While CAMVID has predefined train, test and validation splits, for Cityscapes, we divide the data from each client into training, validation and testing splits for that client in a 60:20:20 ratio. In this manner, we generate 18 clients for Cityscapes and 4 clients for CAMVID. Further details about the dataset size in each center is provided in the supplementary document. The ISIC dataset corresponds to a skin lesion segmentation challenge. We generate three clients for this dataset, which utilize the datasets from ISIC 2016 [21], ISIC 2017 [8] and ISIC 2018 [45], respectively. The data every year is collected from different centers and hence, has different distribution amongst centers. Finally, PolypGen is a colon polyp segmentation dataset which was collected from six different centers. The training, validation and testing splits for each of these were done in a 60:20:20 ratio for Polypgen whereas they were already provided for ISIC. We visualize the pixel-intensity histograms of the CAMVID and ISIC datasets to verify different data distributions amongst clients in Figure 3.

Experimental Setup: We use a two-layer convolutional network for modeling f in the client and DeepLabv3 [7] for modeling g in the server. The number of output channels from the client is kept

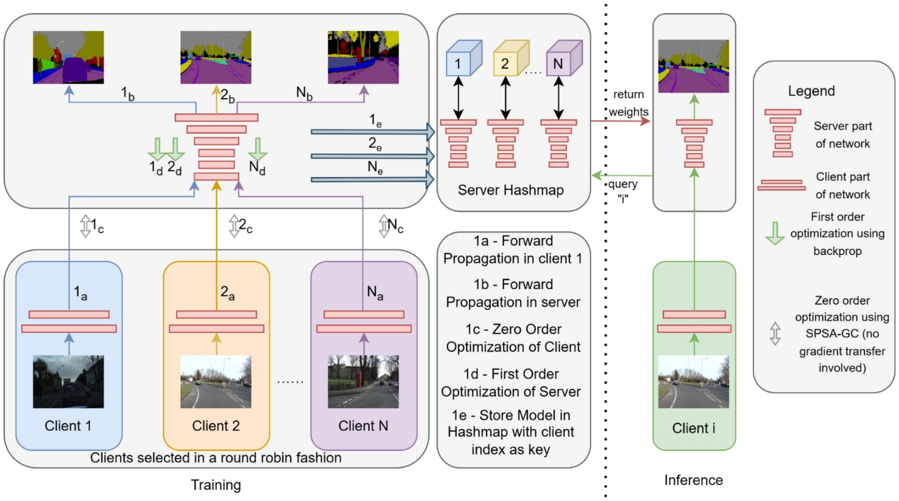
1_b
2_b
N_b
1_e
2_e
N_e
1_d 2_d
N_d
1_c
2_c
N_c
1_a
2_a
N_a
Client 1
Client 2
Client N
Clients selected in a round robin fashion
Training
1
2
N
Server Hashmap
return weights
query "i"
1a - Forward Propagation in client 1
1b - Forward Propagation in server
1c - Zero Order Optimization of Client
1d - First Order Optimization of Server
1e - Store Model in Hashmap with client index as key
Client i
Inference
Legend
Server part of network
Client part of network
First order optimization using backprop
Zero order optimization using SPSA-GC (no gradient transfer involved)

Figure 2: The BlackFed v2 Algorithm. During training, the client is selected in a round-robin fashion. Then (a) client performs a forward pass using its part of the network (b) Server performs a forward pass using its part of the network (c) With server weights fixed, client weights updated using ZOO (d) Keeping client weights fixed, server weights updated using FOO (e) The best server weights are stored in the hashmap corresponding to client index. During inference, the client performs a forward pass and calls the server with the output. Server queries the hashmap using the client index and gets its set of weights, using which the prediction is obtained. Note that there is no gradient transfer, thus making this a black-box setup.

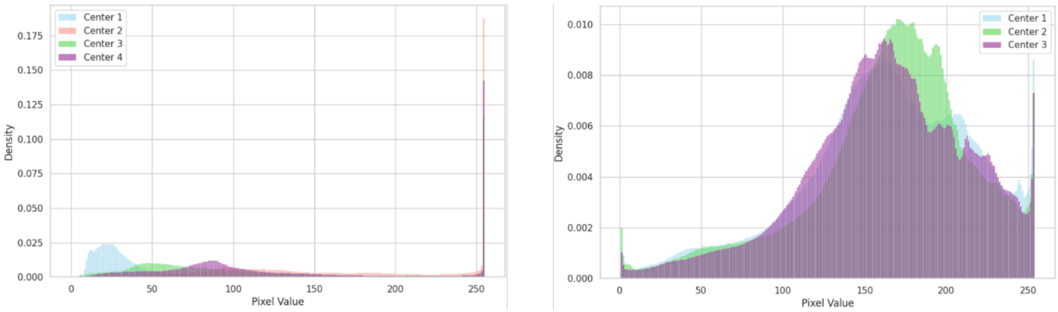

Figure 3: Pixel Density distribution of (L) the CAMVID Dataset and (R) the ISIC Dataset. Since majority of ISIC pixels are either 0 or 255 for all centers, these have been omitted for better visualization. Since each of the clients has a different distribution, data from one client can be considered as Out-of-Distribution (OOD) for other clients.

as 64. Consequently, we start the DeepLabv3 network in the server from the second layer, which expects a 64-channel input. During training, we use $c\_e = 10$ and $s\_e = 10$. The server is optimized using an Adam optimizer and the client is optimized using SPSA-GC. The learning rates of both the client and the server are set to 10^{-4}, based on validation set performance. The batch size for all experiments is 8, and all images undergo random brightness perturbation with brightness parameter set as 2. The images for Cityscapes and CAMVID are resized to 256×512 to maintain their aspect ratio, whereas the images for ISIC and Polypgen are resized to 256×256. All experiments are done using a single Nvidia RTX A5000 GPU per client and a single Nvidia RTX A5000 GPU at the server.

Results: Given the trained client models and the trained server model, we assess a given client's performance with test datasets from its own local data repository and present the mIoU in the "Local" column. This metric represents the local performance of the model on its own dataset. In addition, we assess each client on test data from other centers and note down the average mIoU scores in the Out-of-Distribution ("OOD") column. This metric serves as an estimate of the general performance of a given client post-training. We consider the latter more important as it can be considered to be the direct outcome of the FL paradigm. These results are presented in Table 1 for CAMVID and Cityscapes, and in Table 2 for ISIC and Polypgen. For both tables, in the first row, each client is

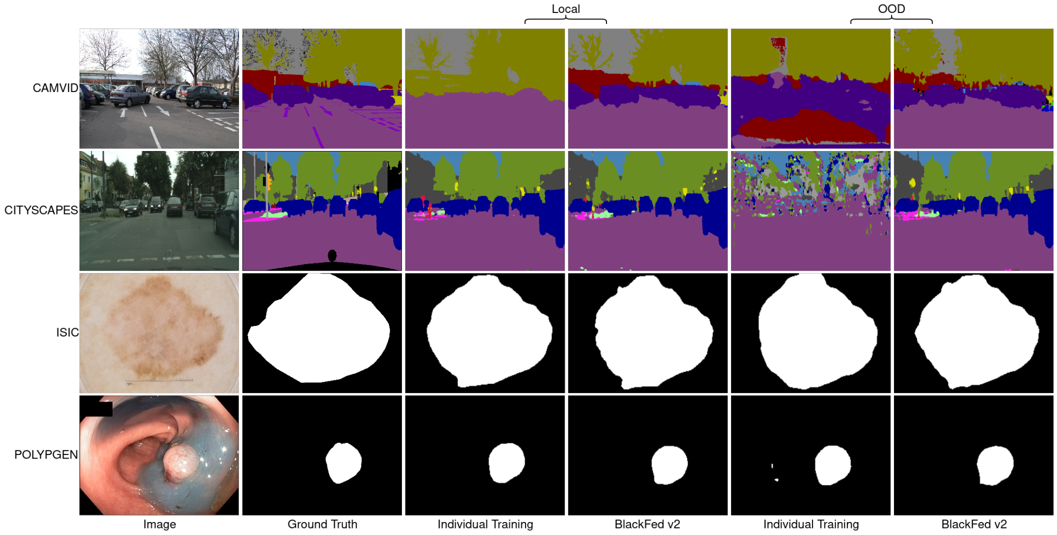

Figure 4: Comparison of our method against individual training. The third and fourth columns denote testing with the local test data, while the fifth and sixth columns denote OOD testing. Our method improves OOD performance of clients without harming their local performance.

trained on its own dataset, independently of others, indicating no collaboration. For this case, while the client performance on its own test set is high, its general performance suffers. The next two rows represent our method (BlackFed v1 and v2). Notably, BlackFed v2 generally exhibits better performance than v1 since it addresses the catastrophic forgetting that occurs during training. The following three rows represent the expected upper bounds to our performance. "Combined Training" represents the scenario where raw data can be freely shared and a single model is trained using the combined data from all clients. "White-box training" represents the case where gradients can flow freely between the server and the client. Thus, instead of the ZOO optimization, we use FOO to optimize the client part of the model. Finally, the last three rows represent the performance of traditional FL using FedAvg, and recent methods like FedPer and FedSeg, where model sharing is allowed. Here, all the clients follow the same model architecture as the server (DeepLabv3) and the server can aggregate weights from the client models. All three methods represent a relaxation over the constraints imposed in our approach, thus acting as an upper bound to the black-box performance. During training, the client and server models are trained for the same number of epochs per client. This produces a more uniform distribution of results across centers, in contrast with traditional FL methods like FedAvg, where the aggregation of weights is weighed by the client dataset size. We observe that for the Polypgen dataset, BlackFed v1 and v2 perform slightly better or on par with individual client performance for OOD case. For this case, there is little difference between the performance of v1 and v2. This behavior may be related to the data distribution of Polypgen and suggests that BlackFedv2 is not able to correctly avoid the catastrophic forgetting for centers C5 and C6. However, for rest of the scenarios, we see that v2 significantly outperforms v1 as well as the individual training cases on OOD mIoU metric. At the same time, the performance on local data does not suffer significantly as compared to the individual training. Moreover, BlackFed v2 performs on par with "Combined" and "White-box" Training. All results of BlackFed have a p-value less than 0.001, showing the statistical significance of our black-box approach. The visual comparisons for our method with the individual training is given in Fig. 4. As can be seen in all four rows, individual training can lead to overfitting, which harms the general OOD performance. Using our method, we are able to improve the OOD results across all datasets without significant decrease in the local results.

Additional Model Architectures While we evaluate a DeepLabv3-based server in our main experiments, we show the effectiveness of our method on additional segmentation architectures. More specifically, for the CAMVID dataset, we choose UNext [46] and SegFormer [51] as the server models and present average mIoU results across the test datasets from each client in Table 3. As can be seen from the table, using our approach can improve the performance over individual training for all three models. As the model complexity increases from UNext to DeepLab to Segformer, we

Table 1: mIoU scores for BlackFed v1 and v2 in comparison with individual and FL-based training strategies for natural datasets. "Local" represents test data from the center. "OOD" represents mean mIoU on test data from rest of the centers. For FedAvg and Combined Training, just one model is trained. Hence, its performance is noted only in each of the local test datasets. For Cityscapes, we only present the average local and OOD performance across centers for brevity. The supplementary contains an expanded version for Cityscapes.

	CAMVID								Cityscapes	
	C1		C2		C3		C4		Average across 18 Centers	
Method	Local	OOD	Local	OOD	Local	OOD	Local	OOD	Local	OOD
Individual	0.63	0.46	**0.83**	0.48	**0.85**	0.65	**0.79**	0.64	0.50	0.44
BlackFed v1	0.55	0.67	0.79	0.65	0.78	0.68	0.66	0.66	0.71	0.71
BlackFed v2	**0.70**	**0.72**	0.78	**0.66**	0.81	**0.70**	0.75	**0.72**	**0.75**	**0.74**
Combined Training	0.74	-	0.81	-	0.84	-	0.77	-	0.77	-
White-box Training	0.67	0.73	0.81	0.72	0.80	0.68	0.74	0.70	0.75	0.75
FedAvg [34]	0.64	-	0.76	-	0.84	-	0.76	-	0.79	-
FedSeg [36]	0.71	-	0.79	-	0.83	-	0.77	-	0.81	-
FedPer [3]	0.65	0.57	0.77	0.68	0.82	0.71	0.76	0.66	0.78	0.72

Table 2: mIoU scores for BlackFed v1 and v2 in comparison with individual and FL-based training strategies for medical datasets. "Local" represents test data from the center. "OOD" represents mean mIoU on test data from rest of the centers. For FedAvg and Combined Training, just one model is trained. Hence, its performance is noted only in each of the local test datasets.

	Polypgen												ISIC					
	C1		C2		C3		C4		C5		C6		C1		C2		C3	
Method	Local	OOD	Local	OOD	Local	OOD	Local	OOD	Local	OOD	Local	OOD	Local	OOD	Local	OOD	Local	OOD
Individual	0.59	0.47	**0.73**	0.47	**0.75**	**0.59**	0.47	0.37	**0.45**	0.44	**0.52**	**0.48**	0.86	0.79	0.85	0.76	0.76	0.84
BlackFed v1	0.59	0.55	0.63	**0.51**	0.64	0.56	**0.55**	0.47	0.34	**0.53**	0.28	0.40	0.84	**0.82**	**0.89**	0.80	0.5	0.69
BlackFed v2	**0.59**	**0.56**	0.64	0.49	0.65	0.51	0.50	**0.47**	0.35	0.49	0.26	0.41	**0.86**	0.80	0.88	**0.80**	**0.77**	**0.88**
Combined Training	0.71	-	0.79	-	0.81	-	0.67	-	0.57	-	0.60	-	0.87	-	0.89	-	0.76	-
White-box Training	0.60	64	0.77	0.59	0.72	0.58	0.61	0.57	0.48	0.62	0.59	0.62	0.56	0.70	0.73	0.66	0.77	0.68
FedAvg [34]	0.68	-	0.78	-	0.83	-	0.61	-	0.54	-	0.64	-	0.87	-	0.87	-	0.78	-

observe a decrease in individual training performance. This trend is reversed for the combined case where there is more training data. This observation indicates overfitting in the individual case due to less individual training data. Using our method, we are able to correctly match the performance and trend of combined training.

5 Ablation Studies

Analysis of the Order of Optimization: In the alternating optimization proposed in Eq. 7, we choose to first update the client followed by the server. This order is important since it allows us to store the correct server weights in the hashmap. If the server is trained before the client, we found that SPSA-GC often gives unstable results and reduces the metric on the validation set after an epoch. This is corrected only when its the turn of the same client again. Conversely, in the case where the client is updated first, the server adapts in a stable manner to the client weights since backpropagation is allowed in the server. Consequently, in the same epoch, we get a higher and more stable performance

Table 3: mIoU for CAMVID dataset with varying model architectures of the global server.

	DeepLabv3				Segformer				UNext			
Method	C1	C2	C3	C4	C1	C2	C3	C4	C1	C2	C3	C4
Individual	0.50	0.57	0.69	0.67	0.27	0.36	0.50	0.34	0.36	0.49	**0.60**	0.51
BlackFed v2	**0.72**	**0.69**	**0.73**	**0.72**	**0.71**	**0.69**	**0.73**	**0.72**	**0.61**	**0.53**	0.43	**0.56**
Combined Training	0.74	0.81	0.84	0.77	0.66	0.77	0.77	0.72	0.59	0.67	0.73	0.67
White-box Training	0.72	0.76	0.84	0.76	0.54	0.55	0.58	0.57	0.36	0.49	0.60	0.51

Table 4: Comparison of client and server-side GFLOPS for different algorithms.

	DeepLabv3 GFLOPS		Segformer GFLOPS		UNext GFLOPS	
Method	Client	Server	Client	Server	Client	Server
Centerwise	353.6	-	7.5	-	475.39	-
FedAvg [34]	353.6	353.6	7.5	7.5	475.39	475.39
Ours	26.6	326.98	5.12	2.38	53.13	422.26

Table 5: Average MIoU scores for different training strategies of BlackFed. Optimizing the client followed by the server improves performance, which is further improved by maintaining the server-side hashmap.

	CAMVID	ISIC	CityScapes	PolypGen
Method	Aveg across 3 centers	Average across 4 centers	Avg across 18 centers	Avg across 6 centers
Optimize Server, then client	0.67	0.66	0.53	0.43
Optimize Client, then Server (BlackFed v1)	0.67	0.76	0.53	0.50
Optimize Client, then Server (BlackFed v2)	**0.72**	**0.83**	**0.74**	**0.50**

for each client, which can be saved in the hashmap for usage during inference. We demonstrate this empirically by comparing the performance of the two training strategies in Table 5.

Analysis of Computational Cost: For each of the three model architectures, namely DeepLabV3, UNext and Segformer, we calculate the floating point operations required in a single forward pass at the institution end. These are shown in Table 4. If the clients were to train individual models with their local data, they would require running the entire forward pass on their local systems. This is also the case in existing FL approaches like FedAvg. In the proposed approach, as can be seen from Row 3 in the table, the client has a significantly reduced load, with the majority of computation being offloaded to the server. The server uses the respective architectures starting from the second layer, while all the clients use a two-layer convolution network, with the second convolutional layer being similar to the first layer of the respective architecture.

Effect of Client and Server rounds: The BlackFed algorithm takes in two hyperparameters, namely c_e and s_e, that determine the number of epochs for training the client and server, respectively. We conduct experiments for observing the effect of changing these parameters on the model performance. We use the CAMVID dataset with server architecture being DeepLabv3 and note down the average mIoU score in Table 6 for different values of c_e and s_e in $\{10, 20, 30, 40, 50\}$. Interestingly, we observe that there was no significant difference in performance by increasing the number of server epochs or client epochs. While increasing the server epochs can improve training, it also makes the server more specific to a given client, thus reducing average performance. On the other hand, more ZOO-based iterations marginally improve the model. In this context, ZOO primarily serves to guide the server towards a more robust minima, resilient to perturbations in client features. Consequently, a greater number of client epochs does not notably impact model performance.

Table 6: Average mIoU scores for BlackFed v1 and v2 in comparison with individual and FL-based training strategies.

	Number of Client Epochs				
Number of Server Epochs	10	20	30	40	50
10	**0.71**	0.69	0.68	0.69	0.68
20	0.70	0.69	0.69	0.69	0.69
30	0.65	0.68	0.70	0.70	0.66
40	0.71	0.70	0.68	0.70	0.70
50	0.67	0.69	0.70	0.66	0.69

6 Conclusion

In this work, we introduce BlackFed, an FL algorithm that enables distributed learning without transfer of gradients or model weights. This characteristic distinguishes our approach as a black-box model compared to existing FL methods, which can be considered as white-box approaches. Recent research on attacking FL methods require the knowledge of either the gradients or the model information, thus rendering BlackFed more resistant to such attacks since gradients or weights are not shared. BlackFed consists of a global server and multiple clients, each possessing their own data. The server is optimized using first order optimization while the client weights are updated using zero order optimization in a round-robin fashion. This introduces the effect of catastrophic forgetting in the network, for which we propose a simple hashmap-based approach. During training, we store the client weights per server so that they can be utilized during inference. With these modifications, our approach demonstrates superior results compared to non-collaborative training and matches the performance of white-box methods, despite being a black-box method itself. Extensive experimentation on the natural and medical domain datasets highlights the effectiveness of BlackFed. Through this endeavor, we aim to propel research towards the development of better privacy preserving federated learning systems. Potential directions for future research can include analysis of other policies for selecting client order during training, and its relation with the disparity in data distribution. One more interesting direction for future research would be the effect of adversarial attacks using generative models on this method. Since masks are shared between the client and server, it would be interesting to check if existing methods are able to recreate client data using them.

Acknowledgments and Disclosure of Funding

This research was supported by a grant from the National Institutes of Health, USA; R01EY033065. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health. The authors have no competing interests in the paper.

References

[1] Acar, D.A.E., Zhao, Y., Matas, R., Mattina, M., Whatmough, P., Saligrama, V.: Federated learning based on dynamic regularization. In: International Conference on Learning Representations (2021), https://openreview.net/forum?id=B7v4QMR6Z9w

[2] Ali, S., Jha, D., Ghatwary, N., Realdon, S., Cannizzaro, R., Salem, O.E., Lamarque, D., Daul, C., Riegler, M.A., Anonsen, K.V., Petlund, A., Halvorsen, P., Rittscher, J., de Lange, T., East, J.E.: A multi-centre polyp detection and segmentation dataset for generalisability assessment. Scientific Data **10**(1), 75 (Feb 2023). https://doi.org/10.1038/s41597-023-01981-y, https://doi.org/10.1038/s41597-023-01981-y

[3] Arivazhagan, M.G., Aggarwal, V., Singh, A.K., Choudhary, S.: Federated learning with personalization layers (2019), https://arxiv.org/abs/1912.00818

[4] Brostow, G.J., Fauqueur, J., Cipolla, R.: Semantic object classes in video: A high-definition ground truth database. Pattern Recognition Letters **30**(2), 88–97 (2009). https://doi.org/https://doi.org/10.1016/j.patrec.2008.04.005, https://www.sciencedirect.com/science/article/pii/S0167865508001220, video-based Object and Event Analysis

[5] Caldarola, D., Caputo, B., Ciccone, M.: Improving generalization in federated learning by seeking flat minima. In: European Conference on Computer Vision. pp. 654–672. Springer (2022)

[6] Chang, Q., Qu, H., Zhang, Y., Sabuncu, M., Chen, C., Zhang, T., Metaxas, D.N.: Synthetic learning: Learn from distributed asynchronized discriminator gan without sharing medical image data. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 13853–13863. IEEE Computer Society, Los Alamitos, CA, USA (jun 2020). https://doi.org/10.1109/CVPR42600.2020.01387, https://doi.ieeecomputersociety.org/10.1109/CVPR42600.2020.01387

[7] Chen, L.C., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation (2017)

[8] Codella, N.C.F., Gutman, D., Celebi, M.E., Helba, B., Marchetti, M.A., Dusza, S.W., Kalloo, A., Liopyris, K., Mishra, N., Kittler, H., Halpern, A.: Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging (isbi), hosted by the international skin imaging collaboration (isic). In: 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018). pp. 168–172 (2018). https://doi.org/10.1109/ISBI.2018.8363547

[9] Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 3213–3223 (2016), https://api.semanticscholar.org/CorpusID:502946

[10] Dayan, I., Roth, H.R., Zhong, A., Harouni, A., Gentili, A., Abidin, A.Z., Liu, A., Costa, A.B., Wood, B.J., Tsai, C.S., Wang, C.H., Hsu, C.N., Lee, C.K., Ruan, P., Xu, D., Wu, D., Huang, E., Kitamura, F.C., Lacey, G., de Antônio Corradi, G.C., Nino, G., Shin, H.H., Obinata, H., Ren, H., Crane, J.C., Tetreault, J., Guan, J., Garrett, J.W., Kaggie, J.D., Park, J.G., Dreyer, K., Juluru, K., Kersten, K., Rockenbach, M.A.B.C., Linguraru, M.G., Haider, M.A., AbdelMaseeh, M., Rieke, N., Damasceno, P.F., e Silva, P.M.C., Wang, P., Xu, S., Kawano, S., Sriswasdi, S., Park, S.Y., Grist, T.M., Buch, V., Jantarabenjakul, W., Wang, W., Tak, W.Y., Li, X., Lin, X., Kwon, Y.J., Quraini, A., Feng, A., Priest, A.N., Turkbey, B., Glicksberg, B., Bizzo, B., Kim, B.S., Tor-Díez, C., Lee, C.C., Hsu, C.J., Lin, C., Lai, C.L., Hess, C.P., Compas, C., Bhatia, D., Oermann, E.K., Leibovitz, E., Sasaki, H., Mori, H., Yang, I., Sohn, J.H., Murthy, K.N.K., Fu, L.C., de Mendonça, M.R.F., Fralick, M., Kang, M.K., Adil, M., Gangai, N., Vateekul, P., Elnajjar, P., Hickman, S., Majumdar, S., McLeod, S.L., Reed, S., Gräf, S., Harmon, S., Kodama, T., Puthanakit, T., Mazzulli, T., de Lavor, V.L., Rakvongthai, Y., Lee, Y.R., Wen, Y., Gilbert, F.J., Flores, M.G., Li, Q.: Federated learning for predicting clinical outcomes in patients with covid-19. Nature Medicine **27**(10), 1735–1743 (Oct 2021). https://doi.org/10.1038/s41591-021-01506-3, `https://doi.org/10.1038/s41591-021-01506-3`

[11] Dimitrov, D.I., Balunović, M., Konstantinov, N., Vechev, M.: Data leakage in federated averaging. In: Transactions on Machine Learning Research (2022), `https://openreview.net/forum?id=e7A0B99zJf`

[12] Dong, J., Wang, L., Fang, Z., Sun, G., Xu, S., Wang, X., Zhu, Q.: Federated class-incremental learning. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10154–10163 (2022). https://doi.org/10.1109/CVPR52688.2022.00992

[13] Dong, N., Voiculescu, I.: Federated contrastive learning for decentralized unlabeled medical images (2021)

[14] Fantauzzo, L., Fanì, E., Caldarola, D., Tavera, A., Cermelli, F., Ciccone, M., Caputo, B.: Feddrive: Generalizing federated learning to semantic segmentation in autonomous driving. In: Proceedings of the 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (2022)

[15] Fowl, L., Geiping, J., Czaja, W., Goldblum, M., Goldstein, T.: Robbing the fed: Directly obtaining private data in federated learning with modified models (2022)

[16] Gao, L., Fu, H., Li, L., Chen, Y., Xu, M., Xu, C.Z.: Feddc: Federated learning with non-iid data via local drift decoupling and correction. In: IEEE Conference on Computer Vision and Pattern Recognition (2022)

[17] Geiping, J., Bauermeister, H., Dröge, H., Moeller, M.: Inverting gradients - how easy is it to break privacy in federated learning? In: Proceedings of the 34th International Conference on Neural Information Processing Systems. NIPS '20, Curran Associates Inc., Red Hook, NY, USA (2020)

[18] Gong, X., Sharma, A., Karanam, S., Wu, Z., Chen, T., Doermann, D., Innanje, A.: Ensemble attention distillation for privacy-preserving federated learning. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 15056–15066 (2021). https://doi.org/10.1109/ICCV48922.2021.01480

[19] Guo, P., Yang, D., Hatamizadeh, A., Xu, A., Xu, Z., Li, W., Zhao, C., Xu, D., Harmon, S., Turkbey, E., Turkbey, B., Wood, B., Patella, F., Stellato, E., Carrafiello, G., Patel, V.M., Roth, H.R.: Auto-fedrl: Federated hyperparameter optimization for multi-institutional medical image segmentation (2022)

[20] Gupta, O., Raskar, R.: Distributed learning of deep neural network over multiple agents. Journal of Network and Computer Applications **116**, 1–8 (08 2018). https://doi.org/10.1016/j.jnca.2018.05.003

[21] Gutman, D., Codella, N.C., Celebi, E., Helba, B., Marchetti, M., Mishra, N., Halpern, A.: Skin lesion analysis toward melanoma detection: A challenge at the international symposium on biomedical imaging (isbi) 2016, hosted by the international skin imaging collaboration (isic). arXiv preprint arXiv:1605.01397 (2016)

[22] Han, S., Park, S., Wu, F., Kim, S., Wu, C., Xie, X., Cha, M.: FedX: Unsupervised Federated Learning with Cross Knowledge Distillation. In: ECCV (2022)

[23] Hitaj, B., Ateniese, G., Perez-Cruz, F.: Deep models under the gan: Information leakage from collaborative deep learning. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. p. 603–618. CCS '17, Association for Computing Machinery, New York, NY, USA (2017). https://doi.org/10.1145/3133956.3134012, https://doi.org/10.1145/3133956.3134012

[24] Hu, M., Zhou, P., Yue, Z., Ling, Z., Huang, Y., Liu, Y., Chen, M.: Fedcross: Towards accurate federated learning via multi-model cross aggregation. ArXiv **abs/2210.08285** (2022), https://api.semanticscholar.org/CorpusID:252917723

[25] Huang, W., Ye, M., Du, B.: Learn from others and be yourself in heterogeneous federated learning. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10133–10143 (2022). https://doi.org/10.1109/CVPR52688.2022.00990

[26] Jeon, J., Kim, J., Lee, K., Oh, S., Ok, J.: Gradient inversion with generative image prior. In: Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J.W. (eds.) Advances in Neural Information Processing Systems (2021), https://openreview.net/forum?id=x9jS8pX3dkx

[27] Jin, X., Chen, P.Y., Hsu, C.Y., Yu, C.M., Chen, T.: Cafe: Catastrophic data leakage in vertical federated learning. ArXiv **abs/2110.15122** (2021), https://api.semanticscholar.org/CorpusID:240070357

[28] Kaissis, G.A., Makowski, M.R., Rückert, D., Braren, R.F.: Secure, privacy-preserving and federated machine learning in medical imaging. Nature Machine Intelligence **2**(6), 305–311 (Jun 2020). https://doi.org/10.1038/s42256-020-0186-1, https://doi.org/10.1038/s42256-020-0186-1

[29] Karimireddy, S.P., Kale, S., Mohri, M., Reddi, S., Stich, S., Suresh, A.T.: SCAFFOLD: Stochastic controlled averaging for federated learning. In: III, H.D., Singh, A. (eds.) Proceedings of the 37th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 119, pp. 5132–5143. PMLR (13–18 Jul 2020), https://proceedings.mlr.press/v119/karimireddy20a.html

[30] Li, Q., He, B., Song, D.: Model-contrastive federated learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2021)

[31] Li, T., Sahu, A.K., Zaheer, M., Sanjabi, M., Talwalkar, A., Smith, V.: Federated optimization in heterogeneous networks (2020)

[32] Li, W., Milletarì, F., Xu, D., Rieke, N., Hancox, J., Zhu, W., Baust, M., Cheng, Y., Ourselin, S., Cardoso, M.J., Feng, A.: Privacy-preserving federated brain tumour segmentation. In: Suk, H.I., Liu, M., Yan, P., Lian, C. (eds.) Machine Learning in Medical Imaging. pp. 133–141. Springer International Publishing, Cham (2019)

[33] Liu, Q., Chen, C., Qin, J., Dou, Q., Heng, P.A.: Feddg: Federated domain generalization on medical image segmentation via episodic learning in continuous frequency space. The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021)

[34] McMahan, B., Moore, E., Ramage, D., Hampson, S., Arcas, B.A.y.: Communication-Efficient Learning of Deep Networks from Decentralized Data. In: Singh, A., Zhu, J. (eds.) Proceedings of the 20th International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research, vol. 54, pp. 1273–1282. PMLR (20–22 Apr 2017), https://proceedings.mlr.press/v54/mcmahan17a.html

[35] Mendieta, M., Yang, T., Wang, p., Lee, M., Ding, Z., Chen, C.: Local learning matters: Rethinking data heterogeneity in federated learning. pp. 8387–8396 (06 2022). https://doi.org/10.1109/CVPR52688.2022.00821

[36] Miao, J., Yang, Z., Fan, L., Yang, Y.: Fedseg: Class-heterogeneous federated learning for semantic segmentation. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 8042–8052 (2023). https://doi.org/10.1109/CVPR52729.2023.00777

[37] Michieli, U., Ozay, M.: Prototype guided federated learning of visual feature representations. ArXiv **abs/2105.08982** (2021), https://api.semanticscholar.org/CorpusID:234777819

[38] Naeem, A., Anees, T., Naqvi, R.A., Loh, W.K.: A comprehensive analysis of recent deep and federated-learning-based methodologies for brain tumor diagnosis. Journal of Personalized Medicine **12**(2) (2022). https://doi.org/10.3390/jpm12020275, https://www.mdpi.com/2075-4426/12/2/275

[39] Oh, C., Hwang, H., Lee, H.y., Lim, Y., Jung, G., Jung, J., Choi, H., Song, K.: Blackvip: Black-box visual prompting for robust transfer learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 24224–24235 (June 2023)

[40] Poirot, M.G., Vepakomma, P., Chang, K., Kalpathy-Cramer, J., Gupta, R., Raskar, R.: Split learning for collaborative deep learning in healthcare (2019)

[41] Reddi, S., Charles, Z.B., Zaheer, M., Garrett, Z., Rush, K., Konečný, J., Kumar, S., McMahan, B. (eds.): Adaptive Federated Optimization (2021), https://openreview.net/forum?id=LkFG3lB13U5

[42] Sheller, M.J., Reina, G.A., Edwards, B., Martin, J., Bakas, S.: Multi-institutional deep learning modeling without sharing patient data: A feasibility study on brain tumor segmentation. In: Crimi, A., Bakas, S., Kuijf, H., Keyvan, F., Reyes, M., van Walsum, T. (eds.) Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries. pp. 92–104. Springer International Publishing, Cham (2019)

[43] Shenaj, D., Fani, E., Toldo, M., Caldarola, D., Tavera, A., Michieli, U., Ciccone, M., Zanuttigh, P., Caputo, B.: Learning across domains and devices: Style-driven source-free domain adaptation in clustered federated learning. In: 2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 444–454. IEEE Computer Society, Los Alamitos, CA, USA (jan 2023). https://doi.org/10.1109/WACV56688.2023.00052, https://doi.ieeecomputersociety.org/10.1109/WACV56688.2023.00052

[44] Tan, Y., Long, G., Liu, L., Zhou, T., Lu, Q., Jiang, J., Zhang, C.: Fedproto: Federated prototype learning across heterogeneous clients. In: AAAI Conference on Artificial Intelligence (2022)

[45] Tschandl, P., Rosendahl, C., Kittler, H.: The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. Scientific Data **5**(1), 180161 (Aug 2018). https://doi.org/10.1038/sdata.2018.161, https://doi.org/10.1038/sdata.2018.161

[46] Valanarasu, J.M.J., Patel, V.M.: Unext: Mlp-based rapid medical image segmentation network. arXiv preprint arXiv:2203.04967 (2022)

[47] Vepakomma, P., Gupta, O., Swedish, T., Raskar, R.: Split learning for health: Distributed deep learning without sharing raw patient data (2018)

[48] Vepakomma, P., Singh, A., Gupta, O., Raskar, R.: Nopeek: Information leakage reduction to share activations in distributed deep learning (2020)

[49] Wang, J., Jin, Y., Wang, L.: Personalizing federated medical image segmentation via local calibration. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds.) Computer Vision – ECCV 2022. pp. 456–472. Springer Nature Switzerland, Cham (2022)

[50] Wu, Y., Zeng, D., Wang, Z., Shi, Y., Hu, J.: Federated contrastive learning for volumetric medical image segmentation. In: de Bruijne, M., Cattin, P.C., Cotin, S., Padoy, N., Speidel, S., Zheng, Y., Essert, C. (eds.) Medical Image Computing and Computer Assisted Intervention – MICCAI 2021. pp. 367–377. Springer International Publishing, Cham (2021)

[51] Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P.: Segformer: Simple and efficient design for semantic segmentation with transformers. In: Neural Information Processing Systems (NeurIPS) (2021)

[52] Xu, A., Li, W., Guo, P., Yang, D., Roth, H., Hatamizadeh, A., Zhao, C., Xu, D., Huang, H., Xu, Z.: Closing the generalization gap of cross-silo federated medical image segmentation. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 20834–20843. IEEE Computer Society, Los Alamitos, CA, USA (jun 2022). https://doi.org/10.1109/CVPR52688.2022.02020, https://doi.ieeecomputersociety.org/10.1109/CVPR52688.2022.02020

[53] Yao, C., Gong, B., Qi, H., Cui, Y., Zhu, Y., Yang, M.: Federated multi-target domain adaptation. In: 2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 1081–1090. IEEE Computer Society, Los Alamitos, CA, USA (jan 2022). https://doi.org/10.1109/WACV51458.2022.00115, https://doi.ieeecomputersociety.org/10.1109/WACV51458.2022.00115

[54] Zhang, H., Hong, J., Deng, Y., Mahdavi, M., Zhou, J.: Understanding deep gradient leakage via inversion influence functions. In: Oh, A., Neumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S. (eds.) Advances in Neural Information Processing Systems. vol. 36, pp. 3921–3944. Curran Associates, Inc. (2023), https://proceedings.neurips.cc/paper_files/paper/2023/file/0c4dd7e3d9f528f0b4f2aca9fbcdca8d-Paper-Conference.pdf

[55] Zhang, L., Shen, L., Ding, L., Tao, D., Duan, L.Y.: Fine-tuning global model via data-free knowledge distillation for non-iid federated learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10174–10183 (2022)

[56] Zhang, M., Sapra, K., Fidler, S., Yeung, S., Alvarez, J.M.: Personalized federated learning with first order model optimization. arXiv preprint arXiv:2012.08565 (2020)

[57] Zhao, B., Mopuri, K.R., Bilen, H.: idlg: Improved deep leakage from gradients. arXiv preprint arXiv:2001.02610 (2020)

[58] Zhu, J., Blaschko, M.: R-gap: Recursive gradient attack on privacy (2021)

Table 7: Data counts for CAMVID, ISIC and Polypgen datasets

	CAMVID				ISIC			Polypgen					
Split	C1	C2	C3	C4	C1	C2	C3	C1	C2	C3	C4	C5	C6
Train	24	61	181	103	6009	720	2000	153	180	274	136	124	52
Val	8	16	50	26	2003	180	150	51	60	91	45	42	18
Test	92	24	74	42	2003	379	600	52	61	92	46	42	18

Table 8: Data counts for Cityscapes dataset

	Cityscapes																	
Split	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18
Train	121	67	221	107	59	154	76	173	137	83	69	65	255	137	100	66	99	85
Val	27	15	48	24	13	34	17	38	30	18	15	15	55	30	22	15	22	19
Test	26	14	47	23	13	33	16	37	29	18	15	14	55	29	22	14	21	18

A Centerwise Dataset Information

In this section, we describe the number of data points in the training, validation and testing splits of each center of each dataset. The center represents the client in FL, where each center is in possession of data that cannot be shared directly with the other centers or to the public. CAMVID [4] has 4 centers, which are described in Table 7. Cityscapes [9] has 18 centers, which are described in Table 8. These centers represent the different cities from which the data is collected for these datasets. ISIC [8, 21, 45] has 3 centers and Polypgen [2]has 4 centers, which represent the different hospitals from which the data is collected. These results are present in Table 7.

B Centerwise Performance for Cityscapes Dataset

While the average mIoU is presented in the main paper, we also present the centerwise performance of DeepLabv3-based server for the Cityscapes dataset in Tables 9 and 10. It can be seen from the tables that individual training performs poorly, the reason being that each of the centers has limited amount of data. In contrast, our method makes use of data from all clients to improve performance. Here, we also see that BlackFed v2 performs better than v1 in all cases, thus showing the effectiveness of the server hashmap in countering catastrophic forgetting. With our approach, the performance of the model reaches close to the white-box methods, but without sharing any gradients or model information.

Table 9: mIoU scores for BlackFed v1 and v2 in comparison with individual and FL-based training strategies for Cityscapes. "Local" represents test data from the center. "OOD" represents mean mIoU on test data from rest of the centers. For FedAvg and Combined Training, just one model is trained. Hence, its performance is noted only in each of the local test datasets.

	C1		C2		C3		C4		C5		C6		C7		C8		C9	
Method	Local	OOD	Local	OOD	Local	OOD	Local	OOD	Local	OOD	Local	OOD	Local	OOD	Local	OOD	Local	OOD
Individual	0.61	0.57	0.61	0.58	0.64	0.52	0.55	0.50	0.35	0.39	0.39	0.39	0.51	0.39	0.33	0.39	0.55	0.52
BlackFed v1	0.76	0.72	0.67	0.71	0.77	0.71	0.75	0.71	0.66	0.70	0.74	0.71	0.71	0.71	0.66	0.72	0.70	0.70
BlackFed v2	**0.78**	**0.74**	**0.71**	**0.74**	**0.81**	**0.74**	**0.77**	**0.74**	**0.70**	**0.73**	**0.76**	**0.74**	**0.75**	**0.73**	**0.66**	**0.75**	**0.71**	**0.72**
Combined Training	0.82	-	0.71	-	0.82	-	0.78	-	0.74	-	0.78	-	0.76	-	0.67	-	0.74	-
White-box Training	0.79	0.74	0.71	0.75	0.81	0.75	0.78	0.75	0.70	0.74	0.78	0.75	0.74	0.74	0.66	0.76	0.73	0.76
FedAvg [34]	0.85	-	0.75	-	0.84	-	0.80	-	0.74	-	0.80	-	0.78	-	0.70	-	0.76	-

Table 10: mIoU scores for BlackFed v1 and v2 in comparison with individual and FL-based training strategies for Cityscapes. "Local" represents test data from the center. "OOD" represents mean mIoU on test data from rest of the centers. For FedAvg and Combined Training, just one model is trained. Hence, its performance is noted only in each of the local test datasets.

	C10		C11		C12		C13		C14		C15		C16		C17		C18	
Method	Local	OOD	Local	OOD	Local	OOD	Local	OOD	Local	OOD	Local	OOD	Local	OOD	Local	OOD	Local	OOD
Individual	0.49	0.45	0.62	0.57	0.58	0.54	0.38	0.39	0.41	0.39	0.56	0.50	0.38	0.39	0.66	0.55	0.37	0.39
BlackFed v1	0.69	0.71	0.74	0.70	0.68	0.71	0.68	0.71	0.74	0.69	0.73	0.68	0.65	0.66	0.69	0.70	0.72	0.70
BlackFed v2	**0.72**	**0.73**	**0.77**	**0.73**	**0.71**	**0.73**	**0.72**	**0.74**	**0.78**	**0.73**	**0.78**	**0.74**	**0.74**	**0.73**	**0.80**	**0.73**	**0.76**	**0.74**
Combined Training	0.74	-	0.80	-	0.74	-	0.73	-	0.81	-	0.79	-	0.77	-	0.82	-	0.78	-
White-box Training	0.73	0.76	0.79	0.75	0.72	0.75	0.71	0.76	0.78	0.75	0.79	0.75	0.72	0.74	0.81	0.75	0.78	0.75
FedAvg [34]	0.76	-	0.80	-	0.75	-	0.74	-	0.82	-	0.83	-	0.77	-	0.84	-	0.81	-

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The paper introduces a black-box based approach for Federated Learning (FL) using a split-nn formulation that does not require gradient and model info sharing. Section 3 defines the approach and Section 4 shows experiments using the approach.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We mention further directions of improvement and future work in the conclusion section.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.

- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: We have provided the complete mathematical formulation and the algorithm, but theoretical claims are not made.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We will be releasing the code, data splits and pretrained models after review. The algorithm for the approach is included in the paper. The experimental settings are also provided in the main paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example

(a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.

(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The code and data splits will be released on Github post review.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: While the code will be released along with pretrained models, we have mentioned the experimental settings in the paper. We have also mentioned the data splits in the supplementary.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: All results of BlackFed have a p-value less than 0.001, showing statistical significance.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Provided in the experiment setup

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: all guidelines followed.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.

- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Introduction discusses about our method's positive impacts. There are no apparent negative impacts.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: No risk posed by the work.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

Justification: We use public datasets that have been cited in the paper.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Code will be released post review and the novel algorithm is described in the paper.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: No crowdsourcing involved.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The study does not involve crowdsourcing or human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.