Federated Split Task-Agnostic Vision Transformer for COVID-19 CXR Diagnosis

Anonymous Author(s)
Affiliation
Address
email

Abstract

Federated learning, which shares the weights of the neural network across clients, is gaining attention in healthcare sector as it enables training on a large corpus of decentralized data while maintaining data privacy. For example, this enables a neural network training for COVID-19 CXR diagnosis without collecting patient CXR data across multiple hospitals. Unfortunately, the exchange of the weights quickly consumes the network bandwidth if highly expressive network architecture is employed. So-called split learning partially solves this problem by dividing a neural network into a client and a server part, so that the client part of the network takes up less extensive computation resources and bandwidth. However, it is not clear how to find the optimal split without sacrificing the overall network performance. To amalgamate these methods and thereby maximize their distinct strengths, here we show that the Vision Transformer, a recently developed deep learning architecture with straightforward decomposable configuration, is ideally suitable for the split learning without sacrificing the performance. Even under the extreme non-independent and non-identically distributed data distribution (non-IID) which emulating a realistic collaboration between hospitals using chest X-ray datasets from multiple sources, the proposed framework was able to attain performance comparable to data-centralized training. In addition, the proposed framework along with heterogeneous multi-task clients also improves individual task performances including the diagnosis of COVID-19, eliminating the needs for sharing large weights with innumerable parameters. Our results affirms the suitability of Transformer for collaborative learning in medical imaging, and paves the way forward for future real-world implementations.

1 Introduction

After its earlier success in many fields, deep neural network has found a pervasive suite of application in healthcare research including medical imaging, becoming a new de facto standard [49, 18, 12, 44, 16, 57, 5, 52, 23]. Training these network requires vast amount of data to achieve robust performance [7, 10, 43]. Despite the fact that multi-center collaboration is mandatory due to the shortage of labeled data in a single institution, collaboration in healthcare research is heavily impeded by difficulties in data sharing stemming from the privacy issues and limited consent of patients [45, 35, 48].

To alleviate this problem, the distributed machine learning methods, devised to enable the computation on multiple clients and servers leaving data reside on the source devices, can be effectively leveraged for healthcare research [6, 39]. Federated learning (FL) is a one of these methods which enables model training on a large corpus of decentralized data [25, 31, 53]. However, FL still holds several limitations in that it depend on clients' computational resources for its client-side parallel computation strategy for update and is not free from privacy concern [26, 30, 47]. In contrast to FL, another

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

distributed machine learning method, split learning (SL) offers better privacy and requires lower computational resources of clients by splitting the network between clients and the server [21, 47], but still possess problems that it shows significant slower convergence than FL and can not learn under extreme non-independent and non-identically distributed (non-IID) data [19].

Especially under unprecedent pandemic of emerging pathogen like COVID-19, under which direct multi-national collaboration is deterred for prevention of epidemics, the collaboration via these distributed machine learning approaches is becoming increasingly important, since these enables to build a model with performance tantamount to data-centralized learning without any direct sharing of raw data between institutions to offer privacy.

Recently proposed Vision Transformer (ViT) architecture [13], inspired by astounding results of Transformer-based models on natural language processing (NLP), have demonstrated impeccable performance on many vision tasks by enabling to model long dependencies within images. Beside this strength, the straightforward design of Transformer allows to easily decompose the entire network into parts: the head for extracting features from input image, the Transformer body to model the dependency between features, and the tail used for mapping features to task-specific output. One of the important contributions in this paper is the observation that this configuration is optimal for split learning where a network should be split into the parts for clients and for server. In addition, as suggested in [8], the Transformer body with sufficient capacity can be shared between various tasks, being suitable for multi-task learning (MTL) to leverage robust representation from multiple related tasks to enhance the generalization performance of individual tasks.

Accordingly, here we propose a novel Federated Split Task-Agnostic (FeSTA) framework equipped with Transformer to simultaneously process multiple chest X-ray (CXR) tasks including diagnosis of COVID-19, emulating a realistic collaboration between several hospitals. To validate the practicability of FeSTA, we also implemented the framework using Friendly Federated Learning Framework (Flower) protocol [2], confirming seamless integration of various components. Experimental results show that our framework can show stable performance even under extreme non-IID setting which is frequently faced situation for collaboration between hospitals while offering privacy, by amalgamating FL and SL to maximally exploit their main advantages. In addition, we show that the proposed FeSTA Transformer along with MTL improves the performances of individual tasks. In summary, our contributions are two folds:

- We proposed a novel FeSTA learning framework equipped with Transformer by utilizing its decomposable design to amalgamate the merit of FL and SL.
- We showed that the model trained with FeSTA framework can leverage the robust representations from multiple related tasks to improve the performance of individual task.

2 Related Work

Distributed Machine Learning. Federated learning (FL) is a distributed machine learning approach originally proposed by Google [25] to enable training of a model via distributed devices and data. It eliminates needs to aggregate the raw data in a centralized way by enabling the model to be updated on the edge devices (e.g. mobile phone, computers in hospital). Specifically, during the training, the server initializes a global model and send it to each client. The clients then train the model with their local data in parallel, and return the updated model to the server. Then, the server aggregates and distributes those model by method such as `FedAvg` [31] to update the global model. This processes (called round) continue repeatedly until the model converges. Though FL enables decentralized training in privacy-preserving manner, it still holds limitations as it largely depends on computation resources of clients and is vulnerable to model inversion attack [51].

Different from FL, split learning (SL) divides the neural network into several sub-networks, and these separated sub-networks are trained under distributed setting [15]. In detail, the first sub-network is trained on client side with local data and then pass the feature to the second sub-network located in server. The server can access only feature from the first sub-network, and train the second sub-network to send the subsequent feature to third sub-network on client. Finally, the third sub-network is able to provide the output of overall network. By inserting black-box sub-networks in both client and server sides, it is possible to offer better privacy than FL. Besides the privacy benefit, SL uses less computational resources than FL at the client sides. Nevertheless, since the one cycle of forward and

backward passes is finished after data and gradients move back and forth across the sub-networks distributed on multiple sides, the convergence of SL is considerably slower than FL. In addition, it was reported that the convergence is not reached at all under extreme non-IID setting [19].

Multi-Task Learning. Multi-task learning (MTL) is a learning strategy to improve the generalization performance of a specific task with the help of information from other related tasks. In MTL, models for multiple related tasks are trained simultaneously [4, 15]. In its early era, the motivation of multi-task learning is to mitigate the data insufficiency problem where number of labeled data is limited for each task to train an accurate learner. MTL helps reusing knowledge and thereby reduce the requirement for labeled data for each task, exhibiting that MTL model can achieve better performance than single-task counterpart in many fields ranging from computer vision [14, 32, 34, 27] to NLP [29, 20, 28, 38]. With increased data, the MTL model can learn more robust representations via knowledge sharing among multiple tasks, resulting in improved performance and less overfitting for individual task.

Vision Transformer. Transformer [46], which was originally developed for NLP, is a deep neural network based on attention mechanism that utilizes appreciably large receptive field. After achieving state-of-the-art (SOTA) performance in NLP, it has also inspired vision community to explore its application on vision tasks to utilize its ability to model long-range dependency within image [24]. The Vision Transformer (ViT) was one of the successful attempts to apply Transformer directly to images, achieving excellent results compared to the SOTA convolutional neural networks in image classification tasks [13]. Furthermore, in addition to the superb performance, the straightforward modular design of ViT facilitates broad applications in many tasks only with minimal change. As suggested in [8], overall ViT can split into the head, body and tail, and Transformer body with sufficient capacity can be shared across multiple tasks once the task-specific heads and tails are altered to be tailored for individual tasks.

Recently, ViT was successfully used for diagnosis and severity prediction of COVID-19, showing the SOTA performance [33]. Specifically, to alleviate the overfitting problem with limited data available, the overall framework is decomposed into two steps: the pre-trained backbone network to classify common low-level CXR features, which was leveraged in the second step by Transformer for high-level diagnosis and severity prediction of COVID-19. By maximally utilizing the merit of large-scale database containing more than 220,000 CXR images, the model has attained stable generalization performance as well as SOTA performance in a variety of external test data from different institutions, even with limited number of labeled cases for COVID-19.

3 Split Task-Agnostic Transformer for CXR COVID-19 Diagnosis

3.1 Network Architecture

Inspired by these works, here we are interested in utilizing ViT for distributed learning in COVID-19 CXR diagnosis, where the collaboration via these distributed machine learning approaches is becoming increasingly important by offering privacy and still allowing similar performance to the data-centralized learning.

The reason we are interested in ViT architecture for this purpose is because the natural configuration of ViT may be optimal for MTL as well as SL where easily decomposable modular design of network is preferred, suggesting a possibility to maximally reconcile the merits of MTL and SL through ViT architecture. Specifically, the clients just train the head and tail parts of the network, whereas the Transformer body is shared across multiple clients. Then, the embedded features from the head network from multiple clients can be leveraged in the second step by Transformer to process individual task including diagnosis of COVID-19. Furthermore, by maximally utilizing the merit of large-scale database from CXR for various tasks, our goal is to demonstrate the stable generalization performance as well as SOTA performance in the external test dataset.

Fig. 1(a) illustrated the proposed split task-agnostic Transformer for CXR. Let $\mathcal{C} = \bigcup_{k=1}^{K} C_k$ be a group of client sets with different CXR tasks, where K denotes the number of tasks and C_k has one or more clients with different datasets each other for the k-th task, i.e. $C_k = \{c_1^k, c_2^k, \ldots, c_{N_k}^k : N_k \geq 1\}$. Each client $c \in C_k$ has task-specific own network architecture for a head $\mathcal{H}_c$ and a tail $\mathcal{T}_c$, which are connected to Transformer $\mathcal{B}$ in the server.

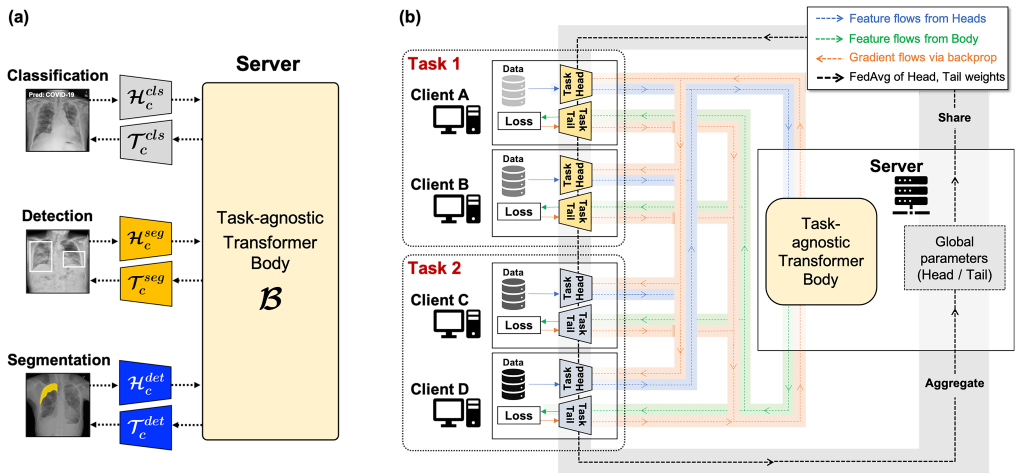

Figure 1: Overall framework of the proposed method. (a) Split task-agnostic CXR Transformer for multi-task learning and (b) the concept of FESTA learning process.

3.2 FESTA: Federated Split Task-Agnostic Learning

The concept of the proposed Federated Split Task-Agnostic (FESTA) learning process is illustrated in Fig 1(a)(b). First, the server initializes the weights of Transformer body and task-specific heads, tails for each task k. Then, it distributes the initialized weights of heads and tails to each client $c \in C_k$. For round $i = 1, 2, \ldots R$, each client (e.g. hospital) perform the forward propagation on their task-specific head and pass their intermediate feature to the server. Specifically, using the local training data $\{(x_c^{(i)}, y_c^{(i)})\}_{i=1}^{N_c}$, the head network $\mathcal{H}_c$ encodes the smashed feature maps $h_c^{(i)}$ and sent it to the server: $h_c^{(i)} = \mathcal{H}_c(x_c^{(i)})$. Then, the server-side Transformer body $\mathcal{B}$ receive the feature from the all clients and generate the features $b_c^{(i)}$ for c in parallel: $b_c^{(i)} = \mathcal{B}(h_c^{(i)})$. Resulting smashed features from the Transformer are allocated to task-specific tail to produce the final prediction $\hat{y}_c^{(i)} = \mathcal{T}_c(b_c^{(i)})$ according to each task, and the forward path finishes. Subsequently, the loss for each client can be calculated as $\ell_c(y_c^{(i)}, \mathcal{T}_c(\mathcal{B}(\mathcal{H}_c(x_c^{(i)}))))$ where $\ell_c(y, \hat{y})$ refers to the c-client specific loss between the target y and the estimate $\hat{y}$. By minimizing the loss with respect to the tail weight, gradient of the local tails are passed reversely to the server. After receiving the gradients, the server performs the back-propagation on the server-side body model, send the gradients back to the clients.

Specifically, for the task-agnostic body update, the following optimization problem is solved:

$$\min_{\mathcal{B}} \sum_{c \in \mathcal{C}} \sum_{i=1}^{N_c} \ell_c(y_c^{(i)}, \mathcal{T}_c(\mathcal{B}(\mathcal{H}_c(x_c^{(i)})))), \tag{1}$$

For the task-specific fine-tuning, the following optimization problem is solved:

$$\min_{\mathcal{H}_c, \mathcal{T}_c} \sum_{i=1}^{N_c} \ell_c(y_c^{(i)}, \mathcal{T}_c(\mathcal{B}(\mathcal{H}_c(x_c^{(i)})))), \tag{2}$$

Finally, the server aggregates and averages the weights of local heads and tails from the clients to update the global heads and tails among the clients for same tasks via `FedAvg`, and distributes back the updated global weights of heads and tails for each task k to the clients. The algorithm is formally presented in Algorithm 1.

3.3 Multi-Task CXR Learning

For synergistic performance improvement, we explore the following three tasks that are commonly used for CXR: classification, segmentation, and object detection. These three tasks were separately trained for individual model, while used simultaneously to train and to evaluate the task-agnostic model for multiple related tasks. The details of each task and dataset are as follows:

Algorithm 1: FESTA: Federated Split Task-Agnostic learning

Function `ServerMain`**:**
  Initialize the body weight $w_{\mathcal{B}}^{(1)}$ and client head/tail weights $(\bar{w}_{\mathcal{H},k}, \bar{w}_{\mathcal{T},k})$ for each task $k \in \{1, ..., K\}$ in server
  for rounds $i = 1, 2, \ldots R$ **do**
    for tasks $k \in \{1, 2, \ldots K\}$ **do in parallel**
      for clients $c \in C_k$ **do in parallel**
        if $i = 1$ **or** $i \in$ UnifyingRounds **then**
          Set client $(w_{\mathcal{H}_c}^{(i)}, w_{\mathcal{T}_c}^{(i)}) \leftarrow (\bar{w}_{\mathcal{H},k}, \bar{w}_{\mathcal{T},k})$
        $h_c^{(i)} \leftarrow$ `ClientHead`(c)
        $b_c^{(i)} \leftarrow \mathcal{B}(h_c^{(i)})$
        $\frac{\partial L_c^{(i)}}{\partial b_c^{(i)}} \leftarrow$ `ClientTail`$(c, b_c^{(i)})$ & Backprop.
        $(w_{\mathcal{H}_c}^{(i+1)}, w_{\mathcal{T}_c}^{(i+1)}) \leftarrow$ `ClientUpdate`$(c, \frac{\partial L_c^{(i)}}{\partial h_c^{(i)}})$
    Update body $w_{\mathcal{B}}^{(i+1)} \leftarrow w_{\mathcal{B}}^{(i)} - \frac{\eta}{K} \sum_{k=1}^{K} \sum_{c \in C_k} \frac{\partial L_c^{(i)}}{N_k \partial w_{\mathcal{B}}^{(i)}}$
    if $i \in$ UnifyingRounds **then**
      for tasks $k \in \{1, 2, \ldots K\}$ **do**
        Update $(\bar{w}_{\mathcal{H},k}, \bar{w}_{\mathcal{T},k}) \leftarrow (\frac{1}{N_k} \sum_{c \in C_k} w_{\mathcal{H}_c}^{(i+1)}, \frac{1}{N_k} \sum_{c \in C_k} w_{\mathcal{T}_c}^{(i+1)})$

Function `ClientHead`(c)**:**
  $x_c \leftarrow$ Current batch of input from client c
  return $\mathcal{H}_c(x_c)$
Function `ClientTail`(c, b_c)**:**
  $y_c \leftarrow$ Current batch of label from client c
  $L_c \leftarrow \ell_c(y_c, \mathcal{T}_c(b_c))$ & Backprop.
  return $\frac{\partial L_c}{\partial b_c}$
Function `ClientUpdate`$(c, \frac{\partial L_c}{\partial h_c})$**:**
  Backprop. & $(w_{\mathcal{H}_c},\ w_{\mathcal{T}_c}) \leftarrow (w_{\mathcal{H}_c} - \eta \frac{\partial L_c}{\partial w_{\mathcal{H}_c}},\ w_{\mathcal{T}_c} - \eta \frac{\partial L_c}{\partial w_{\mathcal{T}_c}})$
  return $(w_{\mathcal{H}_c}, w_{\mathcal{T}_c})$

COVID-19 Classification. This is the main task we want to achieve through FESTA. Since we initialized the weights of classification heads to be the robust feature extractor trained on pre-built large data corpus containing common CXR findings, the classification heads were initialized with the pre-trained weights from CheXpert [22] dataset containing 10 CXR findings (no finding, cardiomegaly, opacity, edema, consolidation, pneumonia, atelectasis, pneumothorax, pleural effusion, support device) labeled by experts. With a total of 224,316 CXR images from 65,240 subjects, 29,420 PA and 161,427 AP view data were available after excluding the 32,387 lateral view images. Table 1 summarizes dataset resources and partitioning for classification task. We used both public datasets containing labels of infectious disease (Valencian Region Medical Image Bank [BIMCV] [11], Brixia [40, 3], National Institutes of Health [NIH] [50]) and CXR data deliberately collected from four hospitals labeled by board-certified radiologists. We put one hospital data aside as an external test dataset to evaluate the classification performance for real-world application. Overall, 17,138 CXR images were used for training/validation and 365 CXR images for test. We modeled data from different sources as individual clients, emulating the real-world collaboration between hospitals. This setting is important to validate our method in non-IID data distribution. The study was ethically approved by the Institutional Review Board at each participating hospital and the requirement for informed consent was waived.

Segmentation. For segmentation task, we have used the Society for Imaging Informatics in Medicine (SIIM) and American College of Radiology (ACR) Pneumothorax Segmentation Challenge

Table 1: Datasets and sources for COVID-19 diagnosis

	Total CXR images	External	Training and validation dataset					
			Client 1	Client 2	Client 3	Client 4	Client 5	Client 6
		Hospital 1	Hospital 2	Hospital 3	Hospital 4	NIH	Brixia	BIMCV
Normal	13,649	320	300	400	8,861	3,768	-	-
Other infection	1,468	39	144	308	977	-	-	-
COVID-19	2,431	6	8	80	-	-	1,929	408
Total CXR	**17,548**	**365**	**452**	**788**	**9,838**	**3,768**	**1,929**	**408**

dataset [41] consisting of 12,047 CXR images for training, 3,205 images for testing. The training dataset was divided randomly with 4:1 ratio into training and validation datasets, and we developed and fine-tuned the model for pneumothorax segmentation with these datasets. Afterward, the segmentation performance of the model was evaluated in testing datasets. Since the data for segmentation was anonymized, it was not possible to divide the dataset according to the sources of acquisition. Instead, we randomly divided the entire dataset into the two clients, emulating the collaboration between two hospitals.

Object Detection. For object detection task, the model was constructed to detect lung opacities in CXR with Radiological Society of North America (RSNA) Pneumonia Detection Challenge dataset [37] that consists of CXR images and labels with bounding box and detailed class information of 26,684 subjects. We randomly divided the entire dataset with 3:1 ratio into training and testing dataset, with which model was trained and evaluated. Similar to the segmentation task, as the data contains no information about the sources of acquisition, the dataset was randomly divided into two clients to simulate the collaboration between hospitals.

4 Experimental Results

4.1 Implementation Details

For the head and tail parts of our model, the networks specialized for each task was utilized. For classification, we used the modified version of the network proposed by Ye et al. [54], which comprises DenseNet combined with PCAM operations for classification task, since it achieved outstanding performance in CXR classification competition. For segmentation, we adopts TransUNet [9] that inserts Transformer between CNN encoder and decoder of UNet [36] to take the advantages of both architectures. Similarly, the 2nd place solution in RSNA pneumonia detection challenge, which is a modified version of RetinaNet for object detection, was utilized. After splitting these models into head and tail, the feature maps between head and tail were mapped to the feature maps with same dimension of $16 \times 16 \times 768$, and used as input of Transformer body. The Transformer body, equipped with 12 layers of standard Transformer encoder with 12 attention heads, transforms the feature maps to embed better representation. Then, the resulting feature maps from body were passed to tails to yield the final outcomes.

As suggested in Park et al. [33], the head for classification was first initialized with pre-trained weights from CheXpert dataset. We minimized the cross-entropy loss for the classification task. For segmentation model, we minimized the binary cross-entropy loss combined with dice and focal loss. Finally, for detection task, we minimized the sum of box classification, box regression, and image classification losses as suggested by Gabruseva et al. [17]. For all task, the batch size was 2 per client, and warm up step was 500. We set the number of total rounds to 12,000, and the weights of each clients' head and tail underwent `FedAvg` per 100 rounds by the server.

Fig. 2 illustrates the implementation details and the experimental settings of the proposed method. To simulate single-task learning in Fig. 2(b), we considered each six data sources as different clients for classification task. On the other hand, for multi-task learning shown in Fig. 2(c), 10 clients were simultaneously used, with six clients for classification and two clients for segmentation and detection tasks, respectively. For the segmentation and detection tasks, the training data set was randomly split into two subsets, whereas non-IID distribution was used for classification task. To adjust the loss scale, the customized weights of 1:2:2 were applied for classification, segmentation, and detection tasks to update the common body weights. We divided the multi-task learning into two steps, jointly

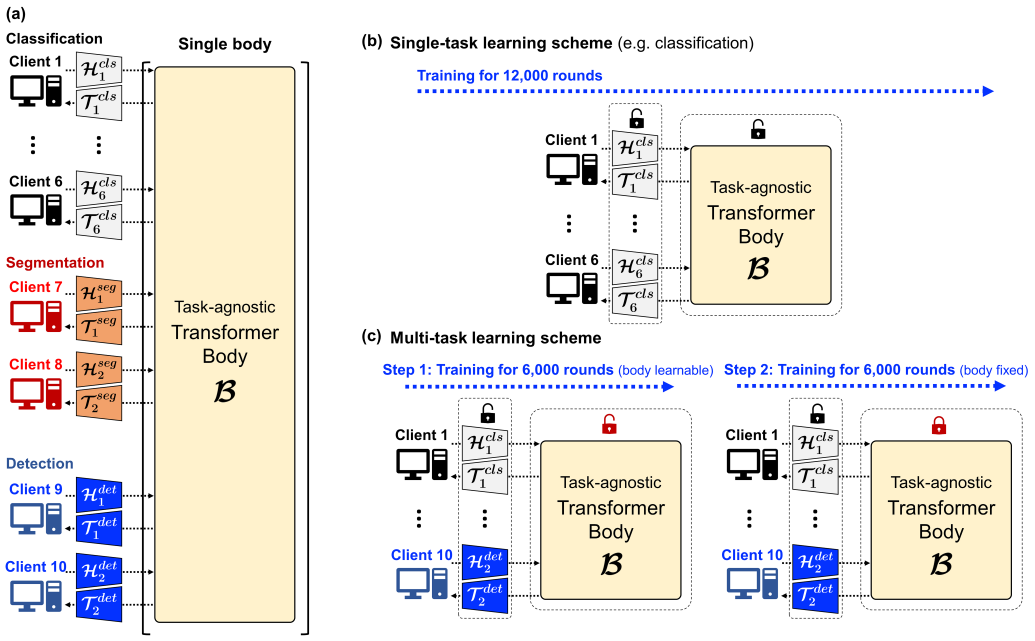

Figure 2: Implementation details of the proposed method. (a) Detailed experimental setting for multi-task learning (MTL) with three different tasks composed of 10 clients (six for classification, two for segmentation, two for detection. Training scheme for (b) single-task learning, and (c) MTL.

training the task-specific heads, tails and the task-agnostic body (6,000 rounds), and fine-tuning only the task-specific heads, tails with the body weights fixed (6,000 rounds).

The FeSTA, federate learning and split learning simulation was performed on the modified version of Flower (licensed by under a Apache-2.0 license) [2] federated learning framework. All experiments were performed with Python version 3.8 and Pytorch version 1.7 on Nvidia RTX 3090, 2080 Ti and 1080 Ti. For more details of implementation, refer to Appendix A.

Performance Metrics. We used the area under receiver operating characteristic curve (AUC) to evaluate the diagnostic performance in the classification task. To evaluate the accuracy of segmentation, the Dice coefficient was used to quantitatively measure the overlap between the segmentation results by model and the ground truths. The detection results was evaluated by calculating the mean average precision (mAP) at different intersection over union, with threshold range from 0.4 to 0.75 with step size of 0.05 as suggested in RSNA [37]. All experiments have been run and evaluated with three different random seeds for the weight initialization to prevent chance from confusing the results.

4.2 Results

FeSTA vs. Other Strategies. We compared the performance of classification model trained with FeSTA with a data-centralized setting and other distributed learning strategies for COVID-19 classification task under the extreme non-IID setting. To simulated data-centralized training, the whole network with connected head, body and tail was trained with the integrated dataset of all six sources. To simulated FL, the whole network was aggregated and distributed by the server with `FedAvg` as suggested by McMahan et al. [31]. On the other hand, for SL, the split sub-networks reside in clients and server side, and the clients side sub-networks were not aggregated as in [47]. The same experimental settings and hyperparameters were applied for fair comparison. For more details of the data-centralized learning and other distributed learning methods, refer to Appendix B.

As shown in Table 2, our method achieved comparable performance to data-centralized learning method as well as outperformed the existing distributed learning methods, suggesting the superiority of our method over other methods.

Table 2: Comparison for performance of the proposed method with other strategies

Strategy	AUC			
	Average	**COVID-19**	**Others**	**Normal**
Data-centralized	0.911 ± 0.016	0.883 ± 0.036	0.927 ± 0.013	0.923 ± 0.004
Federated learning	0.891 ± 0.019	0.840 ± 0.035	0.926 ± 0.018	0.906 ± 0.028
Split learning	0.863 ± 0.005	0.807 ± 0.012	0.892 ± 0.007	0.889 ± 0.019
FESTA (ours)	$\mathbf{0.909 \pm 0.021}$	$\mathbf{0.880 \pm 0.008}$	$\mathbf{0.916 \pm 0.038}$	$\mathbf{0.931 \pm 0.021}$

Note: Experiments were performed repeatedly with three random seeds to report mean and standard deviation. For evaluation of split learning, the average metric between clients are calculated.

Multi-Task Learning vs. Single-Task Learning. We evaluate whether the task-agnostic Transformer body contributes to improve the performance of the entire model by leveraging better representation from the MTL with several related tasks, namely classification, segmentation, and object detection in this work. As provided in Table 3, the models trained with the MTL approach showed at least comparable or even better performance compared with single-task learning counterparts without need to create an individual body model for each task or sharing a large body model between clients, which suggest the distinct merit of MTL approach with our framework.

Table 3: Comparison for performances between single-task and multi-task learning

Tasks	Metrics	Single-task learning	Multi-task learning
Classification	AUC	0.909 ± 0.021	$\mathbf{0.931 \pm 0.004}$
Segmentation	Dice	0.798 ± 0.016	$\mathbf{0.821 \pm 0.003}$
Detection	mAP	0.202 ± 0.008	$\mathbf{0.204 \pm 0.002}$

Note: Experiments were performed repeatedly with three random seeds to report mean and standard deviation.

Model Sizes and Communicative Benefit. Table 4 shows the numbers of parameters and the sizes of sub-networks. The task-agnostic body is the largest among sub-networks in terms of both the number of parameter and model size. This suggest that the largest part of the model does not need to be aggregated and distributed between client-client and client-server, which offers substantial communicative benefit. In all tasks, the size of the body is more than half of the entire network. Therefore, not having to share this huge part means the total training time can be reduced considerably. In our experiments, the training time with the proposed method was four times shorter than the FL approach in which the entire network is shared and distributed. In addition, the proposed method offers another benefit of saving the computational resources of the server by processing multiple tasks with single body model to enable the server to efficiently handle the requests for various tasks from many clients simultaneously.

Table 4: Parameter numbers and model sizes of the sub-networks

Task	Head		Body		Tail	
	Parameters	**Size**	**Parameters**	**Size**	**Parameters**	**Size**
Classification	13.313 M	54.1 MB	66.367 M	265.5 MB	0.002 M	11.7 KB
Segmentation	15.041 M	60.2 MB			7.387 M	29.6 MB
Detection	27.085 M	108.8 MB			19.773 M	79.1 MB

Note: Model sizes were estimated by parameter numbers and file sizes of saved weights.

4.3 Ablation Study

Role of Transformer Body. We performed the ablation study to verify the contribution of the Transformer body on the server. The model without the server-side Transformer body was constructed and trained with the same setting, and compared with the proposed model containing Transformer body. As provided in Table 5, the mean AUC values were higher in the model with Transformer body, implying the advantageous role of the server-side Transformer body.

Numbers of Round for Averaging. Determining the optimal number of rounds for averaging is important. If the aggregation, averaging, and distribution, namely `FedAvg` is performed too frequently, the cost and required resources for communication increases, which can interfere or even preclude the learning process. On the contrary, if averaged too rarely, naive averaging of the learned model parameters of local learner can be devastating, resulting in nonsensical parameters in some layer [55]. Therefore, we performed the ablation to determine the optimal number of round to conduct `FedAvg`. As shown in Table 5, the averaging per 100 rounds showed comparable or better performance to less or more frequent counterparts.

More results of ablation studies are provided in Appendix C.

Table 5: Role of Transformer body and round number for `FedAvg`

Method	AUC			
	Average	**COVID-19**	**Others**	**Normal**
Role of Transformer body				
w/o Transformer body	0.889 ±0.015	0.874 ±0.056	0.895 ±0.011	0.898 ±0.009
w Transformer body	**0.909 ± 0.021**	**0.880 ± 0.008**	**0.916 ± 0.038**	**0.931 ± 0.021**
Number of rounds for `FedAvg`				
per 10 rounds	0.822 ± 0.023	0.724 ± 0.053	0.884 ± 0.017	0.858 ± 0.035
per 100 rounds	**0.909 ± 0.021**	**0.880 ± 0.008**	**0.916 ± 0.038**	**0.931 ± 0.021**
per 1000 rounds	0.903 ± 0.012	0.905 ± 0.019	0.866 ± 0.034	0.939 ± 0.005

Note: Experiments were performed repeatedly with three random seeds to report mean and standard deviation.

5 Conclusions

In this paper, we proposed a novel Federated Split Task-Agnostic (FESTA) framework suitable to leverage the formidable benefit of Vision Transformer to simultaneously process multiple CXR tasks including the diagnosis of COVID-19. With the optimal configuration of Vision Transformer for modulation, it was possible to surpass the existing methods for distributed learning and achieve the performance comparable to data-centralized learning with our framework, even under the extremely skewed data distribution. Moreover, our framework alongside with clients to process multiple related tasks also improves the performances of individual tasks, while eliminating the needs to sharing the data and large weights of body network. These result suggest the suitability of the Transformer for collaborative learning in medical imaging, and pave the way forward future real-world application.

6 Limitation and Potential Negative Societal Impacts

This work is more like a proof-of-the-concept study, rather than a ready-to-use solution for industry. Therefore, it holds some limitations which may occur in real-world application. Recent works on privacy attacks in FL setting have implied that the belief that "Privacy can be protected by the decentralized nature of the FL" is actually not true [1, 51, 56]. In this work, however, the experimental results regarding the privacy issue such as threatening privacy via inversion attack are not suggested. Secondly, although the unique challenges of FL have been suggested by previous work including the problems of significant bottleneck in communication, stragglers, and fault tolerance which is exacerbated than in typical data-centralized learning [42, 26], we have not conducted the experiments to evaluate the robustness of our method against these unique challenges.

A potential risk arises from these limitations that our algorithm may not be free from privacy issue via model inversion attack against the server, since the server retains the parameters of the entire network in process of `FedAvg` of the heads and tails despite the split design of the sub-networks upon both client and server-sides. There is strong needs for further research on privacy and the practical challenges with the proposed method.

References

[1] Y. Aono, T. Hayashi, L. Wang, S. Moriai, et al. Privacy-preserving deep learning: Revisited and enhanced. In *International Conference on Applications and Techniques in Information Security*, pages 100–110. Springer, 2017.

[2] D. J. Beutel, T. Topal, A. Mathur, X. Qiu, T. Parcollet, P. P. de Gusmão, and N. D. Lane. Flower: A friendly federated learning research framework. *arXiv preprint arXiv:2007.14390*, 2020.

[3] A. Borghesi and R. Maroldi. Covid-19 outbreak in italy: experimental chest x-ray scoring system for quantifying and monitoring disease progression. *La radiologia medica*, 125:509–513, 2020.

[4] R. Caruana. Multitask learning. In S. Thrun and L. Pratt, editors, *Learning to learn*, pages 95–133. Springer, Boston, 1998.

[5] K. K. Chandriah and R. V. Naraganahalli. Maximizing a deep submodular function optimization with a weighted max-sat problem for trajectory clustering and motion segmentation. *Applied Intelligence*, pages 1–20, 2021.

[6] K. Chang, N. Balachandar, C. Lam, D. Yi, J. Brown, A. Beers, B. Rosen, D. L. Rubin, and J. Kalpathy-Cramer. Distributed deep learning networks among institutions for medical imaging. *Journal of the American Medical Informatics Association*, 25(8):945–954, 2018.

[7] G. Chartrand, P. M. Cheng, E. Vorontsov, M. Drozdzal, S. Turcotte, C. J. Pal, S. Kadoury, and A. Tang. Deep learning: a primer for radiologists. *Radiographics*, 37(7):2113–2131, 2017.

[8] H. Chen, Y. Wang, T. Guo, C. Xu, Y. Deng, Z. Liu, S. Ma, C. Xu, C. Xu, and W. Gao. Pre-trained image processing transformer. *arXiv preprint arXiv:2012.00364*, 2020.

[9] J. Chen, Y. Lu, Q. Yu, X. Luo, E. Adeli, Y. Wang, L. Lu, A. L. Yuille, and Y. Zhou. Transunet: Transformers make strong encoders for medical image segmentation. *arXiv preprint arXiv:2102.04306*, 2021.

[10] J. De Fauw, J. R. Ledsam, B. Romera-Paredes, S. Nikolov, N. Tomasev, S. Blackwell, H. Askham, X. Glorot, B. O'Donoghue, D. Visentin, et al. Clinically applicable deep learning for diagnosis and referral in retinal disease. *Nature medicine*, 24(9):1342–1350, 2018.

[11] M. De La Iglesia Vayá, J. M. Saborit, J. A. Montell, A. Pertusa, A. Bustos, M. Cazorla, J. Galant, X. Barber, D. Orozco-Beltrán, F. García-García, et al. Bimcv COVID-19+: a large annotated dataset of rx and ct images from COVID-19 patients. *arXiv preprint arXiv:2006.01174*, 2020.

[12] B. E. Dewey, C. Zhao, J. C. Reinhold, A. Carass, K. C. Fitzgerald, E. S. Sotirchos, S. Saidha, J. Oh, D. L. Pham, P. A. Calabresi, et al. Deepharmony: a deep learning approach to contrast harmonization across scanner changes. *Magnetic resonance imaging*, 64:160–170, 2019.

[13] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[14] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE international conference on computer vision*, pages 2650–2658, 2015.

[15] T. Evgeniou and M. Pontil. Regularized multi–task learning. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 109–117, 2004.

[16] Y. Fu, Y. Lei, T. Wang, W. J. Curran, T. Liu, and X. Yang. Deep learning in medical image registration: a review. *Physics in Medicine & Biology*, 65(20):20TR01, 2020.

[17] T. Gabruseva, D. Poplavskiy, and A. Kalinin. Deep learning for automatic pneumonia detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 350–351, 2020.

[18] M. Gaillochet, K. C. Tezcan, and E. Konukoglu. Joint reconstruction and bias field correction for undersampled mr imaging. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 44–52. Springer, 2020.

[19] Y. Gao, M. Kim, S. Abuadbba, Y. Kim, C. Thapa, K. Kim, S. A. Camtepe, H. Kim, and S. Nepal. End-to-end evaluation of federated learning and split learning for internet of things. *arXiv preprint arXiv:2003.13376*, 2020.

[20] H. Guo, R. Pasunuru, and M. Bansal. Soft layer-specific multi-task summarization with entailment and question generation. *arXiv preprint arXiv:1805.11004*, 2018.

[21] O. Gupta and R. Raskar. Distributed learning of deep neural network over multiple agents. *Journal of Network and Computer Applications*, 116:1–8, 2018.

[22] J. Irvin, P. Rajpurkar, M. Ko, Y. Yu, S. Ciurea-Ilcus, C. Chute, H. Marklund, B. Haghgoo, R. Ball, K. Shpanskaya, et al. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 590–597, 2019.

[23] B. Jing, P. Xie, and E. Xing. On the automatic generation of medical imaging reports. *arXiv preprint arXiv:1711.08195*, 2017.

[24] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah. Transformers in vision: A survey. *arXiv preprint arXiv:2101.01169*, 2021.

[25] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

[26] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020.

[27] S. Liu, E. Johns, and A. J. Davison. End-to-end multi-task learning with attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1871–1880, 2019.

[28] X. Liu, P. He, W. Chen, and J. Gao. Multi-task deep neural networks for natural language understanding. *arXiv preprint arXiv:1901.11504*, 2019.

[29] M.-T. Luong, Q. V. Le, I. Sutskever, O. Vinyals, and L. Kaiser. Multi-task sequence to sequence learning. *arXiv preprint arXiv:1511.06114*, 2015.

[30] P. M. Mammen. Federated learning: Opportunities and challenges. *arXiv preprint arXiv:2101.05428*, 2021.

[31] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282. PMLR, 2017.

[32] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert. Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3994–4003, 2016.

[33] S. Park, G. Kim, Y. Oh, J. B. Seo, S. M. Lee, J. H. Kim, S. Moon, J.-K. Lim, and J. C. Ye. Vision transformer using low-level chest x-ray feature corpus for covid-19 diagnosis and severity quantification. *arXiv preprint arXiv:2104.07235*, 2021.

[34] S.-A. Rebuffi, H. Bilen, and A. Vedaldi. Learning multiple visual domains with residual adapters. *arXiv preprint arXiv:1705.08045*, 2017.

[35] L. Rocher, J. M. Hendrickx, and Y.-A. De Montjoye. Estimating the success of re-identifications in incomplete datasets using generative models. *Nature communications*, 10(1):1–9, 2019.

[36] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[37] RSNA. RSNA Pneumonia Detection Challenge. `https://www.kaggle.com/c/rsna-pneumonia-detection-challenge`, 2018.

[38] S. Ruder, J. Bingel, I. Augenstein, and A. Søgaard. Latent multi-task architecture learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4822–4829, 2019.

[39] M. J. Sheller, G. A. Reina, B. Edwards, J. Martin, and S. Bakas. Multi-institutional deep learning modeling without sharing patient data: A feasibility study on brain tumor segmentation. In *International MICCAI Brainlesion Workshop*, pages 92–104. Springer, 2018.

[40] A. Signoroni, M. Savardi, S. Benini, N. Adami, R. Leonardi, P. Gibellini, F. Vaccher, M. Ravanelli, A. Borghesi, R. Maroldi, and D. Farina. Bs-net: learning covid-19 pneumonia severity on a large chest x-ray dataset. *Medical Image Analysis*, page 102046, 2021. ISSN 1361-8415. doi: https://doi.org/10.1016/j.media.2021.102046. URL `https://www.sciencedirect.com/science/article/pii/S136184152100092X`.

[41] SIIM-ACR. SIIM-ACR Pneumothorax Segmentation. `https://www.kaggle.com/c/siim-acr-pneumothorax-segmentation`, 2019.

[42] V. Smith, C.-K. Chiang, M. Sanjabi, and A. Talwalkar. Federated multi-task learning. *arXiv preprint arXiv:1705.10467*, 2017.

[43] C. Sun, A. Shrivastava, S. Singh, and A. Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852, 2017.

[44] N. Tajbakhsh, L. Jeyaseelan, Q. Li, J. N. Chiang, Z. Wu, and X. Ding. Embracing imperfect datasets: A review of deep learning solutions for medical image segmentation. *Medical Image Analysis*, 63:101693, 2020.

[45] W. G. Van Panhuis, P. Paul, C. Emerson, J. Grefenstette, R. Wilder, A. J. Herbst, D. Heymann, and D. S. Burke. A systematic review of barriers to data sharing in public health. *BMC public health*, 14(1):1–9, 2014.

[46] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[47] P. Vepakomma, O. Gupta, T. Swedish, and R. Raskar. Split learning for health: Distributed deep learning without sharing raw patient data. *arXiv preprint arXiv:1812.00564*, 2018.

[48] C. Wachinger, P. Golland, W. Kremen, B. Fischl, M. Reuter, A. D. N. Initiative, et al. Brainprint: A discriminative characterization of brain morphology. *NeuroImage*, 109:232–248, 2015.

[49] G. Wang, J. C. Ye, and B. De Man. Deep learning for tomographic image reconstruction. *Nature Machine Intelligence*, 2(12):737–748, 2020.

[50] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2097–2106, 2017.

[51] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang, and H. Qi. Beyond inferring class representatives: User-level privacy leakage from federated learning. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pages 2512–2520. IEEE, 2019.

[52] Z. Xu, Y. Huo, J. Park, B. Landman, A. Milkowski, S. Grbic, and S. Zhou. Less is more: Simultaneous view classification and landmark detection for abdominal ultrasound images. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 711–719. Springer, 2018.

[53] Q. Yang, Y. Liu, T. Chen, and Y. Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.

[54] W. Ye, J. Yao, H. Xue, and Y. Li. Weakly supervised lesion localization with probabilistic-cam pooling. *arXiv preprint arXiv:2005.14480*, 2020.

[55] M. Yurochkin, M. Agarwal, S. Ghosh, K. Greenewald, N. Hoang, and Y. Khazaeni. Probabilistic federated neural matching. 2018.

[56] B. Zhao, K. R. Mopuri, and H. Bilen. idlg: Improved deep leakage from gradients. *arXiv preprint arXiv:2001.02610*, 2020.

[57] S. K. Zhou, H. Greenspan, C. Davatzikos, J. S. Duncan, B. van Ginneken, A. Madabhushi, J. L. Prince, D. Rueckert, and R. M. Summers. A review of deep learning in medical imaging: Imaging traits, technology trends, case studies with progress highlights, and future promises. *Proceedings of the IEEE*, 2021.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section **??**.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 6.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section 6.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] Codes will be provided upon request.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 4.1.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Table 2, 3 and 5.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section 4.1.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [Yes] See Reference.

(b) Did you mention the license of the assets? [Yes] See Section 4.1. The public datasets can be used for research by citing the articles and links.

(c) Did you include any new assets either in the supplemental material or as a URL? [No] Codes will be provided upon request.

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] See Section 3.3. We use the public datasets and deliberately collected dataset ethically approved by Institutional Review Board (IRB).

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] See Section 3.3. We use the public datasets and deliberately collected dataset ethically approved by IRB.

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [Yes] See Section 3.3. Our deliberately collected dataset is ethically approved by IRB.

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]