ZOOPFL: Exploring Black-box Foundation Models for Personalized Federated Learning

Wang Lu[1*], **Hao Yu**[1*], **Jindong Wang**[2,3†], **Damien Teney**[4], **Haohan Wang**[5], **Yao Zhu**[1], **Yiqiang Chen**[6], **Qiang Yang**[7], **Xing xie**[3], **Xiangyang ji**[1]
[1]Tsinghua University [2]William & Mary [3]Microsoft Research Asia [4] Idiap Research Institute
[5] UIUC [6]Institute of Comput. Tech., CAS [7] HKUST

Abstract

When personalized federated learning (FL) meets large foundation models, new challenges arise from various limitations in resources. In addition to typical limitations such as data, computation, and communication costs, access to the models is also often limited. This paper endeavors to solve both the challenges of *limited resources* and *personalization*. i.e., distribution shifts between clients. To do so, we propose a method named **ZOOPFL** that uses **Z**eroth-**O**rder **O**ptimization for **P**ersonalized **F**ederated **L**earning. ZOOPFL avoids direct interference with the foundation models and instead learns to adapt its inputs through zeroth-order optimization. In addition, we employ simple yet effective linear projections to remap its predictions for personalization. To reduce the computation costs and enhance personalization, we propose input surgery to incorporate an auto-encoder with low-dimensional and client-specific embeddings. We provide theoretical support for ZOOPFL to analyze its convergence. Extensive empirical experiments on computer vision and natural language processing tasks using popular foundation models demonstrate its effectiveness for FL on black-box foundation models. [1]

1 Introduction

In recent years, the growing emphasis on data privacy and security has led to the emergence of federated learning (FL) [86, 12, 13, 9, 62, 35]. FL enables collaborative learning while safeguarding data privacy and security across distributed clients [97]. However, FL faces two key challenges: *limited resources* and *distribution shifts* (Figure 1 (a, b)).

The rise of large foundation models [5] has amplified these challenges. The computational demands and communication costs associated with such models hinder the deployment of existing FL approaches (Figure 1a). [2] Most of them require fine-tuning the models on every client.[3] Moreover, foundation models, often proprietary [77, 69], grant only *black-box* access, making FL resource-efficient applications a pressing research area.

Recent efforts in FL [93, 103, 16, 37] have attempted to reduce the number of optimized parameters to minimize computational and communication costs. As illustrated in Figure 1 (c), existing methods use prompts [45] or adapters [7] to fine-tune foundation models [93]. Other approaches [99, 44] focus on limiting the number of communication rounds. All of them however depend on white-box access

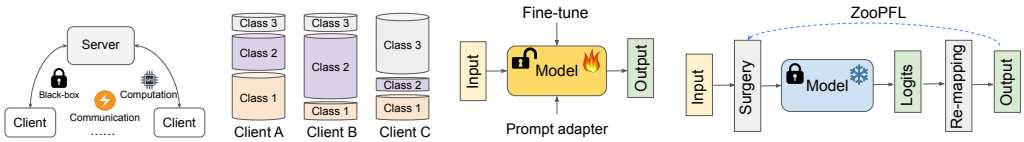
[*]Equal contribution. [†] Corresponding author(jindongwang@outlook.com).

[1]Our code is available at `https://anonymous.4open.science/r/zoopfltest-FC21`.

[2]Communication costs can be estimated as $C = p \times K \times T$, where p, T, K respectively denote the number of parameters, rounds, and clients. With GPT-3 for example [6], $p = 175$ billion parameters, making the communication of entire models impractical.

[3]Training GPT-2-small [61] requires at least two A100 GPUs for 16 hours, a resource unavailable to many.

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



(a) Limited resources (b) Distribution shifts (c) Existing methods (d) Our proposed ZOOPFL

Figure 1: ZOOPFL addresses federated learning with foundation models while coping with limited resources in communication, computation, and model accessibility (a) and being robust to distribution shifts (b). Most existing methods rely on white-box model access (c). ZOOPFL is applicable to black-box models by using input surgery and semantic output re-mapping (d).

Table 1: Comparisons of different methods.

Type	Method	Model scale	Model accessibility	Communication	Computation	Personalization
Base	FedAVG FedBN	Limited	White-box	Inefficient	High	Unsupported Supported
Large model for FL	FedPrompt, FedCLIP PromptFL, pFedPG	Unlimited	White-box	Efficient	Low High	Supported
Zero-order for FL	FwdLLM, BAFFLE FedZO	Unlimited Limited	White-box	Efficient	Low	Supported
Model reprogramming	Reprogrammable-FL	Unlimited	White-box	Efficient	High	Supported
Black-box foundation FL	ZooPFL (Ours)	Unlimited	Black-box	Efficient	Low	Supported

to the foundation models. Moreover, distribution shifts are an additional challenge for FL since the data across clients is not necessarily i.i.d. [39, 78] (Figure 1b). Directly aggregating information e.g., with FedAVG [53] often results in slow convergence and poor performance in each client [25]. Some methods have been designed to address the personalization of large foundation models [40, 65, 92]. However, they cannot deal with black-box models. The method proposed in this paper is designed to cope with label shift, i.e. variations in the distribution of labels among clients (Figure 1b).

In this paper, we propose ZOOPFL to cope with limited resources and personalization for federated learning and black-box foundation models. To cope with black-box models, ZOOPFL proposes two strategies, *input surgery* and *semantic re-mapping*, and learning through zeroth-order optimization (ZOO). To reduce the computational costs of ZOO and share information among clients, we employ an auto-encoder with low embedding dimensions to represent transformations. For better personalization, the client-specific embeddings and semantic re-mapping are preserved by each client. Figure 1 (d) illustrates that our proposed method learns transformations on the inputs and mappings of the outputs through *zeroth-order optimization* [46, 84, 42]. This bears similarities with model reprogramming [15] and Reprogrammable-FL [2], but the latter is unsuitable for black-box models and personalization. To the best of our knowledge, our method is the first to achieve federated learning with large black-box models, a challenging setting that is becoming increasingly relevant to the real world.

In summary, our contributions are four-fold.

1. **Scenario Exploration**: We delve into the challenges posed by fully black-box foundation models in FL. Our contribution lies in understanding and navigating this complex scenario.
2. **ZOOPFL Framework**: We introduce ZOOPFL, a comprehensive solution tailored for FL in resource-constrained and personalized settings. This framework encompasses input surgery and semantic re-mapping. ZOOPFL employs strategic input manipulations, leveraging dedicated embeddings, and employing zeroth-order optimization while it project outputs for specific task and personalization.
3. **Theoretical Support**: We provide theoretical support, enhancing ZOOPFL's credibility and offering insights into its workings.
4. **Empirical Validation**: ZOOPFL is rigorously evaluated through computer vision and natural language processing experiments, demonstrating the effectiveness and versatility of ZOOPFL.

2 Preliminaries

2.1 Black-box Federated Learning with Large Foundation Models

What is the setting? Foundation models can be viewed as black-box that can only provide the outputs according to inputs. It is not allowed to access any internal information on the foundation

models, which means, *no backpropagation* is allowed. Each client preserves the same foundation model *locally*. We do *not consider* the storage of large models and the additional costs associated with inference. We aim to utilize large black-box foundation models for better personalized FL. [4]

Why is it practical? To make the best of large foundation models, we can either fine-tune/adapt the models in their own data or perform FL on the cloud. Therefore, the value of our work lies primarily in two aspects. On the one hand, **fine-tuning or adapting locally is extremely expensive.** Fine-tuning on client side requires high computation and communication costs. Moreover, foundation model providers are not willing to share all information of model resources and it is a possible way that they can provide foundation models in an encrypted form. On the other hand, **FL on the cloud is not the ideal solution.** One cannot trust the cloud providers by uploading all the training data to the cloud. So, the best practice is to perform computation *locally*. Combining the above situations. i.e., updating models locally with low cost, one can conclude that our proposed black-box FL is the only solution. Specifically, note that "black-box" does not only mean we do not have model access; it is a more broad technique for model update when local BP cannot be performed due to large model sizes. In this situation, our proposed method is a preliminary attempt and exploration.

3 Methodology

In this section, we articulate our proposed ZOOPFL. We begin with problem formulation in Sec. 3.1. Then, we show the motivation of designing ZOOPFL in Sec. 3.2. Next, Sec. 3.3 introduces the details of our approach. Finally, we propose some discussions in section 3.4.

3.1 Problem Formulation

We assume there are n different clients $\{C_1, \cdots, C_n\}$ in personalized federated learning scenarios. Each client C_i has its own data $\mathcal{D}_i = \{\mathbf{x}_{i,j}, y_{i,j}\}_{j=1}^{n_i}$ where n_i means the number of data in the ith client. Data in different clients have different distributions, i.e. $P(\mathcal{D}_i) \neq P(\mathcal{D}_j)$. In the personalized FL setting, there exists the same black-box large foundation model in each client, g, which we know nothing inside and can only obtain logit outputs with fixed-size inputs. Our goal is to achieve personalized (i.e., satisfying) performance with black-box foundation models on each client by learning a significant transformation s_i on inputs and a re-mapping r_i on outputs without accessing g for each client $\mathcal{D}_i$. Specifically, denote ℓ a loss function, the learning objective is:

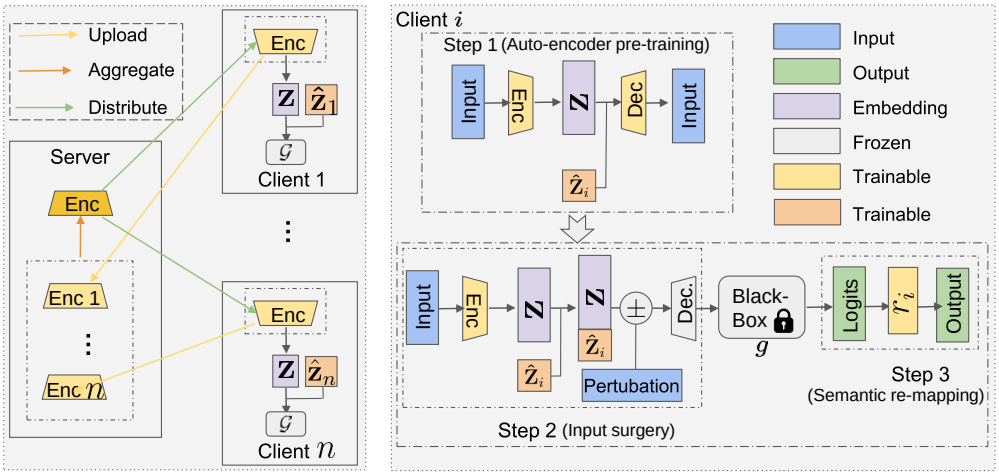
$$\min_{s_i, r_i} \frac{1}{n} \sum_{i=1}^{n} \frac{1}{n_i} \sum_{j=1}^{n_i} \ell(r_i(g(s_i(\mathbf{x}_{i,j}))), y_{i,j}). \tag{1}$$

3.2 Intuition

Input designs affect the performance of foundation models. Different representations with the same inputs can induce foundation models to make completely different predictions, which illustrates that adding interference or reconstructing inputs can be utilized for adaptation. However, most methods that add interference are performed at the sample level [88, 8, 48, 2, 105, 24] , i.e. special design for each sample, that are unsuitable to exchange information among clients and cannot cope with unseen samples. Therefore, it is necessary to reconstruct samples via an auto-encoder to adapt input with unchanged dimensions for foundation models. The exchange of auto-encoder parameters can facilitate the sharing of input transformation information across different clients.

Semantic re-mapping generates more semantically meaningful logits. Although large foundation models have been trained on a huge amount of samples [60], there still exists some classes or situations

[4] The real-life dilemma provided here illustrates the importance of direction. On the one hand, powerful LLMs, such as GPT4 [1] and Gemini [19], only provide APIs, that can be viewed black-box [31]. The data owners are unwilling to upload the raw data to APIs (Data processed by ZOOPFL might be able to be directly uploaded as shown in section D.6). On the other hand, the model suppliers are reluctant to expose all information about their models to others [73]. And clients usually cannot afford the back-propagation costs of LLMs. We believe that encrypting large models into black boxes and running them locally, similar to software [71], may be a solution. Therefore, we propose ZOOPFL, an initial exploration of this subdirection.



(a) Communication of server-client (b) Training process in client i

Figure 2: The framework of ZOOPFL. Please note that communications occur during step 2.

that foundation models cannot cover [83]. However, these new scenarios or classes can be made of existing fundamental elements or similar to some existing categories, which means foundation models can be able to extract remarkable features.[5] Considering layers between remarkable features and final logits as random projecting, re-mapping outputs with a simple linear layer can achieve acceptable performance similar to [32].

Design logic. Since access to foundation models is restricted, we have to rely on zeroth-order optimization methods to train auto-encoders, which leads that directly operating on the outputs of auto-encoders with high dimensions can exhaust unaffordable computational costs. To reduce the costs, we fix decoders and compute differences on embedding with low dimensions. For better personalization, we preserve semantic re-mapping in clients. Specifically, we preserve a client-specific embedding, i.e., a simple one-dimensional vector, for each client, which can be concatenated with embedding to generate adapted inputs with personalized characteristics.

3.3 ZOOPFL

In this paper, we propose ZOOPFL to learn input surgery and semantic re-mapping for black-box large foundation models in federated learning. ZOOPFL aims to adapt inputs to models and project outputs to meaningful semantic space. ZOOPFL mainly consists of three steps, namely, auto-encoder pre-training, input surgery, and semantic re-mapping.[6] Figure 2 shows the pipeline of our approach, where Figure 2(a) describes the communications between clients and the server and Figure 2(b) provides details on how to perform training on a local client.

The training process on a client is described as follows, where steps 2$\sim$3 are iterative.

1. Auto-encoder pre-training: this step directly utilizes inputs to pre-train the auto-encoder which then serves as the input surgery function.
2. Input surgery: this step only updates the encoder of auto-encoder and client-specific embeddings to transform the input consistent with the foundation model.
3. Semantic re-mapping: this step endeavors to re-map logits into meaningful semantic spaces with a simple linear projection.

Auto-encoder Pre-training. Before input surgery and semantic re-mapping that are assisted by labels, ZOOPFL firstly utilizes inputs of samples to pre-train auto-encoders for better initial understanding of client data and we will fix decoders in the next two steps. For client C_i, we denote

[5]Some popular language models such as BERT [20] and GPT-2 [61] in Huggingface utilize a random projection between extracted features and logits.

[6]Note that the pre-training here is different from the pre-training of large foundation models such as self-supervised pre-training. This step is much more efficient than pre-training a large foundation model since we only train a little auto-encoder.

$\hat{\mathbf{z}}_i$ as the ith client-specific embedding and $s_i = o_i \circ q_i$ where q_i and o_i represent the encoder and the decoder respectively. This step is unsupervised and each client utilizes MSE loss to train local s_i:

$$\ell_{MSE} = \mathbb{E}_{(\mathbf{x},y)\sim\mathbb{P}(\mathcal{D}_i)} \left\| o_i([q_i(\mathbf{x}), \hat{\mathbf{z}}_i]) - \mathbf{x} \right\|_2^2, \tag{2}$$

where $[\cdot,\cdot]$ denotes the concatenation operation. The updated encoder and decoder of each client are then transmitted to the server. Similar to FedAVG [53], the server aggregates the collected auto-encoders and distributes the aggregated one, s, to each client.

$$w(s) = \frac{1}{n}\sum_{i=1}^{n} w(s_i), \tag{3}$$

where $w(s)$ represent parameters of s. We assume that all clients contribute equally and participate in training. The above pre-training is iterative and we can obtain well-trained auto-encoders finally.

Input Surgery. After pre-training, input surgery optimizes encoders, q_i, to transform inputs consistent with foundation models. This step only exchanges encoders of clients to share common knowledge while each client preserves a client-specific embedding to represent personalized knowledge. In Figure 2, the foundation model, g, is black-box and the decoder is frozen. In the following, we elaborate on the whole training process in local clients.

In client C_i, an input $\mathbf{x}$ is first fed into the encoder q_i, generating an embedding vector $\mathbf{z} = q_i(\mathbf{x})$. Then we concatenate $\mathbf{z}$ with the client-specific embedding, $\hat{\mathbf{z}}_i$, and obtain the final embedding feature, $\tilde{\mathbf{z}} = [\mathbf{z}, \hat{\mathbf{z}}_i]$, which is then sent to the decoder. Once processed by the decoder, we can obtain $\tilde{\mathbf{x}} = o_i(\tilde{\mathbf{z}})$ with the same dimension as $\mathbf{x}$, and then the adapted input, $\tilde{\mathbf{x}}$, goes through the foundation model and the re-mapping layer, which generates the final prediction, $\tilde{\mathbf{y}}$. We utilize the cross-entropy loss ℓ_{cls} to guide the optimization:

$$\ell_1 = \mathbb{E}_{(\mathbf{x},y)\sim\mathbb{P}(\mathcal{D}_i)} \ell_{cls}(r_i(g(o_i([q_i(\mathbf{x}), \hat{\mathbf{z}}_i])), y). \tag{4}$$

However, the above objective cannot be directly optimized using the standard stochastic gradient descent since the foundation model g is frozen, preventing us from computing its gradient using back-propagation. We adopt the zeroth-order optimization method, specifically, the coordinate-wise gradient estimate (CGE), to learn q_i and $\hat{\mathbf{z}}_i$ [102, 75, 47, 42, 26]. To make the process clear and easy to understand, we freeze r_i and view o_i, g, and r_i as a whole module, $\mathcal{G}$, in this step.

Assume $\mathbf{z} \in \mathbb{R}^{d_1}$ and $\hat{\mathbf{z}}_i \in \mathbb{R}^{d_2}$. According to CGE, by adding a perturbation to $\tilde{\mathbf{z}}$, we obtain the new embedding and the corresponding classification loss,

$$\tilde{\mathbf{z}_1} = \tilde{\mathbf{z}} + \rho\mathbf{e}_j, \ell_{\mathbf{x},1} = \ell_{cls}(r_i(g(o_i(\tilde{\mathbf{z}_1}, y)))), \tag{5}$$

where $\mathbf{e}_j = (0, 0, \cdots, 1, 0, 0 \cdots, 0) \in \mathbb{R}^{d_1+d_2}$ denotes the jth elementary basic vector and ρ is a hyperparameter that describes the extent of the perturbation. Similarly, we can obtain $\tilde{\mathbf{z}_2}$ and $\ell_{\mathbf{x},2}$.

$$\tilde{\mathbf{z}_2} = \tilde{\mathbf{z}} - \rho\mathbf{e}_j, \ell_{\mathbf{x},2} = \ell_{cls}(r_i(g(o_i(\tilde{\mathbf{z}_2}, y)))). \tag{6}$$

Then, we have the gradient of $\mathcal{G}$ w.r.t. $\tilde{\mathbf{z}}$ computed as:

$$\nabla_{\tilde{\mathbf{z}}}\mathcal{G}(\tilde{\mathbf{z}}) = (\nabla_{\tilde{\mathbf{z}}}\mathcal{G}(\tilde{\mathbf{z}})_1, \nabla_{\tilde{\mathbf{z}}}\mathcal{G}(\tilde{\mathbf{z}})_2) \approx \sum_{i=1}^{d_1+d_2} \frac{\ell_{\mathbf{x},2} - \ell_{\mathbf{x},1}}{2\times\rho}\mathbf{e}_j. \tag{7}$$

For $\hat{\mathbf{z}}_i$, we directly update it with corresponding parts of $\nabla_{\tilde{\mathbf{z}}}\mathcal{G}(\tilde{\mathbf{z}})$ via a learning rate γ_2, $\hat{\mathbf{z}}_i^{new} = \hat{\mathbf{z}}_i - \gamma_2 \times \nabla_{\tilde{\mathbf{z}}}\mathcal{G}(\tilde{\mathbf{z}})_2$ where $\nabla_{\tilde{\mathbf{z}}}\mathcal{G}(\tilde{\mathbf{z}})_2$ denotes the last d_2 dimensions of $\nabla_{\tilde{\mathbf{z}}}\mathcal{G}(\tilde{\mathbf{z}})$. For $\nabla_{\tilde{\mathbf{z}}}\mathcal{G}(\tilde{\mathbf{z}})_1$, we can update q_i with the chain rule for differentiation.

$$\nabla_{q_i}\ell_1 = \frac{d\tilde{\mathbf{z}}}{dq_i}\frac{d\mathcal{G}}{d\tilde{\mathbf{z}}} \approx \frac{d\mathbf{z}}{dq_i}\nabla_{\tilde{\mathbf{z}}}\mathcal{G}(\tilde{\mathbf{z}})_1 \approx \frac{d\nabla_{\tilde{\mathbf{z}}}\mathcal{G}(\tilde{\mathbf{z}})_1\mathbf{z}}{dq_i}. \tag{8}$$

Finally, we can update the encoder, $w(q_i^{new}) = w(q_i) - \gamma_1 \times \nabla_{q_i}\ell_1$. Once all clients have updated encoders, we can aggregate encoders in the server and then distribute the aggregated encoder:

$$w(q) = \frac{1}{n}\sum_{i=1}^{n} w(q_i^{new}). \tag{9}$$

Semantic Re-mapping. In the last step, we train the encoder that enables the input consistent with foundation models. Here, we perform semantic re-mapping similar to [32]. This step only occurs in each client and no communication exists for simplicity and personalization. We view all parts before r_i as a whole module, $\mathcal{F}$, with two functions, including extracting features and mapping extracted features to a random space and we freeze $\mathcal{F}$. These two functions correspond to artificial features and the first layer in [32] respectively and we only update r_i corresponding to the second layer of ELM:

$$\ell_2 = \mathbb{E}_{(\mathbf{x},y)\sim\mathbb{P}(\mathcal{D}_i)}\ell_{cls}(r_i(\mathcal{F}(\mathbf{x})), y). \tag{10}$$

Since this part is behind g, $w(r_i)$ can be updated directly.

3.4 Discussion

We perform step 2 and step 3 iteratively. There also exist other zeroth-order optimization methods, e.g., the randomized gradient estimate (RGE) [46]. However, the concrete implementation of zeroth-order optimization is not our focus and we thereby choose CGE for deterministic and stability [47]. We assume foundation models exist on clients in the form of encrypted assets and we do not upload transformed inputs. Moreover, we do care about communication costs and GPU demands instead of training time in clients. To reduce training time, some techniques, such as RGE, random selections on $\mathbf{e}_j$, reduction on $d_1 + d_2$, etc., can be adopted and we leave this as our future work.

Our algorithm converges and the asymptotic convergence rate is $\mathbf{O}(\frac{1}{\sqrt{T}})$. For detailed theoretical analysis and the corresponding proofs, please refer to section A and section B. section C.1 provides more discussion on design logic while section C.2 provides the detailed algorithmic flow.

4 Experiments

4.1 Setup

Datasets and baselines. We evaluate ZOOPFL on 8 classification benchmarks with two modalities including computer vision (CV) and natural language processing (NLP). The benchmarks are COVID-19 (C-19) [63], APTOS [34], Terra100 (T-100) [4], Terra46 [4], SST-2 [81, 67], COLA [81, 87], Financial-phrasebank (Financial) [52], and Flipkart [76]. Brief information can be found in Table 2.[7] We filter meaningless samples and select samples for global class balance. Select strategies can be found in section D.1 while data distributions can be found in section D.2. To our best knowledge, no other methods are proposed and thereby we only compare ZOOPFL with zero-shot pre-trained models (ZS).

Table 2: Information of benchmarks.

Modality	Dataset	Samples	Classes	Clients	Selected Samples
CV	COVID-19	9,198	4	20	9,198
	APTOS	3,662	5	20	1,658
	Terra100	5,883	10	20	5,883
	Terra46	4,741	10	20	4,741
NLP	SST-2	67k	2	20	9,763
	COLA	8.5k	2	20	5,700
	Finanical	4,840	3	10	3,379
	Flipkart	205,053	3	20	3,048

Implementation details. For vision tasks, we set g as CLIP [60] with ResNet50 as the image backbone [60]. r is a linear layer with dimension M where M is the number of classes. q contains several blocks composed of a convolution layer, a RELU activation layer, a Batch Normalization layer, and a Pooling layer while o contains several blocks composed of a convTranspose layer, a RELU activation layer, and a Batch Normalization layer. We set $d_1 = 6 \times 7 \times 7 = 294$ and $d_2 = 2 \times 7 \times 7 = 98$. We set the learning rate for pretraining as 10^{-4} and set other learning rates as hyperparameters. For simplicity, other learning rates are all the same. We set the local epoch number as 1 and set the global round number $T = 120$. Moreover, we do not tune ρ but set $\rho = 5 \times 10^{-3}$. We select the best results according to accuracy on validation parts.

For language tasks, we select four foundation models, including ALBERT-base [36], BERT-base [20], DeBERTa-base [30], and GPT2 [61]. Note that there are recent large language foundation models such as Llama [72] and Falcon [56], but we can only experiment with the above ones due to constrained computational devices.[8] Our method works for all kinds of foundation models in various sizes. q

[7] We have chosen so many clients because it reflects the typical real-world scenario where there are numerous clients, each with relatively small amounts of data [93].

[8] Our hardware is a server with 4 V100 (16G) GPUs, which cannot afford to train larger foundation models.

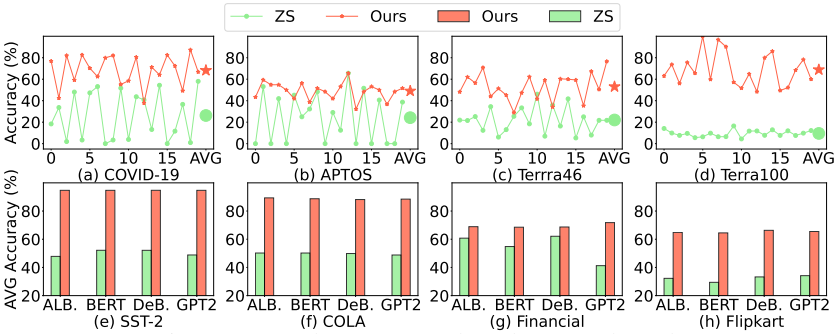
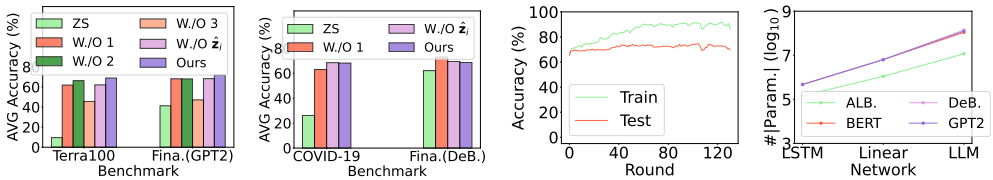


Figure 3: Results on CV (a-d) and NLP (e-h) tasks.



(a) Ablation of each part (b) Other observations (c) Convergence (d) Number of param.

Figure 4: Ablation study, convergence, and communication costs.

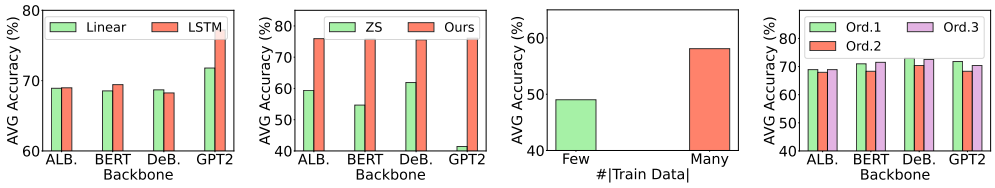
simply contains several linear layers followed by batch normalization layers. Please note that we transform input embeddings processed by foundation models for NLP instead of original texts. We set $d_1 = 128 - 32 = 96$ and $d_2 = 32$. We set the local epoch number as 1 and set the global round number $T = 130$. Other settings are similar to computer vision. For concrete structures of auto-encoders, please refer to section D.3.

4.2 Experimental Results

Figure 3 shows the results on all eight benchmarks and detailed results are in section D.4. And we have the following observations. 1) Our method achieves the best results on average for all benchmarks whatever the backbone is. It significantly outperforms the zero-shot method with remarkable improvements. In computer vision benchmarks, the improvements are about 42%, 25%, 21%, and 59% for COVID-19, Terra100, APTOS, and Terra46 respectively. In natural language processing benchmarks, for SST-2, COLA, and Flipkart, Financial-phrasebank, the improvements are about 43%, 39%, 15%, and 33% respectively. Please note that there only exist a few training data in each client (for COVID-19, each client only has about 50 samples.), which means utilizing foundation models is important. 2) Our method achieves the best accuracy in most clients, demonstrating the necessity of input surgery and semantic re-mapping. As shown in Figure 3(a)-(d), ZooPFL only performs slightly worse than ZS in few clients, e.g. client 13 on COVID-19, which can be due to the instability of zeroth-order optimization. 3) For natural language processing, different backbones bring different performance. From Figure 3(g), we can see that our method based on GPT2 can achieve better results compared to other backbones, although ZS performs the worst with GPT2. However, from Figure 3(f), we can see that ZooPFL based on GPT2 does not achieve the best performance. 4) Why large foundation models cannot achieve acceptable performances on these benchmarks? For computer vision, we choose COVID-19, APTOS, and Terra Incognita and these datasets can be missing during pretraining of CLIP, which leads the failure of CLIP with zero-shot. For natural language processing, although large foundation models can extract remarkable features, they need to be fine-tuned for downstream tasks, which means they may randomly guess without the post-processing. Due to these factors, post-processing to large foundation models is necessary, which is just what we explore in this paper.

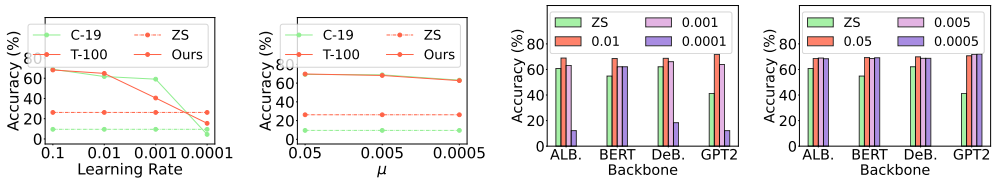
4.3 Analysis and Discussion

Ablation Study. Figure 4(a)-4(b) give experiments on ablation study and we have following observations. 1) In most situations, each part of ZooPFL can bring improvements. 2) Step 3 is more significant than Step 2. Since Step 3 re-maps outputs, it can offer semantic meanings to foundation models for specific tasks, which is more direct and effective intuitively. Step 2 transforms inputs that



(a) Backbone (b) Data splits (c) Data size (d) Optimization order

Figure 5: More discussions by varying backbones, data splits, data sizes, and optimization order.



(a) Learning rate (CV) (b) ρ (CV) (c) Learning rate (NLP) (d) ρ (NLP)

Figure 6: Parameter sensitivity of computer vision and natural language tasks (on finance data).

still go through foundation models or even random projections, and thereby it is indirect and less effective. However, by combining Step 2 with Step 3, we can achieve further improvements. 3) In some situations, client-specific embeddings do not bring remarkable improvements, which can be induced by two reasons. First, CGE is not stable enough and we cannot ensure ZooPFL finds the best global optimals. Second, to ensure fairness, we offer comparison methods without client-specific embeddings containing larger dimensions and thereby these methods can learn better representations for auto-encoders. 4) Step 1 brings significant improvement for CV while it is less effective for NLP. This can be due to two reasons. We provide better auto-encoders for CV but simple linear layers for NLP. Moreover, the closed pretraining of an auto-encoder without subsequent adjustments to the decoder may not be suitable for NLP. Fortunately, ZooPFL can achieve convincing improvements compared to ZS no matter whether adopting step 1.

Convergence and Communication Cost. We provide convergence analysis and communication cost comparisons in Figure 4(c) and 4(d), respectively. Figure 4(c) shows that both the average training accuracy and testing accuracy are convergent. There exist slight disturbances due to instability of CGE and the process of federated learning. Moreover, we can find that there exists a divergence between training and testing, which means there could be further improvements if more generalization techniques could be adopted which we leave as our future work. From Figure 4(d), we can see exchanging in encoders can reduce a significant amount of transmission cost, especially for LSTM, which means our method can be employed in reality.

More insightful analysis. 1) **Can stronger backbones bring better performance?** Figure 5(a) and 4(d) show that auto-encoders comprised of LSTM can bring better performance with fewer communications (especially for GPT2), which means more suitable backbones can lead to better performance. 2) **How can data splits influence the performance?** Figure 5(b) shows that ZooPFL still achieves better performance when using a different parameter for Dirichlet distribution (0.1 vs. 0.2) for NLP data split. In this more personalized situation, ZS maintains a similar performance while ours performs better. 3) **More training data, better results?** As shown in Figure 5(c), we choose the APTOS dataset where our method has the worst performance, to evaluate the influence of training data. We find that more training data can bring further improvements, which is completely consistent with our intuition. 4) **Can optimization order influence performance?** We provide three orders for optimization. Order 1 is what we adopted. Order 2 is to perform step 2 for all rounds and then perform step 3, which means these two steps are split. In order 3, we first optimize the encoder, then client-specific embeddings, and finally semantic re-mapping layers, and these parts are iterative. As shown in Figure 5(d), Order 1 and Order 3 can perform slightly better than Order 2, which demonstrates the necessity of joint optimization. More analysis can be found in section D.

Parameter sensitivity. Figure 6 provides parameter sensitivity and we obtain following observations. 1) Our method is stable for a wide range of parameters although CGE may lead instability. 2) For most situations, larger learning rates with Adam can bring better performance. 3) ZooPFL can achieve further improvement if we finetune hyperparameters more carefully. For example, we can choose larger learning rates, e.g.0.5 or choose more suitable ρ for specific tasks, e.g. 0.05 for CV.

5 Related Work

Federated learning makes it possible to perform distributed multi-party computing without comprising privacy [101, 79, 53, 97, 55, 80, 58, 89, 70, 49]. FedAVG is the baseline algorithm for FL by exchanging parameters instead of raw data, which has been used in many applications [38, 3, 62]. When FedAVG meets non-iid data, it can suffer from low convergence speed and terrible **personalization** performance [64, 39, 40, 33, 93, 82, 66, 17]. [65] proposed PerFedMask that considers the computational capability of different devices while some other work considered utilizing knowledge distillation for personalization [14]. Besides personalization, there exists research focusing on generalization [12, 29, 59]. Our method can deal with situations where distribution shifts exist.

Since deep learning has entered the era of **large foundation models** [5, 91, 106], some novel issues, e.g. computation costs and communication costs, are coming into being, leading operations on the whole network impossible [85, 18, 10, 21, 90, 68, 43, 11, 28, 104]. FedPrompt [103] studied prompt tuning in a model split aggregation way while FedCLIP [51] designed an attention-based adapter for CLIP [60]. FwdLLM [93] combined BP-free training with parameter-efficient training methods while pFedPG [95] deployed a personalized prompt generator at the server to produce client-specific visual prompts. These methods all require access to the internals of large models.

Besides data privacy, **model privacy** also raised attention recently [54]. Model suppliers are usually more willing to only provide predictions for given inputs or just provide a product that can only generate predictions [77]. In this paper, we view these protected foundation models as black-box models [27, 50]. Little work paid attention to finetuning or optimizing in this field, but most related work focused on attacks [96, 98]. One related work is FedZO [22] which utilized **zero-order optimization** [26], but it did not consider utilizing large foundation models. Some other work also made use of zero-order optimization for federated learning [41, 100, 23], but none of them utilized large black-box foundation models.

Model reprogramming (MR) [74, 94, 15] provides a similar solution to ZOOPFL. It trains the inserted input transformation and output mapping layers while keeping the source pretrained model inact to enable resource-efficient cross-domain machine learning. The main purpose of model reprogramming is to transfer knowledge to targets and it can be viewed as a sub-field of transfer learning. Recently, [2] proposed the first framework, Reprogrammable-FL, adapting MR to the setting of differentially private federated learning. Reprogrammable-FL learned an input transformation for samples and added learned perturbations to the original samples. It preserved local input transformations and shared output transformation layers, which are totally in contrast to ours. Moreover, ZOOPFL is proposed for black-box foundation models and can provide an ideal personalization capability.

6 Conclusion and Discussion

We proposed ZOOPFL which can deal with large black-box models in federated learning. ZOOPFL mainly consists of two parts, including input surgery and semantic re-mapping. Moreover, with a client-specific embedding, ZOOPFL can be more personalized. We demonstrated its effectiveness on both CV and NLP tasks. ZOOPFL achieved remarkable performance without large communication costs and high demands of GPUs.

As the first exploration in black-box federated learning for large foundation models, ZOOPFL can be more perfect by pursuing the following avenues. 1) Since the stability and speed of CGE influence the performance of step 2, it can be better to seek more stable and efficient optimization algorithms. 2) Foundation models in ZOOPFL can be enhanced by other ways, e.g., auxiliary models, to serve as a complement to foundation models. 3) Experiments with larger foundation models can be performed for evaluation if computational resources are enough in the future.

Broader Impacts

ZOOPFL has transformative impacts across various sectors by enabling personalized models while ensuring data privacy. In healthcare, it enhances personalized medicine and remote monitoring. In finance, it improves fraud detection and financial services. Smart cities and IoT applications leverage ZooPFL for customized services and device optimization. ZOOPFL's ability to handle client heterogeneity and enhance privacy drives innovation across these fields.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] Huzaifa Arif, Alex Gittens, and Pin-Yu Chen. Reprogrammable-fl: Improving utility-privacy tradeoff in federated learning via model reprogramming. In *2023 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pages 197–209. IEEE, 2023.

[3] Syreen Banabilah, Moayad Aloqaily, Eitaa Alsayed, Nida Malik, and Yaser Jararweh. Federated learning review: Fundamentals, enabling technologies, and future applications. *Information processing & management*, 59(6):103061, 2022.

[4] Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *Proceedings of the European conference on computer vision (ECCV)*, pages 456–473, 2018.

[5] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[6] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[7] Dongqi Cai, Yaozong Wu, Shangguang Wang, Felix Xiaozhu Lin, and Mengwei Xu. Fedadapter: Efficient federated learning for modern nlp. *arXiv preprint arXiv:2205.10162*, 2022.

[8] Jialun Cao, Meiziniu Li, Ming Wen, and Shing-chi Cheung. A study on prompt design, advantages and limitations of chatgpt for deep learning program repair. *arXiv preprint arXiv:2304.08191*, 2023.

[9] Timothy Castiglia, Yi Zhou, Shiqiang Wang, Swanand Ravindra Kadhe, Nathalie Baracaldo Angel, and Stacy Patterson. Less-vfl: Communication-efficient feature selection for vertical federated learning. In *International Conference on Machine Learning*, 2023.

[10] Chaochao Chen, Xiaohua Feng, Jun Zhou, Jianwei Yin, and Xiaolin Zheng. Federated large language model: A position paper. *arXiv preprint arXiv:2307.08925*, 2023.

[11] Daoyuan Chen, Liuyi Yao, Dawei Gao, Bolin Ding, and Yaliang Li. Efficient personalized federated learning via sparse model-adaptation. In *International Conference on Machine Learning*, pages 5234–5256. PMLR, 2023.

[12] Hong-You Chen and Wei-Lun Chao. On bridging generic and personalized federated learning for image classification. In *International Conference on Learning Representations*, 2022.

[13] Hong-You Chen, Cheng-Hao Tu, Ziwei Li, Han Wei Shen, and Wei-Lun Chao. On the importance and applicability of pre-training for federated learning. In *The Eleventh International Conference on Learning Representations*, 2023.

[14] Huancheng Chen, Chaining Wang, and Haris Vikalo. The best of both worlds: Accurate global and personalized models through federated learning with data-free hyper-knowledge distillation. In *The Eleventh International Conference on Learning Representations*, 2023.

[15] Pin-Yu Chen. Model reprogramming: Resource-efficient cross-domain machine learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 22584–22591, 2024.

[16] Shengchao Chen, Guodong Long, Tao Shen, and Jing Jiang. Prompt federated learning for weather forecasting: toward foundation models on meteorological data. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pages 3532–3540, 2023.

[17] Yiyue Chen, Haris Vikalo, and Chianing Wang. Fed-qssl: A framework for personalized federated learning under bitwidth and data heterogeneity. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 11443–11452, 2024.

[18] Yuanyuan Chen, Zichen Chen, Pengcheng Yu, and Han Yu. Fedobd: opportunistic block dropout for efficiently training large-scale neural networks through federated learning. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pages 3541–3549, 2023.

[19] Google DeepMind. Gemini.

[20] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[21] Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, et al. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence*, 5(3):220–235, 2023.

[22] Wenzhi Fang, Ziyi Yu, Yuning Jiang, Yuanming Shi, Colin N Jones, and Yong Zhou. Communication-efficient stochastic zeroth-order optimization for federated learning. *IEEE Transactions on Signal Processing*, 70:5058–5073, 2022.

[23] Haozhe Feng, Tianyu Pang, Chao Du, Wei Chen, Shuicheng Yan, and Min Lin. Does federated learning really need backpropagation? *arXiv preprint arXiv:2301.12195*, 2023.

[24] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit Haim Bermano, Gal Chechik, and Daniel Cohen-or. An image is worth one word: Personalizing text-to-image generation using textual inversion. In *The Eleventh International Conference on Learning Representations*, 2022.

[25] Liang Gao, Huazhu Fu, Li Li, Yingwen Chen, Ming Xu, and Cheng-Zhong Xu. Feddc: Federated learning with non-iid data via local drift decoupling and correction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10112–10121, 2022.

[26] Saeed Ghadimi and Guanghui Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

[27] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, 51(5):1–42, 2018.

[28] Tao Guo, Song Guo, Junxiao Wang, Xueyang Tang, and Wenchao Xu. Promptfl: Let federated participants cooperatively learn prompts instead of models-federated learning in age of foundation model. *IEEE Transactions on Mobile Computing*, 2023.

[29] Sharut Gupta, Kartik Ahuja, Mohammad Havaei, Niladri Chatterjee, and Yoshua Bengio. Fl games: A federated learning framework for distribution shifts. In *Workshop on Federated Learning: Recent Advances and New Challenges (in Conjunction with NeurIPS 2022)*, 2022.

[30] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*, 2021.

[31] Bairu Hou, Joe O'connor, Jacob Andreas, Shiyu Chang, and Yang Zhang. Promptboosting: Black-box text classification with ten forward passes. In *International Conference on Machine Learning*, pages 13309–13324. PMLR, 2023.

[32] Gao Huang, Guang-Bin Huang, Shiji Song, and Keyou You. Trends in extreme learning machines: A review. *Neural Networks*, 61:32–48, 2015.

[33] Liangze Jiang and Tao Lin. Test-time robust personalization for federated learning. In *The Eleventh International Conference on Learning Representations*, 2023.

[34] Sohier Dane Karthik, Maggie. Aptos 2019 blindness detection, 2019.

[35] Weirui Kuang, Bingchen Qian, Zitao Li, Daoyuan Chen, Dawei Gao, Xuchen Pan, Yuexiang Xie, Yaliang Li, Bolin Ding, and Jingren Zhou. Federatedscope-llm: A comprehensive package for fine-tuning large language models in federated learning. *arXiv preprint arXiv:2309.00363*, 2023.

[36] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*, 2020.

[37] Guanghao Li, Wansen Wu, Yan Sun, Li Shen, Baoyuan Wu, and Dacheng Tao. Visual prompt based personalized federated learning. *Transactions on Machine Learning Research*, 2023.

[38] Li Li, Yuxi Fan, Mike Tse, and Kuo-Yi Lin. A review of applications in federated learning. *Computers & Industrial Engineering*, 149:106854, 2020.

[39] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020.

[40] Xiaoxiao Li, Meirui JIANG, Xiaofei Zhang, Michael Kamp, and Qi Dou. Fedbn: Federated learning on non-iid features via local batch normalization. In *International Conference on Learning Representations*, 2021.

[41] Zan Li and Li Chen. Communication-efficient decentralized zeroth-order method on heterogeneous data. In *2021 13th International Conference on Wireless Communications and Signal Processing (WCSP)*, pages 1–6. IEEE, 2021.

[42] Xiangru Lian, Huan Zhang, Cho-Jui Hsieh, Yijun Huang, and Ji Liu. A comprehensive linear speedup analysis for asynchronous stochastic parallel optimization from zeroth-order to first-order. *Advances in Neural Information Processing Systems*, 29, 2016.

[43] Zihao Lin, Yan Sun, Yifan Shi, Xueqian Wang, Lifu Huang, Li Shen, and Dacheng Tao. Efficient federated prompt tuning for black-box large pre-trained models. *arXiv preprint arXiv:2310.03123*, 2023.

[44] Chang Liu, Chenfei Lou, Runzhong Wang, Alan Yuhan Xi, Li Shen, and Junchi Yan. Deep neural network fusion via graph matching with applications to model ensemble and federated learning. In *International Conference on Machine Learning*, pages 13857–13869. PMLR, 2022.

[45] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35, 2023.

[46] Sijia Liu, Pin-Yu Chen, Bhavya Kailkhura, Gaoyuan Zhang, Alfred O Hero III, and Pramod K Varshney. A primer on zeroth-order optimization in signal processing and machine learning: Principals, recent advances, and applications. *IEEE Signal Processing Magazine*, 37(5):43–54, 2020.

[47] Sijia Liu, Bhavya Kailkhura, Pin-Yu Chen, Paishun Ting, Shiyu Chang, and Lisa Amini. Zeroth-order stochastic variance reduction for nonconvex optimization. *Advances in Neural Information Processing Systems*, 31, 2018.

[48] Vivian Liu and Lydia B Chilton. Design guidelines for prompt engineering text-to-image generative models. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, pages 1–23, 2022.

[49] Yang Liu, Yan Kang, Tianyuan Zou, Yanhong Pu, Yuanqin He, Xiaozhou Ye, Ye Ouyang, Ya-Qin Zhang, and Qiang Yang. Vertical federated learning: Concepts, advances, and challenges. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[50] Lennart Ljung. Black-box models from input-output measurements. In *IMTC 2001. Proceedings of the 18th IEEE instrumentation and measurement technology conference. Rediscovering measurement in the age of informatics (Cat. No. 01CH 37188)*, volume 1, pages 138–146. IEEE, 2001.

[51] Wang Lu, HU Xixu, Jindong Wang, and Xing Xie. Fedclip: Fast generalization and personalization for clip in federated learning. In *ICLR 2023 Workshop on Trustworthy and Reliable Large-Scale Machine Learning Models*, 2023.

[52] P. Malo, A. Sinha, P. Korhonen, J. Wallenius, and P. Takala. Good debt or bad debt: Detecting semantic orientations in economic texts. *Journal of the Association for Information Science and Technology*, 65, 2014.

[53] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[54] Fan Mo, Ali Shahin Shamsabadi, Kleomenis Katevas, Soteris Demetriou, Ilias Leontiadis, Andrea Cavallaro, and Hamed Haddadi. Darknetz: towards model privacy at the edge using trusted execution environments. In *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services*, pages 161–174, 2020.

[55] Jaehoon Oh, SangMook Kim, and Se-Young Yun. Fedbabu: Toward enhanced representation for federated image classification. In *International Conference on Learning Representations*, 2022.

[56] Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Hamza Alobeidli, Alessandro Cappelli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. The refinedweb dataset for falcon llm: Outperforming curated corpora with web data only. *Advances in Neural Information Processing Systems*, 36, 2024.

[57] Krishna Pillutla, Kshitiz Malik, Abdel-Rahman Mohamed, Mike Rabbat, Maziar Sanjabi, and Lin Xiao. Federated learning with partial model personalization. In *International Conference on Machine Learning*, pages 17716–17758. PMLR, 2022.

[58] Tao Qi, Fangzhao Wu, Lingjuan Lyu, Yongfeng Huang, and Xing Xie. Fedsampling: a better sampling strategy for federated learning. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pages 4154–4162, 2023.

[59] Zhe Qu, Xingyu Li, Rui Duan, Yao Liu, Bo Tang, and Zhuo Lu. Generalized federated learning via sharpness aware minimization. In *International Conference on Machine Learning*, pages 18250–18280. PMLR, 2022.

[60] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[61] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[62] Nuria Rodríguez-Barroso, Daniel Jiménez-López, M Victoria Luzón, Francisco Herrera, and Eugenio Martínez-Cámara. Survey on federated learning threats: Concepts, taxonomy on attacks and defences, experimental study and challenges. *Information Fusion*, 90:148–173, 2023.

[63] Unais Sait, KG Lal, S Prajapati, Rahul Bhaumik, Tarun Kumar, S Sanjana, and Kriti Bhalla. Curated dataset for covid-19 posterior-anterior chest radiography images (x-rays). *Mendeley Data*, 1, 2020.

[64] Felix Sattler, Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek. Robust and communication-efficient federated learning from non-iid data. *IEEE transactions on neural networks and learning systems*, 31(9):3400–3413, 2019.

[65] Mehdi Setayesh, Xiaoxiao Li, and Vincent WS Wong. Perfedmask: Personalized federated learning with optimized masking vectors. In *The Eleventh International Conference on Learning Representations*, 2023.

[66] Mingjia Shi, Yuhao Zhou, Kai Wang, Huaizheng Zhang, Shudong Huang, Qing Ye, and Jiancheng Lv. Prior: Personalized prior for reactivating the information overlooked in federated learning. *Advances in Neural Information Processing Systems*, 36, 2024.

[67] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics.

[68] Jingwei Sun, Ziyue Xu, Hongxu Yin, Dong Yang, Daguang Xu, Yiran Chen, and Holger R Roth. Fedbpt: Efficient federated black-box prompt tuning for large language models. *arXiv preprint arXiv:2310.01467*, 2023.

[69] Tianxiang Sun, Yunfan Shao, Hong Qian, Xuanjing Huang, and Xipeng Qiu. Black-box tuning for language-model-as-a-service. In *International Conference on Machine Learning*, pages 20841–20855. PMLR, 2022.

[70] Yitong Tang. Adapted weighted aggregation in federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 23763–23765, 2024.

[71] E Thambiraja, G Ramesh, and Dr R Umarani. A survey on various most common encryption techniques. *International journal of advanced research in computer science and software engineering*, 2(7), 2012.

[72] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[73] Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. Stealing machine learning models via prediction {APIs}. In *25th USENIX security symposium (USENIX Security 16)*, pages 601–618, 2016.

[74] Yun-Yun Tsai, Pin-Yu Chen, and Tsung-Yi Ho. Transfer learning without knowing: Reprogramming black-box machine learning models with scarce data and limited resources. In *International Conference on Machine Learning*, pages 9614–9624. PMLR, 2020.

[75] Chun-Chen Tu, Paishun Ting, Pin-Yu Chen, Sijia Liu, Huan Zhang, Jinfeng Yi, Cho-Jui Hsieh, and Shin-Ming Cheng. Autozoom: Autoencoder-based zeroth order optimization method for attacking black-box neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 742–749, 2019.

[76] Nirali Vaghani and Mansi Thummar. Flipkart product reviews with sentiment dataset, 2023.

[77] Eva AM Van Dis, Johan Bollen, Willem Zuidema, Robert van Rooij, and Claudi L Bockting. Chatgpt: five priorities for research. *Nature*, 614(7947):224–226, 2023.

[78] Raviteja Vemulapalli, Warren Richard Morningstar, Philip Andrew Mansfield, Hubert Eichner, Karan Singhal, Arash Afkanpour, and Bradley Green. Federated training of dual encoding models on small non-iid client datasets. In *ICLR 2023 Workshop on Pitfalls of limited data and computation for Trustworthy ML*, 2023.

[79] Paul Voigt and Axel Von dem Bussche. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 10(3152676):10–5555, 2017.

[80] Wei Wan, Shengshan Hu, Minghui Li, Jianrong Lu, Longling Zhang, Leo Yu Zhang, and Hai Jin. A four-pronged defense against byzantine attacks in federated learning. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 7394–7402, 2023.

[81] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*, 2019.

[82] Jiaqi Wang, Xingyi Yang, Suhan Cui, Liwei Che, Lingjuan Lyu, Dongkuan DK Xu, and Fenglong Ma. Towards personalized federated learning via heterogeneous model reassembly. *Advances in Neural Information Processing Systems*, 36, 2024.

[83] Yidong Wang, Zhuohao Yu, Jindong Wang, Qiang Heng, Hao Chen, Wei Ye, Rui Xie, Xing Xie, and Shikun Zhang. Exploring vision-language models for imbalanced learning. *International Journal of Computer Vision (IJCV)*, 2023.

[84] Yining Wang, Simon Du, Sivaraman Balakrishnan, and Aarti Singh. Stochastic zeroth-order optimization in high dimensions. In *International conference on artificial intelligence and statistics*, pages 1356–1365. PMLR, 2018.

[85] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 139–149, 2022.

[86] Stefanie Warnat-Herresthal, Hartmut Schultze, Krishnaprasad Lingadahalli Shastry, Sathyanarayanan Manamohan, Saikat Mukherjee, Vishesh Garg, Ravi Sarveswara, Kristian Händler, Peter Pickkers, N Ahmad Aziz, et al. Swarm learning for decentralized and confidential clinical machine learning. *Nature*, 594(7862):265–270, 2021.

[87] Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641, 2019.

[88] Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C Schmidt. A prompt pattern catalog to enhance prompt engineering with chatgpt. *arXiv preprint arXiv:2302.11382*, 2023.

[89] Yue Wu, Shuaicheng Zhang, Wenchao Yu, Yanchi Liu, Quanquan Gu, Dawei Zhou, Haifeng Chen, and Wei Cheng. Personalized federated learning under mixture of distributions. In *International Conference on Machine Learning*, pages 37860–37879. PMLR, 2023.

[90] Guangxuan Xiao, Ji Lin, and Song Han. Offsite-tuning: Transfer learning without full model. *arXiv preprint arXiv:2302.04870*, 2023.

[91] Pengwei Xing, Songtao Lu, and Han Yu. Fedlogic: Interpretable federated multi-domain chain-of-thought prompt selection for large language models. *arXiv preprint arXiv:2308.15324*, 2023.

[92] Jian Xu, Xinyi Tong, and Shao-Lun Huang. Personalized federated learning with feature alignment and classifier collaboration. In *The Eleventh International Conference on Learning Representations*, 2023.

[93] Mengwei Xu, Yaozong Wu, Dongqi Cai, Xiang Li, and Shangguang Wang. Federated fine-tuning of billion-sized language models across mobile devices. *arXiv preprint arXiv:2308.13894*, 2023.

[94] Shoukai Xu, Jiangchao Yao, Ran Luo, Shuhai Zhang, Zihao Lian, Mingkui Tan, and Yaowei Wang. Towards efficient task-driven model reprogramming with foundation models. *arXiv preprint arXiv:2304.02263*, 2023.

[95] Fu-En Yang, Chien-Yi Wang, and Yu-Chiang Frank Wang. Efficient model personalization in federated learning via client-specific prompt generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19159–19168, 2023.

[96] Ming Yang, Hang Cheng, Fei Chen, Ximeng Liu, Meiqing Wang, and Xibin Li. Model poisoning attack in differential privacy-based federated learning. *Information Sciences*, 630:158–172, 2023.

[97] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.

[98] Ruikang Yang, Jianfeng Ma, Junying Zhang, Saru Kumari, Sachin Kumar, and Joel JPC Rodrigues. Practical feature inference attack in vertical federated learning during prediction in artificial internet of things. *IEEE Internet of Things Journal*, 2023.

[99] Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan Greenewald, Nghia Hoang, and Yasaman Khazaeni. Bayesian nonparametric federated learning of neural networks. In *International conference on machine learning*, pages 7252–7261. PMLR, 2019.

[100] Eric Zelikman, Qian Huang, Percy Liang, Nick Haber, and Noah D Goodman. Just one byte (per gradient): A note on low-bandwidth decentralized language model finetuning using shared randomness. *arXiv preprint arXiv:2306.10015*, 2023.

[101] Chen Zhang, Yu Xie, Hang Bai, Bin Yu, Weihong Li, and Yuan Gao. A survey on federated learning. *Knowledge-Based Systems*, 216:106775, 2021.

[102] Yimeng Zhang, Yuguang Yao, Jinghan Jia, Jinfeng Yi, Mingyi Hong, Shiyu Chang, and Sijia Liu. How to robustify black-box ml models? a zeroth-order optimization perspective. In *International Conference on Learning Representations*, 2022.

[103] Haodong Zhao, Wei Du, Fangqi Li, Peixuan Li, and Gongshen Liu. Fedprompt: Communication-efficient and privacy-preserving prompt tuning in federated learning. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

[104] Hanhan Zhou, Tian Lan, Guru Prasadh Venkataramani, and Wenbo Ding. Every parameter matters: Ensuring the convergence of federated learning with dynamic heterogeneous models reduction. *Advances in Neural Information Processing Systems*, 36, 2024.

[105] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022.

[106] Weiming Zhuang, Chen Chen, and Lingjuan Lyu. When foundation model meets federated learning: Motivations, challenges, and future directions. *arXiv preprint arXiv:2306.15546*, 2023.

Contents

A Theoretical Analysis **18**

B Proofs **18**

B.1 Convergence Analysis of ZooPFL 21

C Methodology **25**

C.1 More Discussion on Design Logic 25

C.2 Summary 25

D Experiments **25**

D.1 Description of the Datasets 25

D.2 Data Distributions 27

D.3 Model Structures 27

D.4 Additional Results 27

D.5 Resource Consumption Analysis 29

D.6 Visualization Study 29

A Theoretical Analysis

We present the convergence analysis of ZOOPFL. There exist three parts to optimize during step 2 and step 3, including the parameters of the encoder ($\hat{\mathbf{u}}$), clients specific embeddings ($\mathbf{v}_{1,i}$), and semantic re-mapping layers ($\mathbf{v}_{2,i}$). Following [57], we group parameters into two parts, i.e. $\mathbf{u}_i := \hat{\mathbf{u}} \cup \mathbf{v}_{1,i}$, $\mathbf{v}_i := \mathbf{v}_{2,i}$.[9] Now we give the main conclusion with proofs in section B.

Theorem 1. *Suppose assumptions 1, 2, 3, and 4 hold, and the learning rates in* ZOOPFL *are chosen as $\gamma_{\mathbf{u}} = \eta/(L_{\mathbf{u}}\tau_{\mathbf{u}})$ and $\gamma_{\mathbf{v}} = \eta/(L_{\mathbf{v}}\tau_{\mathbf{v}})$, with*

$$\eta \leq min\left\{\frac{1}{24(1+\mu^2)}, \frac{m}{128\chi^2(n-m)}, \sqrt{\frac{m}{\chi^2 n}}\right\}. \tag{11}$$

Then, right after the training of T epochs, ignoring absolute constants, we have

$$\frac{1}{T}\sum_{t=0}^{T-1}\left(\frac{1}{L_{\mathbf{u}}}\mathbb{E}\left[\Delta_{\mathbf{u}}^{(t)}\right] + \frac{m}{nL_{\mathbf{v}}}\mathbb{E}\left[\Delta_{\mathbf{v}}^{(t)}\right]\right) \leq \frac{\Delta F_0}{\eta T} + \eta\sigma_{alt,1}^2 + \eta^2\sigma_{alt,2}^2. \tag{12}$$

Corollary 1. *An optimal learning rate is:*

$$\begin{aligned}\eta = &\left(\frac{\Delta F_0}{T\sigma_{alt,1}^2}\right)^{1/2} \bigwedge \left(\frac{\Delta F_0^2}{T^2\sigma_{alt,2}^2}\right)^{1/3} \\ &\bigwedge \frac{1}{1+\mu^2} \bigwedge \frac{m}{\chi^2(n-m)} \bigwedge \sqrt{\frac{m}{\chi^2 n}}.\end{aligned} \tag{13}$$

We have, ignoring absolute constants,

$$\begin{aligned}&\frac{1}{T}\sum_{t=0}^{T-1}\left(\frac{1}{L_{\mathbf{u}}}\mathbb{E}\left[\Delta_{\mathbf{u}}^{(t)}\right] + \frac{m}{nL_{\mathbf{v}}}\mathbb{E}\left[\Delta_{\mathbf{v}}^{(t)}\right]\right) \leq \\ &\frac{(\Delta F_0\sigma_{alt,1}^2)^{1/2}}{\sqrt{T}} + \frac{(\Delta F_0^2\sigma_{alt,2}^2)^{1/3}}{T^{2/3}} \\ &+ \frac{\Delta F_0}{T}\left(1+\mu^2+\chi^2(\frac{n}{m}-1) + \sqrt{\chi^2\frac{n}{m}}\right).\end{aligned} \tag{14}$$

The measure of convergence of Algorithm 1 is in terms of the weighted average of square norms of the gradients of loss function $\mathbb{E}\left[\Delta_{\mathbf{u}}^{(t)}\right]$ and $\mathbb{E}\left[\Delta_{\mathbf{v}}^{(t)}\right]$ through iterations from 1 to $T-1$, i.e. the left hand of (14). As the square norms of the gradients of loss function at the optimal solution is zero, whether or not these norms approach zero is a good criterion of the convergence. With this choice of optimal learning rate, it is clear from the right hand of (14) that our algorithm converges and the asymptotic convergence rate is $\mathbf{O}(\frac{1}{\sqrt{T}})$.

B Proofs

The proof is based on the theoretic work of personalized federate learning pioneered in [57]. Firstly, we will make some assumptions on our models (parameters) akin to those in [57] with some differences specific to our scenario. One can refer [57] for more details.

Recall that our loss function is defined as follows:[10]

$$F(\hat{\mathbf{u}}, \mathbf{v}) = \frac{1}{n}\sum_{i=1}^{n} F_i(\hat{\mathbf{u}}, \mathbf{v}_i), \tag{15}$$

where $F_i(\hat{\mathbf{u}}, \mathbf{v}_i) = \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_i} f_i(\hat{\mathbf{u}}, \mathbf{v}_i, \mathbf{x})$. $\hat{\mathbf{u}}$ denotes the sharing parameters, i.e. parameters of s_i, while $\mathbf{v}_i$ denotes parameters preserved in each client. According to the structure of ZOOPFL, $\mathbf{v}_i$ is also decomposed into two parts: $\mathbf{v}_i = \mathbf{v}_{1,i} \cup \mathbf{v}_{2,i}$ corresponding to parameters of $\hat{\mathbf{z}}_i$ and r_i respectively, and as pointed out in main sections of our paper, we regroup our parameters as follows: for each device i, $\mathbf{u}_i := \hat{\mathbf{u}} \cup \mathbf{v}_{1,i}$, $\mathbf{v}_i := \mathbf{v}_{2,i}$.

[9] Different from [57], optimized parameters in ZOOPFL contain three parts and we utilize ZOO instead of gradients.

[10] As we will regroup parameters later, we use $\hat{\mathbf{u}}$ instead of $\mathbf{u}$ to make symbols consistent.

Assumption 1. *(Smoothness). For each device $i = 1, 2, \ldots n$, the object F_i is smooth, i.e., it is continuously differentiable and,*

1. $\mathbf{u}_i \to \nabla_{\mathbf{u}} F_i(\mathbf{u}_i, \mathbf{v}_i)$ *is $L_{\mathbf{u}}$-Lipschitz for all $\mathbf{v}_i$,*
2. $\mathbf{v}_i \to \nabla_{\mathbf{v}} F_i(\mathbf{u}_i, \mathbf{v}_i)$ *is $L_{\mathbf{v}}$-Lipschitz for all $\mathbf{u}_i$,*
3. $\mathbf{v}_i \to \nabla_{\mathbf{u}} F_i(\mathbf{u}_i, \mathbf{v}_i)$ *is $L_{\mathbf{uv}}$-Lipschitz for all $\mathbf{u}_i$, and,*
4. $\mathbf{u}_i \to \nabla_{\mathbf{v}} F_i(\mathbf{u}_i, \mathbf{v}_i)$ *is $L_{\mathbf{vu}}$-Lipschitz for all $\mathbf{v}_i$.*

Further, we assume for some $\chi > 0$ that

$$max\{L_{\mathbf{uv}}, L_{\mathbf{vu}}\} \leq \chi\sqrt{L_{\mathbf{u}} L_{\mathbf{v}}}. \tag{16}$$

Assumption 2. *(Bounded Gradient). For each device $i = 1, 2, \ldots n$, the object F_i has bounded gradient, that is, there exists $M_{\mathbf{u}}, M_{\mathbf{v}} > 0$ such that*

$$\begin{aligned} |\nabla_{\mathbf{u}} f_i(\mathbf{u}_i, \mathbf{v}_i, \mathbf{x})| &< M_{\mathbf{u}}, \\ and \quad |\nabla_{\mathbf{v}} f_i(\mathbf{u}_i, \mathbf{v}_i, \mathbf{x})| &< M_{\mathbf{v}} \quad \forall \mathbf{x} \in \mathcal{D}_i. \end{aligned} \tag{17}$$

Assumption 3. *(Bounded Variance). Let $\mathcal{D}_i$ denote a probability distribution over the data space $\mathcal{Z}$ on device i. There exist functions $G_{i,\mathbf{u}}$ and $G_{i,\mathbf{v}}$ which are unbiased estimates of $\nabla_{\mathbf{u}} F_i$ and $\nabla_{\mathbf{v}} F_i$ respectively. That is, for all $\mathbf{u}_i, \mathbf{v}_i$:*

$$\begin{aligned} \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_i}[G_{i,\mathbf{u}}(\mathbf{u}_i, \mathbf{v}_i, \mathbf{x})] &= \nabla_{\mathbf{u}} F_i(\mathbf{u}_i, \mathbf{v}_i), \\ and \quad \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_i}[G_{i,\mathbf{v}}(\mathbf{u}_i, \mathbf{v}_i, \mathbf{x})] &= \nabla_{\mathbf{v}} F_i(\mathbf{u}_i, \mathbf{v}_i). \end{aligned} \tag{18}$$

Furthermore, the variance of these estimators is at most $\sigma_{\mathbf{u}}^2$ and $\sigma_{\mathbf{v}}^2$ respectively. That is,

$$\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_i}||G_{i,\mathbf{u}}(\mathbf{u}_i, \mathbf{v}_i, \mathbf{x}) - \nabla_{\mathbf{u}} F_i(\mathbf{u}_i, \mathbf{v}_i)||^2 \leq \sigma_{\mathbf{u}}^2, \tag{19}$$

$$\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_i}||G_{i,\mathbf{v}}(\mathbf{u}_i, \mathbf{v}_i, \mathbf{x}) - \nabla_{\mathbf{v}} F_i(\mathbf{u}_i, \mathbf{v}_i)||^2 \leq \sigma_{\mathbf{v}}^2. \tag{20}$$

In this work, we usually take the particular form $G_{i,\mathbf{u}}(\mathbf{u}_i, \mathbf{v}_i, \mathbf{x}) = \nabla_{\mathbf{u}} f_i((\mathbf{u}_i, \mathbf{v}_i), \mathbf{x})$, which is the gradient of the loss on datatpoint $\mathbf{x} \sim \mathcal{D}_i$ under the model $(\mathbf{u}_i, \mathbf{v}_i)$, and similarly for $G_{i,\mathbf{v}}$.

As our model has a black box LLM, we can't get the gradient of parameters in this part. So we resort to zero-order optimization partially. In particular, we take differences of function values to estimate unknown gradients in that part. The resulting method is dubbed stochastic difference descent method. More precisely, Let G be a continuous function on $\mathbf{x}$, $\boldsymbol{\rho}$ be a fixed vector with the same dimension of $\mathbf{x}$ and its norm $||\boldsymbol{\rho}|| = \rho$, we denote $\Delta_{\boldsymbol{\rho},\mathbf{x}} G(x) := (G(\mathbf{x} + \boldsymbol{\rho}) - G(\mathbf{x}))/\rho$ or $(G(\mathbf{x} + \boldsymbol{\rho}) + G(\mathbf{x} - \boldsymbol{\rho}) - 2G(\mathbf{x}))/(2\rho)$ to be the difference of G at point $\mathbf{x}$ with step $\boldsymbol{\rho}$. Then the way we update $\mathbf{x}$ is similar to that in stochastic gradient descent method:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \gamma \Delta_{\boldsymbol{\rho},\mathbf{x}} G(\mathbf{x})|_{\mathbf{x}=\mathbf{x}_k}. \tag{21}$$

Let us denote $\tilde{G}_{i,\mathbf{u}}(\mathbf{u}_i, \mathbf{v}_i, \mathbf{x}) := \Delta_{\boldsymbol{\rho},\mathbf{u}} f_i((\mathbf{u}_i, \mathbf{v}_i), \mathbf{x}), \tilde{G}_{i,\mathbf{v}}(\mathbf{u}_i, \mathbf{v}_i, \mathbf{x}) := \Delta_{\boldsymbol{\rho},\mathbf{v}} f_i((\mathbf{u}_i, \mathbf{v}_i), \mathbf{x})$. Unlike Assumption. 3, $\tilde{G}_{i,\mathbf{u}}\mathbf{u}_i, \mathbf{v}_i, \mathbf{x})$ (resp.
$\tilde{G}_{i,\mathbf{v}}(\mathbf{u}_i, \mathbf{v}_i, \mathbf{x})$) is not an unbiased estimation of $\nabla_{\mathbf{u}} F_i(\mathbf{u}_i, \mathbf{v}_i)$ (resp. $\nabla_{\mathbf{v}} F_i(\mathbf{u}_i, \mathbf{v}_i)$). However, under Assumption. 1,Assumption. 2 and Assumption. 3, we have the following estimates:

Lemma 1. *(Bounded 1st and 2nd moments).*

$$||\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_i} \tilde{G}_{i,\mathbf{u}}(\mathbf{u}_i, \mathbf{v}_i, \mathbf{x}) - \nabla_{\mathbf{u}} F_i(\mathbf{u}_i, \mathbf{v}_i)|| \leq L_{\mathbf{u}}\rho, \tag{22}$$

$$||\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_i} \tilde{G}_{i,\mathbf{v}}(\mathbf{u}_i, \mathbf{v}_i, \mathbf{x}) - \nabla_{\mathbf{v}} F_i(\mathbf{u}_i, \mathbf{v}_i)|| \leq L_{\mathbf{u}}\rho. \tag{23}$$

Furthermore,

$$\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_i}||\tilde{G}_{i,\mathbf{u}}(\mathbf{u}_i, \mathbf{v}_i, \mathbf{x}) - \nabla_{\mathbf{u}} F_i(\mathbf{u}_i, \mathbf{v}_i))||^2 \leq 2L_{\mathbf{u}}^2\rho^2 + 2\sigma_{\mathbf{u}}^2, \tag{24}$$

$$\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_i}||\tilde{G}_{i,\mathbf{v}}(\mathbf{u}_i, \mathbf{v}_i, \mathbf{x}) - \nabla_{\mathbf{v}} F_i(\mathbf{u}_i, \mathbf{v}_i))||^2 \leq 2L_{\mathbf{v}}^2\rho^2 + 2\sigma_{\mathbf{v}}^2. \tag{25}$$

Proof. For each $\mathbf{x} \in \mathcal{D}_i$ there exists a $\mathbf{u}_i'$ between $\mathbf{u}_i$ and $\mathbf{u}_i + \boldsymbol{\rho}$ in each component such that $(\nabla_{\mathbf{u}} f_i(\mathbf{u}_i + \boldsymbol{\rho}, \mathbf{v}_i, \mathbf{x}) - \nabla_{\mathbf{u}} f_i(\mathbf{u}_i, \mathbf{v}_i, \mathbf{x}))/\rho = \nabla_{\mathbf{u}} f_i(\mathbf{u}_i', \mathbf{v}_i, \mathbf{x})$. Then by the smoothness assumption:

$$||\nabla_{\mathbf{u}} f_i(\mathbf{u}_i', \mathbf{v}_i, \mathbf{x}) - \nabla_{\mathbf{u}} f_i(\mathbf{u}_i, \mathbf{v}_i, \mathbf{x})|| \leq L_{\mathbf{u}}||\mathbf{u}_i' - \mathbf{u}_i|| \leq L_{\mathbf{u}}||\boldsymbol{\rho}|| = L_{\mathbf{u}}\rho. \tag{26}$$

Thus

$$\tilde{G}_{i,\mathbf{u}}(\mathbf{u}_i, \mathbf{v}_i, \mathbf{x}) - \nabla_{\mathbf{u}} F_i(\mathbf{u}_i, \mathbf{v}_i) = \tag{27}$$

$$\tilde{G}_{i,\mathbf{u}}(\mathbf{u}_i, \mathbf{v}_i, \mathbf{x}) - \nabla_{\mathbf{u}} f_i((\mathbf{u}_i, \mathbf{v}_i), \mathbf{x}) + \nabla_{\mathbf{u}} f_i((\mathbf{u}_i, \mathbf{v}_i), \mathbf{x}) - \nabla_{\mathbf{u}} F_i(\mathbf{u}_i, \mathbf{v}_i). \tag{28}$$

Taking expectation on both sides and applying (26) and $\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_i} \nabla_{\mathbf{u}} f_i((\mathbf{u}_i, \mathbf{v}_i), \mathbf{x}) = \nabla_{\mathbf{u}} F_i(\mathbf{u}_i, \mathbf{v}_i)$, we get

$$\begin{aligned} &\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_i} \tilde{G}_{i,\mathbf{u}}(\mathbf{u}_i, \mathbf{v}_i, \mathbf{x}) - \nabla_{\mathbf{u}} F_i(\mathbf{u}_i, \mathbf{v}_i) \\ &= \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_i} \tilde{G}_{i,\mathbf{u}}(\mathbf{u}_i, \mathbf{v}_i, \mathbf{x}) - \nabla_{\mathbf{u}} f_i((\mathbf{u}_i, \mathbf{v}_i), \mathbf{x}) \\ &\leq \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_i} |\tilde{G}_{i,\mathbf{u}}(\mathbf{u}_i, \mathbf{v}_i, \mathbf{x}) - \nabla_{\mathbf{u}} f_i((\mathbf{u}_i, \mathbf{v}_i), \mathbf{x})| \leq L_{\mathbf{u}} \rho. \end{aligned} \tag{29}$$

Taking absolute values on both sides completes the proof of (22). The same is true for (23).

For (24) and (25), we note that by Cauchy-Schwartz inequality

$$||\tilde{G}_{i,\mathbf{u}}(\mathbf{u}_i, \mathbf{v}_i, \mathbf{x}) - \nabla_{\mathbf{u}} F_i(\mathbf{u}_i, \mathbf{v}_i)||^2 \tag{30}$$

$$\leq 2||\tilde{G}_{i,\mathbf{u}}(\mathbf{u}_i, \mathbf{v}_i, \mathbf{x}) - \nabla_{\mathbf{u}} f_i(\mathbf{u}_i, \mathbf{v}_i, \mathbf{x})||^2 \tag{31}$$

$$+ 2||\nabla_{\mathbf{u}} f_i(\mathbf{u}_i, \mathbf{v}_i, \mathbf{x}) - \nabla_{\mathbf{u}} F_i(\mathbf{u}_i, \mathbf{v}_i)||^2. \tag{32}$$

Taking expectation on both sides and using (26) and Assumption. 3 complete the proof. The same is true for (25).

□

As our model has the form $F = r \circ J \circ s$ with s corresponding to the encoder, J corresponding to decoder and black-box LLM, and r corresponding to linear remapping, the following corollary is useful.

Corollary 2. *Let s_i, J_i, r_i be three continuously differentiable functions such that $f_i(\mathbf{u}_i, \mathbf{v}_i, \mathbf{x}) = r_i \circ J_i \circ s_i$, then we have*

$$||\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_i} \nabla r_{i,\mathbf{u}} \circ \Delta_{\boldsymbol{\rho}} J_{i,\mathbf{u}} \circ \nabla_{\mathbf{u}} s_{i,\mathbf{u}}(\mathbf{u}_i, \mathbf{v}_i, \mathbf{x}) - \nabla_{\mathbf{u}} F_i(\mathbf{u}_i, \mathbf{v}_i)|| \leq M_{\mathbf{u}}^2 L_{\mathbf{u}} \rho, \tag{33}$$

$$||\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_i} \nabla r_{i,\mathbf{u}} \circ \Delta_{\boldsymbol{\rho}} J_{i,\mathbf{u}} \circ \nabla_{\mathbf{v}_i} s_{i,\mathbf{u}}(\mathbf{u}_i, \mathbf{v}_i, \mathbf{x}) - \nabla_{\mathbf{v}} F_i(\mathbf{u}_i, \mathbf{v}_i)|| \leq M_{\mathbf{v}}^2 L_{\mathbf{v}} \rho. \tag{34}$$

Furthermore,

$$\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_i} ||\nabla r_{i,\mathbf{u}} \circ \Delta_{\boldsymbol{\rho}} J_{i,\mathbf{u}} \circ \nabla_{\mathbf{u}} s_{i,\mathbf{u}}(\mathbf{u}_i, \mathbf{v}_i, \mathbf{x}) - \nabla_{\mathbf{u}} F_i(\mathbf{u}_i, \mathbf{v}_i)||^2 \leq 2M_{\mathbf{u}}^4 L_{\mathbf{u}}^2 \rho^2 + 2\sigma_{\mathbf{u}}^2, \tag{35}$$

$$\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_i} ||\nabla r_{i,\mathbf{u}} \circ \Delta_{\boldsymbol{\rho}} J_{i,\mathbf{u}} \circ \nabla_{\mathbf{v}_i} s_{i,\mathbf{u}}(\mathbf{u}_i, \mathbf{v}_i, \mathbf{x}) - \nabla_{\mathbf{v}} F_i(\mathbf{u}_i, \mathbf{v}_i)||^2 \leq 2M_{\mathbf{v}}^4 L_{\mathbf{v}}^2 \rho^2 + 2\sigma_{\mathbf{v}}^2. \tag{36}$$

Proof. Using the bounded gradient assumption and following the steps in the proof of Lemma. 1 completes the proof. □

Finally, we make a gradient diversity assumption.

Assumption 4. *(Partial Gradient Diversity). There exists $\delta \geq 0$ and $\mu \geq 0$ such that for all $\mathbf{u}$ and $\mathbf{v}$,*

$$\frac{1}{n} \sum_{i=1}^{n} ||\nabla_{\mathbf{u}} F_i(\mathbf{u}_i, \mathbf{v}_i) - \nabla_{\mathbf{u}} F(\mathbf{u}, \mathbf{v})||^2 \leq \delta^2 + \mu^2 ||\nabla_{\mathbf{u}} F(\mathbf{u}, \mathbf{v})||^2. \tag{37}$$

Please refer [57] for more background on some of these assumptions.

B.1 Convergence Analysis of ZOOPFL

We give the error bounds results of Algorithm 1, thus theoretically establishing the convergence property.

In our case, we rename the parameters so that $\mathbf{u} := \hat{\mathbf{u}} \cup \mathbf{v}_1, \mathbf{v} := \mathbf{v}_2$. As per Appendix A.3 in [57], we use the constants

$$\begin{aligned}\sigma_{alt,1}^2 = \frac{\delta^2}{L_{\mathbf{u}}}\left(1-\frac{m}{n}\right) + \frac{2M_{\mathbf{u}}^4 L_{\mathbf{u}}^2\rho^2 + 2\sigma_{\mathbf{u}}^2}{L_{\mathbf{u}}} \\ + \frac{(2M_{\mathbf{v}}^4 L_{\mathbf{v}}^2\rho^2 + 2\sigma_{\mathbf{v}}^2)(m+\chi^2(n-m))}{L_{\mathbf{v}} n},\end{aligned} \tag{38}$$

$$\sigma_{alt,2}^2 = \frac{2M_{\mathbf{u}}^4 L_{\mathbf{u}}^2\rho^2 + 2\sigma_{\mathbf{u}}^2 + \delta^2}{L_{\mathbf{u}}}(1-\tau_{\mathbf{u}}^{-1}) \tag{39}$$

$$+ \frac{(2M_{\mathbf{v}}^4 L_{\mathbf{v}}^2\rho^2 + 2\sigma_{\mathbf{v}}^2)m}{L_{\mathbf{v}} n}(1-\tau_{\mathbf{u}}^{-1}) + \frac{\chi^2(2M_{\mathbf{v}}^4 L_{\mathbf{v}}^2\rho^2 + 2\sigma_{\mathbf{v}}^2)}{L_{\mathbf{v}}}. \tag{40}$$

and the definitions

$$\begin{aligned}\Delta_{\mathbf{u}}^{(t)} &= \frac{1}{n}\sum_{i=1}^{n} ||\nabla_{\mathbf{u}} F_i\left(\mathbf{u}_i^{(t)}, \mathbf{v}_i^{(t)}\right)||^2, \\ \textit{and,} \quad \Delta_{\mathbf{v}}^{(t)} &= \frac{1}{n}\sum_{i=1}^{n} ||\nabla_{\mathbf{v}} F_i\left(\mathbf{u}_i^{(t)}, \mathbf{v}_i^{(t)}\right)||^2.\end{aligned} \tag{41}$$

Theorem 1. *Suppose Assumption. 1,Assumption. 2,Assumption. 3 and Assumption. 4 hold and the learning rates in* ZOOPFL *are chosen as* $\gamma_{\mathbf{u}} = \eta/(L_{\mathbf{u}}\tau_{\mathbf{u}})$ *and* $\gamma_{\mathbf{v}} = \eta/(L_{\mathbf{v}}\tau_{\mathbf{v}})$*, with*

$$\eta \leq \textit{min}\left\{\frac{1}{24(1+\mu^2)}, \frac{m}{128\chi^2(n-m)}, \sqrt{\frac{m}{\chi^2 n}}\right\}. \tag{42}$$

Then, right after the training of T *epochs, ignoring absolute constants, we have*

$$\frac{1}{T}\sum_{t=0}^{T-1}\left(\frac{1}{L_{\mathbf{u}}}\mathbb{E}\left[\Delta_{\mathbf{u}}^{(t)}\right] + \frac{m}{nL_{\mathbf{v}}}\mathbb{E}\left[\Delta_{\mathbf{v}}^{(t)}\right]\right) \leq \frac{\Delta F_0}{\eta T} + \eta\sigma_{alt,1}^2 + \eta^2\sigma_{alt,2}^2. \tag{43}$$

Corollary 3. *An optimal learning rate is chosen as follows*

$$\begin{aligned}\eta = \left(\frac{\Delta F_0}{T\sigma_{alt,1}^2}\right)^{1/2} \bigwedge \left(\frac{\Delta F_0^2}{T^2\sigma_{alt,2}^2}\right)^{1/3} \bigwedge \frac{1}{1+\mu^2} \\ \bigwedge \frac{m}{\chi^2(n-m)} \bigwedge \sqrt{\frac{m}{\chi^2 n}}.\end{aligned} \tag{44}$$

We have, ignoring absolute constants,

$$\begin{aligned}&\frac{1}{T}\sum_{t=0}^{T-1}\left(\frac{1}{L_{\mathbf{u}}}\mathbb{E}\left[\Delta_{\mathbf{u}}^{(t)}\right] + \frac{m}{nL_{\mathbf{v}}}\mathbb{E}\left[\Delta_{\mathbf{v}}^{(t)}\right]\right) \leq \\ &\frac{(\Delta F_0\sigma_{alt,1}^2)^{1/2}}{\sqrt{T}} + \frac{(\Delta F_0^2\sigma_{alt,2}^2)^{1/3}}{T^{2/3}} \\ &+ \frac{\Delta F_0}{T}\left(1+\mu^2+\chi^2(\frac{n}{m}-1) + \sqrt{\chi^2\frac{n}{m}}\right).\end{aligned} \tag{45}$$

Proof. The proof is invoking Lemma 25 in [57] upon establishing Thm. 1 □

We will refer the readers to [57] for the proof of convergence of FedAlt algorithm therein. One of our novel differences to [57] is that our black-box model is secure so its gradient is invisible to us, leading us to consider zero-order (gradient free) optimization. Thus we first establish a result analogous to lemma 22 in [57] for zero-order optimization.

Lemma 2. *Consider $f : R^d \to R$ which is L-smooth, its norm of gradient is bounded by M and fix a $\mathbf{w}^{(0)} \in R^d$. Define the sequence $(\mathbf{w}^{(t)})$ of iterates produced by stochastic difference descent with step $\boldsymbol{\rho}$ and a fixed learning rate γ starting from $\mathbf{w}^{(0)}$:*

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \gamma \tilde{g}^{(t)}, \tag{46}$$

where $\tilde{g}$ is an unbiased estimation to $\Delta_{\boldsymbol{\rho},\mathbf{w}} f(\mathbf{w}^{(t)})$ (not an unbiased estimation to $\nabla f(\mathbf{w}^{(t)})$) with bounded variance $2M^4L^2\rho^2 + 2\sigma^2$. Fix a number τ of steps. If $\gamma \leq (\sqrt{2}\tau L)^{-1}$, we have the bound

$$\begin{aligned}&\sum_{t=0}^{\tau-1} ||\mathbf{w}^{(t)} - \mathbf{w}^{(0)}||^2 \\ &\leq 8\gamma^2\tau^2(\tau-1)||\nabla f(\mathbf{w}^{(0)})||^2 + 4\gamma^2\tau^2(\tau-1)(2M^4L^2\rho^2 + 2\sigma^2).\end{aligned} \tag{47}$$

Proof. If $\tau = 1$, we have nothing to prove. Assume now that $\tau \geq 2$. Let $\mathcal{F}^{(t)}$ be the sigma-algebra generated by $\mathbf{w}^{(t)}$ and denote $\mathbb{E}_t[\cdot] = \mathbb{E}[\cdot|\mathcal{F}^{(t)}]$. We will use the inequality

$$\begin{aligned}&\mathbb{E}_t||\tilde{g}^{(t)}||^2 = \mathbb{E}_t||\tilde{g}^{(t)} - \nabla f(\mathbf{w}^{(t)})||^2 + ||\nabla f(\mathbf{w}^{(t)})||^2 \\ &\leq 2M^4L^2\rho^2 + 2\sigma^2 + ||\nabla f(\mathbf{w}^{(t)})||^2.\end{aligned} \tag{48}$$

We then successively deduce,

$$\mathbb{E}_t||\mathbf{w}^{(t+1)} - \mathbf{w}^{(0)}||^2 \tag{49}$$
$$= ||\mathbf{w}^{(t)} - \mathbf{w}^{(0)} - \gamma\tilde{g}^{(t)}||^2 \tag{50}$$
$$\leq \left(1 + \frac{1}{\tau-1}\right)||\mathbf{w}^{(t)} - \mathbf{w}^{(0)}||^2 + \gamma^2\tau\mathbb{E}_t||\tilde{g}^{(t)}||^2 \tag{51}$$
$$\leq \left(1 + \frac{1}{\tau-1}\right)||\mathbf{w}^{(t)} - \mathbf{w}^{(0)}||^2 + 2\gamma^2\tau||\nabla f(\mathbf{w}^{(t)} - \nabla f(\mathbf{w}^{(0)})||^2 \tag{52}$$
$$+ 2\gamma^2\tau||\nabla f(\mathbf{w}^{(0)})||^2 + \gamma^2\tau(2M^4L^2\rho^2 + 2\sigma^2) \tag{53}$$
$$\leq \left(1 + \frac{1}{\tau-1} + 2\gamma^2\tau L^2\right)||\mathbf{w}^{(t)} - \mathbf{w}^{(0)}||^2 + 2\gamma^2\tau||\nabla f(\mathbf{w}^{(0)})||^2 \tag{54}$$
$$+ \gamma^2\tau(2M^4L^2\rho^2 + 2\sigma^2) \tag{55}$$
$$\leq \left(1 + \frac{2}{\tau-1}\right)||\mathbf{w}^{(t)} - \mathbf{w}^{(0)}||^2 + 2\gamma^2\tau||\nabla f(\mathbf{w}^{(0)})||^2 \tag{56}$$
$$+ \gamma^2\tau(2M^4L^2\rho^2 + 2\sigma^2). \tag{57}$$

Above, we used (a) the inequality $2\alpha\beta \leq \alpha^2/\delta^2 + \delta^2\beta^2$ for reals α, β, δ,(b) (48),(c) L-smoothness of f, and,(d) the condition on the learning rate.

Let $C = 2\gamma^2\tau||\nabla f(\mathbf{w}^{(0)})||^2 + \gamma^2\tau(2M^4L^2\rho^2 + 2\sigma^2)$. Unrolling the inequality and summing up the series for all $t \leq \tau - 1$

$$||\mathbf{w}^{(t)} - \mathbf{w}^{(0)}||^2 \leq C\sum_{j=0}^{t-1}\left(1 - \frac{2}{\tau-1}\right)^j \leq \frac{C}{2}(\tau-1)\left(1 + \frac{2}{\tau-1}\right)^t \tag{58}$$

$$\leq \frac{C}{2}(\tau-1)\left(1 + \frac{2}{\tau-1}\right)^{\tau-1} \leq \frac{C}{2}(\tau-1)e^2, \tag{59}$$

where we used the bound $(1 + 1/\alpha)^\alpha \leq e$ for all $\alpha > 0$. Summing over t and using the numerical bound $e^2 < 8$ completes the proof.

□

Lemma 3. *Consider the setting of Lemma. 2. If $\gamma \leq (2\tau L)^{-1}$, we have the bound*

$$||\mathbf{w}^{(\tau)} - \mathbf{w}^{(0)}||^2 \leq 16\gamma^2\tau^2||\nabla f(\mathbf{w}^{(0)})||^2 + 8\gamma^2\tau^2(2M^4L^2\rho^2 + 2\sigma^2). \tag{60}$$

Proof. Proceeding similar to the last proof (expect using $\delta = \tau$) gives us

$$\begin{aligned}
&\mathbb{E}_t||\mathbf{w}^{(t+1)} - \mathbf{w}^{(0)}||^2 \\
&\leq \left(1 + \frac{2}{\tau}\right)||\mathbf{w}^{(t)} - \mathbf{w}^{(0)}||^2 + 4\gamma^2\tau||\nabla f(\mathbf{w}^{(0)})||^2 \\
&+ 2\gamma^2\tau(2M^4L^2\rho^2 + 2\sigma^2).
\end{aligned} \tag{61}$$

Unrolling and summing up the sequence completes the proof, similar to that of Lemma. 2 □

With all the preparations, we now take the proof of Thm. 1.

Proof. Recall that our parameters are renamed as $\mathbf{u} := \hat{\mathbf{u}} \cup \mathbf{v}_1, \mathbf{v} := \mathbf{v}_2$.

The first step is to start with

$$F(\mathbf{u}^{(t+1)}, \mathbf{v}^{(t+1)}) - F(\mathbf{u}^{(t)}, \mathbf{v}^{(t)}) = F(\mathbf{u}^{(t)}, \mathbf{v}^{(t+1)}) - F(\mathbf{u}^{(t)}, \mathbf{v}^{(t)}) \tag{62}$$

$$+ F(\mathbf{u}^{(t+1)}, \mathbf{v}^{(t+1)}) - F(\mathbf{u}^{(t)}, \mathbf{v}^{(t+1)}). \tag{63}$$

The first line is referred to $\mathbf{v}$-step and the second $\mathbf{u}$-step. The smoothness assumption bounds the $\mathbf{u}$-step:

$$F(\mathbf{u}^{(t+1)}, \mathbf{v}^{(t+1)}) - F(\mathbf{u}^{(t)}, \mathbf{v}^{(t+1)}) \tag{64}$$

$$= \frac{1}{n}\sum_{i=1}^{n} F_i(\mathbf{u}_i^{(t+1)}, \mathbf{v}_i^{(t+1)}) - F_i(\mathbf{u}_i^{(t)}, \mathbf{v}_i^{(t)}) \tag{65}$$

$$\leq \frac{1}{n}\sum_{i=1}^{n}\left(\left\langle \nabla_{\mathbf{u}} F_i(\mathbf{u}_i^{(t)}, \mathbf{v}_i^{(t+1)}), \mathbf{u}_i^{(t+1)} - \mathbf{u}_i^{(t)}\right\rangle + \frac{L_{\mathbf{u}}}{2}||\mathbf{u}_i^{(t+1)} - \mathbf{u}_i^{(t)}||^2\right). \tag{66}$$

As discussed in [57], the most challenging thing is that the two terms in angle bracket $\langle\rangle$ are not independent random variables. Indeed, they both depend on the sampling $S^{(t)}$ of devices. The way to circumvent it is to introduce virtual full participation for $\mathbf{v}$-step update to eliminate this dependence structure and obtain a good estimate of the error it introduces. Briefly speaking, virtual full participation for parameters $\mathbf{v}$ is to assume all devices to update the $\mathbf{v}$ parameters (it is just technically assumed but not done in practice) that is independent of sampling of $S^{(t)}$ devices, breaking the dependence between $\mathbf{u}^{(t+1)}$ and $\mathbf{v}^{(t+1)}$. We ask the readers to read [57] for full details.

We use the notation $\tilde{\mathbf{v}}^{(t+1)}$ to denote the virtual update of $\mathbf{v}$. Then the proceeding inequality goes on as:

$$\left\langle \nabla_{\mathbf{u}} F_i(\mathbf{u}_i^{(t)}, \mathbf{v}_i^{(t+1)}), \mathbf{u}_i^{(t+1)} - \mathbf{u}_i^{(t)}\right\rangle + \frac{L_{\mathbf{u}}}{2}||\mathbf{u}_i^{(t+1)} - \mathbf{u}_i^{(t)}||^2 \tag{67}$$

$$= \left\langle \nabla_{\mathbf{u}} F_i(\mathbf{u}_i^{(t)}, \tilde{\mathbf{v}}_i^{(t+1)}), \mathbf{u}_i^{(t+1)} - \mathbf{u}_i^{(t)}\right\rangle + \frac{L_{\mathbf{u}}}{2}||\mathbf{u}_i^{(t+1)} - \mathbf{u}_i^{(t)}||^2 \tag{68}$$

$$+ \left\langle \nabla_{\mathbf{u}} F_i(\mathbf{u}_i^{(t)}, \mathbf{v}_i^{(t+1)}) - \nabla_{\mathbf{u}} F_i(\mathbf{u}_i^{(t)}, \tilde{\mathbf{v}}_i^{(t+1)}), \mathbf{u}_i^{(t+1)} - \mathbf{u}_i^{(t)}\right\rangle \tag{69}$$

$$\leq \left\langle \nabla_{\mathbf{u}} F_i(\mathbf{u}_i^{(t)}, \tilde{\mathbf{v}}_i^{(t+1)}), \mathbf{u}_i^{(t+1)} - \mathbf{u}_i^{(t)}\right\rangle + L_{\mathbf{u}}||\mathbf{u}_i^{(t+1)} - \mathbf{u}_i^{(t)}||^2 \tag{70}$$

$$+ \frac{1}{2L_{\mathbf{u}}}||\nabla_{\mathbf{u}} F_i(\mathbf{u}_i^{(t)}, \mathbf{v}_i^{(t+1)}) - \nabla_{\mathbf{u}} F_i(\mathbf{u}_i^{(t)}, \tilde{\mathbf{v}}_i^{(t+1)})||^2 \tag{71}$$

$$\leq \left\langle \nabla_{\mathbf{u}} F_i(\mathbf{u}_i^{(t)}, \tilde{\mathbf{v}}_i^{(t+1)}), \mathbf{u}_i^{(t+1)} - \mathbf{u}_i^{(t)}\right\rangle + L_{\mathbf{u}}||\mathbf{u}_i^{(t+1)} - \mathbf{u}_i^{(t)}||^2 \tag{72}$$

$$+ \frac{\chi^2 L_{\mathbf{v}}}{2}||\tilde{\mathbf{v}}^{(t+1)} - \mathbf{v}_i^{(t+1)}||^2. \tag{73}$$

The last two inequalities follow from Young's inequality and Lipschitzness of $\mathbf{v}_i \hookrightarrow \nabla_{\mathbf{u}} F_i(\mathbf{u}_i, \mathbf{v}_i)$ respectively.

The usage of virtual update is to ensure that $\tilde{\mathbf{v}}^{(t+1)}$ is independent of $S^{(t)}$. This allows us to take an expectation w.r.t the sampling $S^{(t)}$ of the devices.

Recall that under the new parameters $\mathbf{u}$ and $\mathbf{v}$, the only difference to the setting of FedAlt in [57] is that we need zero-order optimization to update $\mathbf{u}$ parameter instead of first-order gradient method. Thus, we can use the calculations in [57] with F replaced by F_i and then add them together from 1 to n to similarly arrive at the following expression:

$$\mathbb{E}_t\left[F(\mathbf{u}^{(t+1)},\mathbf{v}^{(t+1)})-F(\mathbf{u}^{(t)},\mathbf{v}^{(t+1)})\right] \tag{74}$$

$$\leq -\frac{\gamma_{\mathbf{u}}\tau_{\mathbf{u}}}{4}\mathbb{E}_t[\tilde{\Delta}_{\mathbf{u}}^{(t)}]+\frac{2\gamma_{\mathbf{u}}L_{\mathbf{u}}^2}{n}\sum_{i=1}^{n}\sum_{k=0}^{\tau_{\mathbf{u}}-1}\mathbb{E}_t||\tilde{\mathbf{u}}_{i,k}^{(t)}-\mathbf{u}^{(t)}||^2 \tag{75}$$

$$+4\gamma_{\mathbf{v}}^2\tau_{\mathbf{v}}^2L_{\mathbf{v}}(2M_{\mathbf{v}}^4L_{\mathbf{v}}^2\rho^2+2\sigma_{\mathbf{v}}^2)\chi^2(1-m/n) \tag{76}$$

$$+\frac{L_{\mathbf{u}}\gamma_{\mathbf{u}}^2\tau_{\mathbf{u}}^2}{m}(2M_{\mathbf{u}}^4L_{\mathbf{u}}^2\rho^2+2\sigma_{\mathbf{u}}^2+3\delta^2(1-\frac{m}{n}))+8\gamma_{\mathbf{v}}^2\tau_{\mathbf{v}}^2L_{\mathbf{v}}\chi^2(1-m/n)\Delta_{\mathbf{v}}^{(t)}, \tag{77}$$

note here $\tilde{\mathbf{u}}_{i,k}^{(t)}$ is $\mathbf{u}$-parameter updates via stochastic difference descent method rather than stochastic gradient descent method. We bound this term with Lemma. 2, invoking the assumption on gradient diversity. And then plugging the resulting estimate back in, we get

$$\mathbb{E}_t\left[F(\mathbf{u}^{(t+1)},\mathbf{v}^{(t+1)})-F(\mathbf{u}^{(t)},\mathbf{v}^{(t+1)})\right] \tag{78}$$

$$\leq -\frac{\gamma_{\mathbf{u}}\tau_{\mathbf{u}}}{8}\mathbb{E}_t[\tilde{\Delta}_{\mathbf{u}}^{(t)}]+\frac{L_{\mathbf{u}}\gamma_{\mathbf{u}}^2\tau_{\mathbf{u}}^2}{m}(2M_{\mathbf{u}}^4L_{\mathbf{u}}^2\rho^2+2\sigma_{\mathbf{u}}^2+2\delta^2(1-m/n)) \tag{79}$$

$$+4\gamma_{\mathbf{v}}^2\tau_{\mathbf{v}}^2L_{\mathbf{v}}(2M_{\mathbf{v}}^4L_{\mathbf{v}}^2\rho^2+2\sigma_{\mathbf{v}}^2)\chi^2(1-m/n) \tag{80}$$

$$+8\gamma_{\mathbf{v}}^2\tau_{\mathbf{v}}^2L_{\mathbf{v}}\chi^2(1-m/n)\Delta_{\mathbf{v}}^{(t)}+ \tag{81}$$

$$8\gamma_{\mathbf{v}}^2L_{\mathbf{u}}^3\tau_{\mathbf{u}}^2(\tau_{\mathbf{u}}-1)(2M_{\mathbf{u}}^4L_{\mathbf{u}}^2\rho^2+2\sigma_{\mathbf{u}}^2+2\delta_{\mathbf{u}}^2). \tag{82}$$

The bound on $\mathbf{v}$-step has exactly the same form as presented in [57] since when conditioned on $\mathbf{u}^{(t)},\mathbf{v}^{(t)}$ all functions used in updating $\mathbf{v}$ is continuously differentiable. Plugging this bound on $\mathbf{v}$-step into (62), we get

$$\mathbb{E}_t\left[F(\mathbf{u}^{(t+1)},\mathbf{v}^{(t+1)}-F(\mathbf{u}^{(t)},\mathbf{v}^{(t)}\right] \tag{83}$$

$$\leq -\frac{\gamma_{\mathbf{u}}\tau_{\mathbf{u}}}{8}\mathbb{E}_t[\tilde{\Delta}_{\mathbf{u}}^{(t)}]-\frac{\gamma_{\mathbf{v}}\tau_{\mathbf{v}}m}{16n}\mathbb{E}_t[\Delta_{\mathbf{v}}^{(t)}] \tag{84}$$

$$+4\gamma_{\mathbf{v}}^2L_{\mathbf{v}}\tau_{\mathbf{v}}^2(2M_{\mathbf{v}}^4L_{\mathbf{v}}^2\rho^2+2\sigma_{\mathbf{v}}^2)\left(\frac{m}{n}+\chi^2(1-m/n)\right) \tag{85}$$

$$+\frac{\gamma_{\mathbf{u}}^2L_{\mathbf{u}}\tau_{\mathbf{u}}^2}{m}(2M_{\mathbf{u}}^4L_{\mathbf{u}}^2\rho^2+2\sigma_{\mathbf{u}}^2+2\delta^2(1-m/n)) \tag{86}$$

$$+8\gamma_{\mathbf{u}}^3L_{\mathbf{u}}^2\tau_{\mathbf{u}}^2(\tau_{\mathbf{u}}-1)(2M_{\mathbf{u}}^4L_{\mathbf{u}}^2\rho^2+2\sigma_{\mathbf{u}}^2+2\delta^2) \tag{87}$$

$$+\frac{4\gamma_{\mathbf{v}}^3L_{\mathbf{v}}^2\tau_{\mathbf{v}}^2(\tau_{\mathbf{v}}-1)(2M_{\mathbf{v}}^4L_{\mathbf{v}}^2\rho^2+2\sigma_{\mathbf{v}}^2)m}{n}. \tag{88}$$

Taking an unconditional expectation, summing it over $t=0$ to $T-1$ and rearranging this gives

$$\frac{1}{T}=\sum_{t=0}^{T-1}\left(\frac{\gamma_{\mathbf{u}}\tau_{\mathbf{u}}}{8}\mathbb{E}[\tilde{\Delta}_{\mathbf{u}}^{(t)}]+\frac{\gamma_{\mathbf{v}}\tau_{\mathbf{v}}m}{16}\mathbb{E}[\Delta_{\mathbf{v}}^{(t)}]\right) \tag{89}$$

$$\leq \frac{\Delta F_0}{T}+4\gamma_{\mathbf{v}}^2L_{\mathbf{v}}\tau_{\mathbf{v}}^2(2M_{\mathbf{v}}^4L_{\mathbf{v}}^2\rho^2+2\sigma_{\mathbf{v}}^2)\left(\frac{m}{n}+\chi^2(1-m/n)\right) \tag{90}$$

$$+\frac{\gamma_{\mathbf{u}}^2L_{\mathbf{u}}\tau_{\mathbf{u}}^2}{m}(2M_{\mathbf{u}}^4L_{\mathbf{u}}^2\rho^2+2\sigma_{\mathbf{u}}^2+2\delta^2(1-m/n)) \tag{91}$$

$$+8\gamma_{\mathbf{u}}^3L_{\mathbf{u}}^2\tau_{\mathbf{u}}^2(\tau_{\mathbf{u}}-1)(2M_{\mathbf{u}}^4L_{\mathbf{u}}^2\rho^2+2\sigma_{\mathbf{u}}^2+2\delta^2) \tag{92}$$

$$+\frac{4\gamma_{\mathbf{v}}^3L_{\mathbf{v}}^2\tau_{\mathbf{v}}^2(\tau_{\mathbf{v}}-1)(2M_{\mathbf{v}}^4L_{\mathbf{v}}^2\rho^2+2\sigma_{\mathbf{v}}^2)m}{n}. \tag{93}$$

This is a bound in terms of the virtual updates $\tilde{\mathbf{v}}^{(t+1)}$. Similarly to the manipulations in [57] we can relate $\tilde{\Delta}_{\mathbf{u}}^{(t)}$ with $\Delta_{\mathbf{u}}^{(t)}$. [11], and finally we get:

$$\frac{1}{T}\sum_{t=0}^{T-1}\Big(\frac{\gamma_{\mathbf{u}}\tau_{\mathbf{u}}}{16}\mathbb{E}[\Delta_{\mathbf{u}}^{(t)}]+\frac{\gamma_{\mathbf{v}}\tau_{\mathbf{v}}m}{32n}\mathbb{E}[\Delta_{\mathbf{v}}^{(t)}]\Big) \tag{94}$$

$$\leq \frac{1}{T}\sum_{t=0}^{T-1}\Big(\frac{\gamma_{\mathbf{u}}\tau_{\mathbf{u}}}{9}\mathbb{E}[\tilde{\Delta}_{\mathbf{u}}^{(t)}]+\frac{\gamma_{\mathbf{v}}\tau_{\mathbf{v}}m}{16n}\mathbb{E}[\Delta_{\mathbf{v}}^{(t)}]\Big)+ \tag{95}$$

$$\gamma_{\mathbf{u}}\tau_{\mathbf{u}}\gamma_{\mathbf{v}}^2\tau_{\mathbf{v}}^2(2M_{\mathbf{u}}^4L_{\mathbf{u}}^2\rho^2+2\sigma_{\mathbf{u}}^2)\chi^2L_{\mathbf{u}}L_{\mathbf{v}} \tag{96}$$

$$\leq \frac{\Delta F_0}{T}+4\gamma_{\mathbf{v}}^2L_{\mathbf{v}}\tau_{\mathbf{v}}^2(2M_{\mathbf{v}}^4L_{\mathbf{v}}^2\rho^2+2\sigma_{\mathbf{v}}^2)\Big(\frac{m}{n}+\chi^2(1-m/n)\Big) \tag{97}$$

$$+\frac{\gamma_{\mathbf{u}}^2L_{\mathbf{u}}\tau_{\mathbf{u}}^2}{m}(2M_{\mathbf{u}}^4L_{\mathbf{u}}^2\rho^2+2\sigma_{\mathbf{u}}^2+2\delta^2(1-m/n)) \tag{98}$$

$$+8\gamma_{\mathbf{u}}^3L_{\mathbf{u}}^2\tau_{\mathbf{u}}^2(\tau_{\mathbf{u}}-1)(2M_{\mathbf{u}}^4L_{\mathbf{u}}^2\rho^2+2\sigma_{\mathbf{u}}^2+2\delta^2) \tag{99}$$

$$+\frac{4\gamma_{\mathbf{v}}^3L_{\mathbf{v}}^2\tau_{\mathbf{v}}^2(\tau_{\mathbf{v}}-1)(2M_{\mathbf{v}}^4L_{\mathbf{v}}^2\rho^2+2\sigma_{\mathbf{v}}^2)m}{n} \tag{100}$$

$$+\gamma_{\mathbf{v}}\tau_{\mathbf{u}}\gamma_{\mathbf{v}}^2\tau_{\mathbf{v}}^2(2M_{\mathbf{v}}^4L_{\mathbf{v}}^2\rho^2+2\sigma_{\mathbf{v}}^2)\chi^2L_{\mathbf{u}}L_{\mathbf{v}}. \tag{101}$$

Plugging in $\gamma_{\mathbf{u}}=\eta/(L_{\mathbf{u}}\tau_{\mathbf{u}})$ and $\gamma_{\mathbf{v}}=\eta/(L_{\mathbf{v}}\tau_{\mathbf{v}})$ completes the proof.

□

C Methodology

C.1 More Discussion on Design Logic

Design for the Decoder. In the first step, we pre-train the decoder to adapt to the corresponding dataset, enabling it to better understand the embeddings formed by the encoder. In the second step, freezing the decoder is motivated by two considerations: 1. reducing the computational cost of zero-order optimization; 2. minimizing the communication cost in federated learning.

The mechanism design of the semantic re-mapping process. We consider the scenario where the foundational model may not be suitable for a particular task, and we cannot modify it. Instead, we can only adjust the inputs and outputs of the model to make it compatible with the task. Regarding semantic re-mapping, the main consideration is that the foundational model may be trained for common tasks (such as the final layer of the base NLP classification model in Hugging Face being a random mapping). To adapt it to a specific task, a simple re-mapping is needed.

C.2 Summary

Here, we briefly provide a summary. Algorithm 1 describes the concrete process of ZOOPFL. s_i is consist of two components, q_i (i.e. the encoder) and o_i (i.e. the decoder), which are optimized via Adam in pretraining. Then, q_i is optimized via CGE during Input Surgery. Finally, r_i is optimized via Adam in Semantic-remapping.

D Experiments

D.1 Description of the Datasets

Computer vision datasets.

COVID-19 [63]. It is a public posterior-anterior chest radiography images dataset with four classes, including 1,281 COVID-19 X-Rays, 3,270 Normal X-Rays, 1,656 viral-pneumonia X-Rays, and 3,001 bacterial-pneumonia X-Rays. We split data into 20 clients via the Dirichlet distribution following [99] and each client has different distributions on the label space. In each client, only 10% of data

[11] More precisely, we simply take the same steps for F_i and then add all of them together.

Algorithm 1 ZOOPFL

Input: n clients' datasets $\{\mathcal{D}_i\}_{i=1}^n$
Output: Input surgery, $\{s_i\}_{i=1}^n$, semantic re-mapping, $\{r_i\}_{i=1}^n$, client-specific embeddings, $\{\hat{\mathbf{z}}_i\}_{i=1}^n$

for $t = 1$ to T **do** ▷ Step 1
  for $i = 1$ to n **do**
    Pretrain Client i according to $\ell_{MSE} = \mathbb{E}_{(\mathbf{x},y)\sim\mathbb{P}(\mathcal{D}_i)} \|o_i([q_i(\mathbf{x}), \hat{\mathbf{z}}_i]) - \mathbf{x}\|_2^2$
  Upload $\{w(s_i)\}_{i=1}^n$ to the server
  Aggregate weights, $w(s) = \frac{1}{n}\sum_{i=1}^n w(s_i)$
  Distribute client weight $w(s)$ to each client
for $t = 1$ to T **do**
  for $i = 1$ to n **do** ▷ Step 2
    for $k = 1$ to $d_1 + d_2$ **do** ▷ Compute Gradients via CGE
      $\tilde{\mathbf{z}_1} = \tilde{\mathbf{z}} + \rho\mathbf{e}_j, \ell_{\mathbf{x},1} = \ell_{cls}(r_i(g(o_i(\tilde{\mathbf{z}_1}, y))))$
      $\tilde{\mathbf{z}_2} = \tilde{\mathbf{z}} - \rho\mathbf{e}_j, \ell_{\mathbf{x},2} = \ell_{cls}(r_i(g(o_i(\tilde{\mathbf{z}_2}, y))))$
      Compute differences on embeddings, $\nabla_{\tilde{\mathbf{z}}}\mathcal{G}(\tilde{\mathbf{z}}) \approx \sum_{i=1}^{d_1+d_2} \frac{\ell_{\mathbf{x},2}-\ell_{\mathbf{x},1}}{2\times\rho}\mathbf{e}_j$
      Compute gradients, $\nabla_{q_i}\ell_1 = \frac{d\tilde{\mathbf{z}}}{dq_i}\frac{d\mathcal{G}}{d\tilde{\mathbf{z}}} \approx \frac{d\mathbf{z}}{dq_i}\nabla_{\tilde{\mathbf{z}}}\mathcal{G}(\tilde{\mathbf{z}})_1 \approx \frac{d\nabla_{\tilde{\mathbf{z}}}\mathcal{G}(\tilde{\mathbf{z}})_1\mathbf{z}}{dq_i}$
      Update parameters, $w(q_i^{new}) = w(q_i) - \gamma_1 \times \nabla_{q_i}\ell_1$
  Upload $\{w(q_i^{new})\}_{i=1}^n$ to the server
  Aggregate weights, $w(q) = \frac{1}{n}\sum_{i=1}^n w(q_i^{new})$
  Distribute q to clients
  for $i = 1$ to n **do** ▷ Step 3
    Train semantic re-mapping according to $\ell_2 = \mathbb{E}_{(\mathbf{x},y)\sim\mathbb{P}(\mathcal{D}_i)}\ell_{cls}(r_i(\mathcal{F}(\mathbf{x})), y)$

are utilized for training and the rest data are split evenly into two parts for validation and testing respectively.

APTOS [34]. It is an image dataset that judges the severity of diabetic retinopathy on a scale of 0 to 4. The original dataset contains 3,662 training images and 1,928 testing images but it suffers from heavily unbalanced. We only utilize the training part in our setting and we randomly choose 400 samples for classes 0 and 2. Our processed dataset contains 1658 samples. We split data into 20 clients and 20% of data serve for training similar to COVID-19.

Terra Incognita [4]. It is a common dataset that contains photographs of wild animals taken by camera traps at locations L100, L38, L43, and L46. It totally contains 24,788 samples with 10 classes. We randomly choose two locations, i.e. L46 and L100, to construct two benchmarks, i.e. Terra46 and Terra100. For these two benchmarks, we split data into 20 clients and 20% of data serve for training similar to COVID-19.

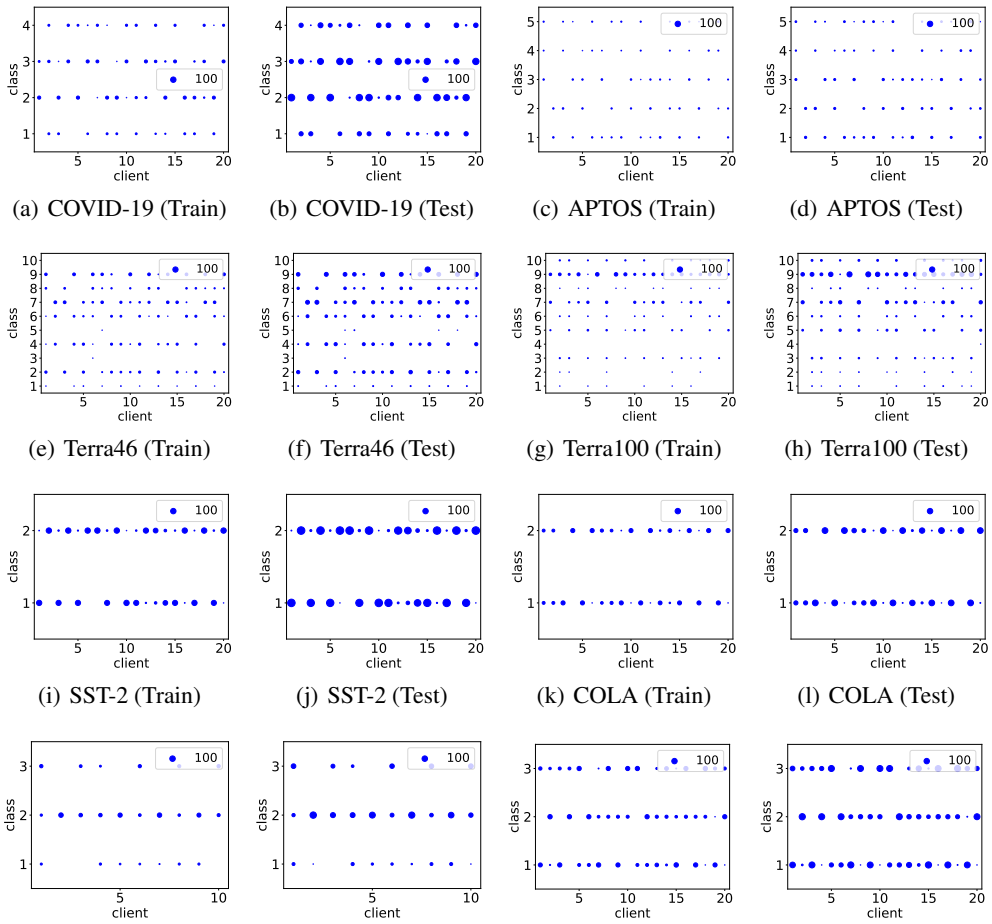
Natural language processing datasets.

SST-2 [81, 67]. The Stanford Sentiment Treebank contains sentences from movie reviews and the labels come from human annotations of their sentiment. It contains about 67k training samples with two classes. We choose sentences with the number of words ranging from 20 to 50 [12] and obtain 9763 samples. We split data into 20 clients and 20% of data serve for training similar to COVID-19.

COLA [81, 87]. The Corpus of Linguistic Acceptability contains English acceptability judgments drawn from books and journal articles on linguistic theory and it judges a sequence of words whether it is a grammatical English sentence. It contains 8.5k training samples with two classes. We choose sentences with the number of words less than 30 and obtain 5700 samples in total. For the class with more samples, we randomly choose parts to ensure balance. We split data into 20 clients and 20% of data serve for training similar to COVID-19.

Financial-phrasebank [52]. It consists of sentences from financial news categorized by sentiment. It contains 4840 samples with three classes, including positive, neutral, and negative. We choose sentences with the number of words less than 60 and obtain 3379 samples in total. We split data into 10 clients and 20% of data serve for training similar to COVID-19.

[12] We split sentences into words via spaces.



(a) COVID-19 (Train) (b) COVID-19 (Test) (c) APTOS (Train) (d) APTOS (Test)

(e) Terra46 (Train) (f) Terra46 (Test) (g) Terra100 (Train) (h) Terra100 (Test)

(i) SST-2 (Train) (j) SST-2 (Test) (k) COLA (Train) (l) COLA (Test)

(m) Financial. (Train) (n) Financial. (Test) (o) Flipkart (Train) (p) Flipkart (Test)

Figure 7: Data distributions. The size of dots represents the number of samples.

Flipkart [76]. This dataset contains information on product name, product price, rate, reviews, summary and sentiment. It has 205,053 samples with multiple labels. We choose sentiment analysis as our task and there can be three classes, including positive, neutral, and negative. We choose reviews with lengths more than 30 and we randomly choose parts of classes with more samples for balance. The processed dataset contains 3048 samples for each class. We split data into 20 clients and 20% of data serve for training similar to COVID-19.

D.2 Data Distributions

Figure 7 shows the complete data distributions on the rest benchmarks.

D.3 Model Structures

Table 3 shows details on auto-encoders. Please note that the encoder accounts for approximately half of the parameter count.

D.4 Additional Results

Table 4, Table 5, Table 6, and Table 7 show more detailed results on computer vision and natural language processing. We have added relevant experiments. 'Alone*' denotes locally trained black-box models on individual clients, while 'FedAVG*' represents direct federated learning without any

Table 3: The structures of auto-encoders.

CV			NLP(Linear)		
Layer(type)	Output Shape	#|Param|	Layer(type)	Output Shape	#|Param|
Conv2d-1	[-1,32,224,224]	896	Linear-1	[-1,128]	6,291,584
ReLU-2	[-1,32,224,224]	0	BatchNorm1d-2	[-1,128]	256
BatchNorm2d-3	[-1,32,224,224]	64	ReLU-3	[-1,128]	0
MaxPool2d-4	[-1,32,112,112]	0	Linear-4	[-1,256]	33,024
Conv2d-5	[-1,64,112,112]	18,496	BatchNorm1d-5	[-1,256]	512
ReLU-6	[-1,64,112,112]	0	ReLU-6	[-1,256]	0
BatchNorm2d-7	[-1,64,112,112]	128	Linear-7	[-1,96]	24,672
MaxPool2d-8	[-1,64,56,56]	0	BatchNorm1d-8	[-1,96]	192
Conv2d-9	[-1,128,56,56]	73,856	ReLU-9	[-1,96]	0
ReLU-10	[-1,128,56,56]	0	Linear-10	[-1,256]	33,024
BatchNorm2d-11	[-1,128,56,56]	256	BatchNorm1d-11	[-1,256]	512
MaxPool2d-12	[-1,128,28,28]	0	ReLU-12	[-1,256]	0
Conv2d-13	[-1,32,28,28]	36,896	Linear-13	[-1,128]	32,896
ReLU-14	[-1,32,28,28]	0	BatchNorm1d-14	[-1,128]	256
BatchNorm2d-15	[-1,32,28,28]	64	ReLU-15	[-1,128]	0
MaxPool2d-16	[-1,32,14,14]	0	Linear-16	[-1,49152]	6,340,608
Conv2d-17	[-1,8,14,14]	2,312	Tanh-17	[-1,49152]	0
ReLU-18	[-1,8,14,14]	0	NLP(LSTM)		
BatchNorm2d-19	[-1,8,14,14]	16	LSTM-1	[-1,64,128]	459,776
MaxPool2d-20	[-1,8,7,7]	0	Linear-2	[-1,96]	12,384
Linear-21	[-1,294]	115,542	LSTM-3	[-1,64,128]	132,096
ReLU-22	[-1,294]	0	Linear-4	[-1,64,768]	99,072
BatchNorm1d-23	[-1,294]	588	Tanh-5	[-1,64,768]	0
ConvTranspose2d-24	[-1,32,14,14]	2,336			
ReLU-25	[-1,32,14,14]	0			
BatchNorm2d-26	[-1,32,14,14]	64			
ConvTranspose2d-27	[-1,128,28,28]	36,992			
ReLU-28	[-1,128,28,28]	0			
BatchNorm2d-29	[-1,128,28,28]	256			
ConvTranspose2d-30	[-1,64,56,56]	73,792			
ReLU-31	[-1,64,56,56]	0			
BatchNorm2d-32	[-1,64,56,56]	128			
ConvTranspose2d-33	[-1,32,112,112]	18,464			
ReLU-34	[-1,32,112,112]	0			
BatchNorm2d-35	[-1,32,112,112]	64			
ConvTranspose2d-36	[-1,16,224,224]	4,624			
ReLU-37	[-1,16,224,224]	0			
BatchNorm2d-38	[-1,16,224,224]	32			
Conv2d-39	[-1,3,224,224]	51			

Table 4: Results on four computer vision benchmarks. **Bold** means the best.

DataSet	Method	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	AVG
COVID-19	ZS	18.36	33.66	1.94	48.06	3.38	47.32	53.17	0.00	3.40	51.71	3.90	43.69	40.78	13.17	54.37	0.00	11.65	36.59	0.97	58.05	26.21
	Ours	76.81	42.44	82.04	59.22	82.61	70.24	62.44	79.90	82.04	55.12	58.54	80.58	37.86	71.22	64.08	82.52	72.33	49.27	87.38	66.83	**68.17**
APTOS	ZS	0.00	53.12	0.00	41.94	0.00	45.16	25.00	32.26	48.39	0.00	29.03	12.50	65.62	0.00	51.61	0.00	40.62	0.00	0.00	38.71	24.20
	Ours	43.33	59.38	54.84	54.84	50.00	41.94	56.25	38.71	51.61	48.39	41.94	53.12	65.62	32.26	48.39	53.12	50.00	36.67	48.39	51.61	**49.02**
Terra46	ZS	21.93	21.55	25.22	12.39	34.48	6.09	13.04	25.44	33.33	18.42	46.09	6.96	35.96	16.38	41.74	5.31	25.00	7.96	21.74	21.74	22.04
	Ours	48.25	62.07	56.52	70.80	43.97	51.30	45.22	28.95	47.37	62.28	41.74	59.13	34.21	60.34	60.00	59.29	35.34	67.26	50.43	76.52	**53.05**
Terra100	ZS	14.13	9.89	7.69	9.57	5.56	6.38	9.78	6.45	6.52	16.48	4.40	11.70	11.83	7.78	12.90	7.69	11.96	7.61	9.78	12.22	9.52
	Ours	63.04	73.63	56.04	75.53	65.56	100.00	59.78	96.77	90.22	57.14	51.65	64.89	48.39	80.00	86.02	49.45	52.17	68.48	78.26	60.00	**68.85**

Table 5: Average results on CV. **Bold** means the best.

BenchMark	COVID-19	APTOS	Terra46	Terra100	AVG
ZS	26.21	24.20	22.04	9.52	20.49
Alone*	36.05	24.98	19.17	16.58	24.20
FedAVG*	36.61	24.51	33.19	26.50	30.20
Ours	**68.17**	**49.02**	**53.05**	**68.65**	**59.72**

Table 6: Average results on NLP. **Bold** means the best.

Benchmark	Method	ALBERT	BERT	DeBERTa	GPT2	AVG
SST-2	ZS	47.84	52.16	52.14	52.14	51.07
	Alone*	91.81	94.70	84.47	54.22	81.30
	FedAVG*	55.51	74.38	71.51	50.20	62.90
	Ours	94.70	94.72	94.70	94.70	**94.71**
COLA	ZS	50.22	50.22	49.87	48.82	49.78
	Alone*	69.12	68.55	76.89	68.20	70.69
	FedAVG*	52.02	58.77	50.75	56.49	54.51
	Ours	89.34	88.73	88.16	88.46	**88.67**
Financial	ZS	60.76	54.82	62.11	41.26	54.74
	Alone*	63.56	62.70	64.18	62.11	63.14
	FedAVG*	60.02	63.44	61.07	38.07	55.65
	Ours	68.91	68.54	68.69	71.79	**69.48**
Flipkart	ZS	32.26	29.43	33.27	34.16	32.28
	Alone*	48.59	33.40	56.06	33.88	42.98
	FedAVG*	35.86	34.44	43.39	39.21	38.23
	Ours	64.85	64.50	66.32	65.50	**65.29**

Table 7: Results on Financial-phrasebank. **Bold** means the best.

Backbones	Client	1	2	3	4	5	6	7	8	9	10	AVG
ALBERT	ZS	31.85	91.18	64.71	58.21	78.36	42.96	85.82	38.24	67.41	48.89	60.76
	Ours	52.59	99.26	62.50	58.96	87.31	52.59	94.78	50.00	77.78	53.33	**68.91**
BERT	ZS	42.22	70.59	58.82	49.25	57.46	47.41	67.16	43.38	58.52	53.33	54.82
	Ours	52.59	99.26	62.50	58.96	87.31	50.37	94.78	44.12	77.78	57.78	**68.54**
DeBERTa	ZS	25.93	99.26	62.50	58.96	87.31	36.30	94.78	31.62	77.78	46.67	62.11
	Ours	52.59	99.26	66.18	58.96	87.31	42.96	94.78	46.32	77.78	60.74	**68.69**
GPT2	ZS	41.48	41.18	49.26	37.31	43.28	37.04	48.51	33.82	42.96	37.78	41.26
	Ours	52.59	99.26	75.00	61.94	87.31	52.59	91.04	50.74	81.48	65.93	**71.79**

Table 8: Resource Consumption.

Phase	Metric	Foundation Models	ALB.	BERT	DeB.	GPT2
Inference	FLOPS	Base	5.4475E+09	5.4413E+09	7.2483E+09	5.0627E+06
		Ours(Linear)	5.4497E+09	5.4540E+09	7.2611E+09	1.7772E+07
		Ours(LSTM)	5.4656E+09	5.4856E+09	7.2927E+09	4.9377E+07
	Time (S)	Base	0.4278	0.3308	0.6475	0.6107
		Ours(Linear)	0.5856	0.4767	0.5887	0.5321
		Ours(LSTM)	0.7582	0.6028	0.7143	0.6753
	Metric	Foundation Models	ALB.	BERT	DeB.	GPT2
Train	GPU Memory (MB)	Base	5742	5380	8356	7874
		Ours(Linear)	1718	3086	3346	3838
		Ours(LSTM)	1738	2076	2336	2718
	Metric	Foundation Models	ALB.	BERT	DeB.	GPT2
/	Storage (M)	Base	45	418	532	475
		Ours(Linear)	54	467	580	524
		Ours(LSTM)	46	421	534	478

personalization modules. Please note that the method of fine-tuning local inputs and outputs is also proposed by us. Thus we do not include these two comparison methods in the main paper.

D.5 Resource Consumption Analysis

Table 8 shows resource consumption on other metrics. Since our model contains the foundation model on each client, it will slightly consume more FLOPS, inference time (running ten times to calculate the time.), and storage. We can observe that, compared to the foundational model, the incremental changes can be negligible, and sometimes even slightly less resource-intensive (possibly due to the instability of the computing environment). Please note that it is the incremental consumption that we are truly concerned about. In terms of GPU consumption, although our model is larger, it consumes less resources. It is reasonable that our approach consumes less GPU since we do not need to compute and store the gradients of foundational models.

D.6 Visualization Study

Figure 8 describes the visualization of both the original images and the corresponding restored images via the Autoencoder on Terra100. We can observe that the recovered images almost eliminate the corresponding category information. However, they generate distinct patterns for different categories.

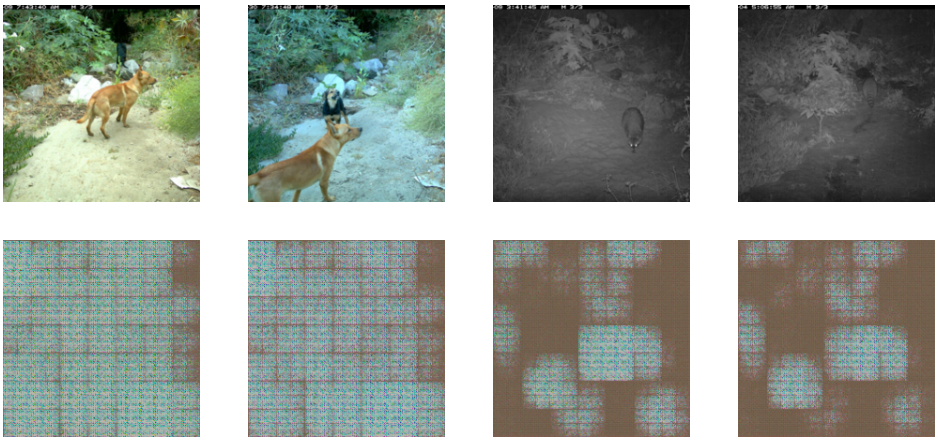


Figure 8: Visualizations of both the original images and the corresponding restored images via the Autoencoder on Terra100. The upper row is the original figures while the bottom row is the corresponding restored images via the Autoencoder. The first two columns are of the fourth class while the last two columns are of the eighth class.

Therefore, we can even attempt to upload processed data to foundation models in the server and obtain feedback outputs for remapping if data privacy and security could not be too strict and important.

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 main claims have reflected the paper's contributions and scope.

   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 have included limitations in the last 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: [Yes]

Justification: We have provided the full set of assumptions and a complete proof in the appendix.

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: The implementation details can be found in the appendix.

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: We have included the open access to code.

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, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We have included experimental settings/details.

   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: We have included sufficient experimental results and analysis.

   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: We have included resource analysis.

   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: The research is with the NeurIPS Code of Ethics.

   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: We have discussed broader 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: The paper poses no such risksthe paper poses no such risks

    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 have cited the original paper that produced the code package or dataset.

    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: [NA]

Justification: The paper does not release new assets.

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: The paper does not involve crowdsourcing nor research with human subjects.

    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 paper does not involve crowdsourcing nor research with 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.