FedGTST: Boosting Global Transferability of Federated Models via Statistics Tuning

Evelyn Ma, Chao Pan, Rasoul Etesami, Han Zhao, Olgica Milenkovic
University of Illinois Urbana-Champaign
{pingm, chaopan2, etesami1, hanzhao, milenkov}@illinois.edu

Abstract

The performance of Transfer Learning (TL) significantly depends on effective pretraining, which not only requires extensive amounts of data but also substantial computational resources. As a result, in practice, it is challenging to successfully perform TL at the level of individual model developers. Federated Learning (FL) addresses these challenges by enabling collaborations among individual clients through an indirect expansion of the available dataset, distribution of the computational across different clients, and privacy-preserving communication mechanisms. Despite several attempts to design effective transferable FL approaches, several important issues remain unsolved. First, existing methods primarily focus on optimizing transferability within local client domains, thereby ignoring transferability across clients. Second, most approaches focus on analyzing indirect transferability metrics, which does not allow for accurate assessment of the final target loss and the degree of transferability. To address these issues, we introduce two important FL features into the model. The first boosts transferability via an exchange protocol between the clients and the server that includes information about cross-client Jacobian (gradient) norms. The second feature promotes an increase of the average of the Jacobians of the clients at the server side, which is subsequently used as a local regularizer that reduces the cross-client Jacobian variance. A rigorous analysis of our transferable federated algorithm, termed FedGTST (Federated Global Transferability via Statistics Tuning), reveals that increasing the averaged Jacobian norm across clients and reducing the Jacobian variance ensures tight control of the target loss. This insight leads to an upper bound on the target loss of transferable FL regarding the source loss and source-target domain discrepancy. Empirically, experiments on public benchmarks show that FedGTST significantly outperforms other baselines, such as FedSR.

1 Introduction

Transfer Learning (TL) has received significant interest in the machine learning community due to its ability to extract representative features from source tasks and use them to improve the generalization capability on related target domain problems [48, 52]. In addition to boosting the performance of a target domain model, TL also reduces the computational cost of fine-tuning the target domain model. Nevertheless, effective source pretraining in TL is practically challenging for individual model developers because it requires access to large datasets as well as significant computational resources [35]. To resolve this problem, one can leverage *Federated Learning* (FL), which refers to decentralized learning protocols used in mobile and IoT devices [18]. FL not only increases access to multiple datasets in a decentralized manner and alleviates the computational burden of individual clients, but it also protects the privacy of local data [32]. As a result, a number of recent works have outlined methods for transferable FL, including FedADG (Federated Adversarial Domain Generalization) [55], FedCDG (Federated Contrastive Domain Generalization) [53], FedSR

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

 https://doi.org/10.52202/079017-1233

(Federated Simple Representations) [34], FedIIR (Federated Implicit Invariant Relationships) [12], FedCCST (Federated Cross-Client Style Transfer) [4], FedMM (Federated Adversarial Domain Adaptation) [42] and StableFDG (Stable Federated Domain Generalization) [36]. Despite the promising preliminary findings provided by the techniques above, several combinations of important issues remain unsolved across the spectrum of methods.

For **privacy leakage**, the limitations of existing methods include: **1.** FedADG forces each client source domain to align its representation distribution with that of the target domain, and therefore violates data privacy because source domains are given access to the target domain in order to perform the alignment; **2.** FedCCST boosts global transferability by increasing local diversity to avoid local overfitting. It therefore requires clients to share their local representations with each other and this information is subsequently used for local data augmentation. This is a direct violation of FL privacy constraints; **3.** StableFDG expands local data diversity by sharing style statistics (i.e., representations, including means and variances). This clearly leads to the leakage of local privacy-sensitive information.

For **local overfitting**, the shortcomings of a group of the methods above are as follows: **1.** FedSR learns a simple representation through regularization during local training, by exploiting the similarity between the representation and the data, given the labels. However, since the regularized local training relies completely on local structures (i.e., local models, representations, labels, data), it leads to overfitting of local distributions, and thus has limited capability to learn cross-client invariant features, which is key for global transferability; **2.** FedCDG uses a contrastive local regularizer on representations generated by various samples within the same class. This leads to overfitting in the local domain since no cross-client information is exploited.

For **communication complexity**, we observe that: **1.** FedIIR is suboptimal. Although it mitigates the problem of privacy violation and avoids local overfitting by adding a local regularizer capturing the distance between the local gradient and the global gradient, it requires communicating gradients between the clients and the server and therefore doubles the communication cost compared to baselines (additionally, FedIIR performs well for a large number of clients, but offers average performance when this number is small); **2.** Similar communication complexity problems are faced by FedCCST and StableFDG, which rely on communicating styles (i.e., representations); **3.** FedMM requires significant communication overhead for adaptation using distribution-matching techniques [58], which involves solving an intractable non-convex-non-concave optimization problem [56].

Finally, prior works mostly **lack explicit theoretical analyses of global transferability:** they do not tend to quantify the performance/loss of the pretrained model fine-tuned on the target domain.

In summary, perhaps the most important unresolved problem with known transferable FL models (with the exception of FedIIR) is that they use centralized TL approaches during local training, and do not fully exploit features specific to federated learning (for details, see also the discussion in Section 2).

Our contributions. We describe what is, to the best of our knowledge, the first approach to federated transfer learning termed *Federated Global Transferability via Statistics Tuning (FedGTST)* that simultaneously alleviates the above issues faced by existing methods. Our main contributions can be summarized as follows.

1. We use a new regularizer that encodes cross-client statistics and forces the local training process to tune the global statistics in a "direction" that improves global transferability rather than just local transferability. This is achieved through subtractions of global norms of Jacobians (gradients) communicated by the server.
2. We suggest to only communicate scalars, more precisely, *Jacobian norms*, which introduces a negligible communication overhead in the overall model exchange protocol.
3. We ensure that our communication schemes do not allow uncontrolled access to data and thereby ensure data privacy.
4. We rigorously prove that even though only small discrepancies among local gradients may exist upon regularization, transferability can be low as regularization can impede the growth of the gradient norm. To boost the Jacobian norm, we implement specialized protocols at both the client and server levels. Finally, we establish relevant bounds on the transferability loss for this setting.

The main technical insights provided by our analysis are as follows. Two FL-specific factors, a small *cross-client Jacobian variance* and larger *cross-client Jacobian norm* are indicative of good transferability. These factors are *direct* performance indicators, unlike *indirect factors* (e.g., feature invariance) which only suggest improved transferability. Our findings are based on the first *direct measure* of transferability, which equals the loss on the target domain incurred by the pretrained federated model. The FL-specific factors govern the bounds on the loss and therefore allow one to control them for better transferability. We validate these findings through extensive experiments which show that FedGTST outperforms methods such as FedSR and FedIIR by as much as 10%.

2 Related Work

FL is a machine learning paradigm in which multiple entities collaborate to train a global model without sharing their local data (see [15] for a comprehensive review). FL has gained significant attention due to its potential to address privacy concerns while enabling large-scale collaborative learning [44]. Relevant to this work, [32] proposed the Federated Averaging (FedAvg) algorithm, which aggregates model updates from multiple client devices to train a global model. Another relevant line of work [51] introduced FedProx, a federated optimization algorithm that incorporates proximal terms to handle non-iid data distributions. FL methods nevertheless still face several challenges. One challenge is dealing with highly heterogeneous local datasets [26] , for which the recent work FedImpro[46] proposed leveraging aggregated feature distributions to address client drift. Another challenge is the communication overhead incurred during the aggregation of model updates [2]. Minimizing communication costs while ensuring convergence and data privacy remain active topics of research in FL. Also, many FL solutions primarily emphasize performance in the client domain without considering the performance of the model on unseen domains.

TL is a powerful machine learning technique that allows models to leverage knowledge gained from one task to improve performance on another related task [35]. TL has been widely adopted in various domains such as computer vision, natural language processing, and speech recognition, where labeled data may be scarce or expensive to acquire [47, 48]. A common approach in TL involves fine-tuning a pre-trained model on a target task using a small amount of labeled data, which often leads to improved generalization and faster convergence compared to training from scratch [52]. Recent works in TL have focused on developing more effective algorithms, such as domain adaptation methods that address the discrepancy between the source and target domains [11]. Additionally, TL techniques have been used to handle tasks with limited amounts of labeled data through techniques like semi-supervised and self-supervised learning [40]. TL still faces challenges such as negative transfer, where information from the source task actually degrades performance on the target task; and, it requires careful selecting of appropriate pretrained models and transfer strategies for specific tasks and domains [35]. Current TL methods often require that one entity possesses knowledge of all data, violating the privacy requirements of FL. Moreover, we comment on *Gradient Matching in TL* in Appendix I.1

Transferable Federated Learning (TFL) is an emerging research area at the intersection of FL and TL. One of earliest contributions to the field, *FedADG* [55], encourages the transferablity of FL through adversarial local training. However, the work does not provide theoretical guarantees, and existing studies [50] indicate that adversarial robustness does not necessarily lead to better transferability. Other methods, such as *FedSR* [34] and *FedCDG* [53], enhance transferability by adapting standard representation learning from a single-agent to a federated setting; they do not incorporate FL-specific features (i.e., instructions provided by the server, cross-client model properties etc). Note that although *FedSR* has successfully included centralized invariant feature learning into FL, it uses centralized methods locally and then shares information with the global model, and thereby does not fully exploiting FL capabilities. Thus, using FedSR, each client can learn very different representations that are hard to aggregate at the central server. More precisely, FedSR does not communicating information that can help improve the transferability of the global model. Among all the previously discussed methods (FedADG, FedCDG, FedSR, FedIIR, FedCCST, and StableFDG), FedIIR is the closest to our approach and may be seen as a special case of our method which has better performance, smaller communication complexity and comes with provable global transferability guarantees. Furthermore, we discuss the distinctions and connections between TFL and Generalization of FL, a topic potentially relevant to TFL, in Appendix I.2.

 https://doi.org/10.52202/079017-1233

3 Preliminaries

General Supervised Learning Settings. We denote the data space by $\mathcal{X}$, the feature space by $\mathcal{Z}$, and the label space by $\mathcal{Y}$. A model $h : \mathcal{X} \to \mathcal{Y}$ typically takes the form $h = g \circ f$, where $f : \mathcal{X} \to \mathcal{Z}$ is a feature extractor and $g : \mathcal{Z} \to \mathcal{Y}$ is a classifier. Denote the function class for the entire model, the feature extractor and the classifier by $\mathcal{H}, \mathcal{F}, \mathcal{G}$, respectively, so that $h \in \mathcal{H}, f \in \mathcal{F}, g \in \mathcal{G}$. Denote the weights of model $\psi \in \{f, g, h\}$ as w_ψ. Given a loss function $l : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ and a domain distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, the population loss $L_{\mathcal{D}}(h)$ is defined as

$$L_{\mathcal{D}}(h) := \mathbb{E}_{(x,y)\sim\mathcal{D}}\, l\left(h(x), y\right). \tag{1}$$

General Framework of TFL. In TL, the two typical learning phases are: a) pretraining on the source domain; and, b) finetuning on the target domain. In the context of TFL, pretraining is conducted via FL over source (local) domains, while the global model is trained and then finetuned on the target domain during the second phase. In both phases, supervised learning is performed with full access to the labels. More details are provided next.

Pretraining Phase in TFL: FL on Source (Local) Domains. The source domain is a composition of the agents' local domains, $\{\mathcal{D}^{(k)}\}$, with $k \in [K]$ denoting the client index and K representing the total number of clients. The source loss is defined as the standard federated loss on the source domains.

$$L_{src}(g \circ f) := \frac{1}{K}\sum_{k=1}^{K} L_{\mathcal{D}^{(k)}}(g \circ f). \tag{2}$$

Let $h^* = g^* \circ f^*$ be an optimal global solution for the objective (2). In FL approaches, the problem solution is the result of the central server's aggregation of local models into a global one. We denote the local solutions involved in creating the optimal global solution ψ^* ($\psi \in \{f, g, h\}$) by $\{\psi^{*(k)}\}$; through averaging aggregation, we obtain the optimal global weights $w^*_\psi = \frac{1}{K}\sum_k w^{*(k)}_\psi$.

Finetuning Phase in TFL: Supervised Finetuning on the Target Domain. Upon obtaining the optimal pretrained global solution $h^* = g^* \circ f^*$, the pretrained feature extractor f^* is fixed and applied to the target domain $\mathcal{D}_T$. The target loss is defined as the loss on the target domain $\mathcal{D}_T$, i.e.,

$$L_{tgt}(g \circ f^*) := L_{\mathcal{D}_T}(g \circ f^*). \tag{3}$$

Through finetuning, a new classifier $g^*_T := \arg\inf_{g\in\mathcal{G}} L_{tgt}(g \circ f^*)$ is determined by minimizing the target objective (3).

Transferability Assessment. With a slight abuse of notation, we define the optimal target loss as

$$L^*_{tgt} := L_{tgt}(g^*_T \circ f^*).$$

We formally define the *measure of transferability of TFL* as the optimal target loss L^*_{tgt}, as it *directly* reflects the performance of a transferable model on the target domain. A smaller L^*_{tgt}, or a tighter bound on it, indicates better transferability.

4 Theoretical Bounds on the Target Loss

4.1 A General Bound Based on Discrepancy/Divergence

We start with Definitions 1 and 2 borrowed from existing TL studies that characterize the domain discrepancy, and then propose a new domain divergence tailored to Transferable FL (TFL), including the cross-client discrepancy in Definition 2 and the source-target discrepancy in Definition 3.

Definition 1 ($\mathcal{G}, \mathcal{F}$-discrepancy [50])**.** Given a classifier class $\mathcal{G}$, a feature extractor class $\mathcal{F}$, the source domain $\mathcal{D}_S$ and the target domain $\mathcal{D}_T$, with a slight abuse of notation, the $(\mathcal{G}, \mathcal{F})$-discrepancy is defined as

$$d_{\mathcal{G},\mathcal{F}}(\mathcal{D}_S, \mathcal{D}_T) := \sup_{f\in\mathcal{F}} \left| \inf_{g\in\mathcal{G}} L_{\mathcal{D}^{(k)}}(g \circ f) - \inf_{g\in\mathcal{G}} L_{\mathcal{D}_T}(g \circ f) \right|. \tag{4}$$

Remark. The $\mathcal{G}, \mathcal{F}$-discrepancy has also been used in the analysis of domain adaptation [50].

We next adapt the $\mathcal{H}$-discrepancy to measure the cross-client domain discrepancy in transferable FL (Definition 2), and tailor the $\mathcal{G}, \mathcal{F}$-discrepancy (Definition 1) to measure the source–target discrepancy in TFL (Definition 3).

Definition 2 (Cross-Client Divengence for TFL)**.** Given a model class $\mathcal{H}$ and federated local domains $\mathcal{D}_S^{fed} = \{\mathcal{D}^{(k)}\}_{k\in[K]}$, with $d_{\mathcal{H}}(\cdot,\cdot)$ defined as Definition 5, the intra-client discrepancy is defined as

$$\overline{d}_{\mathcal{H}}(\mathcal{D}_S^{fed}) := \frac{1}{K(K-1)} \sum_{k_1 \neq k_2} d_{\mathcal{H}} \left(\mathcal{D}_S^{(k_1)}, \mathcal{D}_S^{(k_2)} \right). \tag{5}$$

Note that $\overline{d}_{\mathcal{H}}(\mathcal{D}_S)$ equals the average of $\mathcal{H}$-discrepancies over all local domain pairs and therefore measures the intra-discrepancy on the non-iid distributed source domains. When source domains are iid across clients, we have $\overline{d}_{\mathcal{H}}(\mathcal{D}_S) = 0$.

Definition 3 (Source-Target Discrepancy for TFL)**.** Given a classifier class $\mathcal{G}$, a feature extractor class $\mathcal{F}$, federated local domains $\mathcal{D}_S^{fed} = \{\mathcal{D}^{(k)}\}_{k\in[K]}$, and the target domain $\mathcal{D}_T$, with slight abuse of notation, the federated $(\mathcal{G},\mathcal{F})$-discrepancy is defined as

$$d_{\mathcal{G},\mathcal{F}}(\mathcal{D}_S^{fed}, \mathcal{D}_T) := \frac{1}{K} \sum_{k\in[K]} d_{\mathcal{G},\mathcal{F}}(\mathcal{D}^{(k)}, \mathcal{D}_T). \tag{6}$$

Based on the two TFL-specific domain discrepancy definitions (i.e., Definition 2 and 3), we derive a general bound on the TFL loss in Theorem 1. The TFL-specific source-target discrepancy (Definition 3) is further used in Theorem 2 which presents a bound on the target loss using cross-client statistics. For our theoretical analyses, we need the following common assumptions.

Assumption 4.1 (Convexity and Smoothness)**.** *We assume that the loss function l satisfies two conditions: (1) l is convex w.r.t. w_h; (2) l is Lipschitz smooth for w_h with a constant $\alpha > 0$.*

Remark. Assumption 4.1 is easy to meet in practice, and it arises in many linear models (linear regression, SVM etc).

Theorem 1 (Bound Based on TFL-specific Domain Discrepancy)**.** *Under Assumptions 4.1 (Convexity and Smoothness), the optimal target loss is bounded as*

$$L_{tgt}^* \leq \frac{1}{K} \sum_{k=1}^{K} \left[L_{\mathcal{D}^{(k)}} \left(h^{*(k)} \right) \right] + \overline{d}_{\mathcal{H}}(\mathcal{D}_S^{fed}) + d_{\mathcal{G},\mathcal{F}}(\mathcal{D}_S^{fed}, \mathcal{D}_T), \tag{7}$$

where $h^{*(k)}$ denotes the optimal local model of client k (see Section 3).

Semantic Interpretation. In Theorem 1 , the optimal target loss L_{tgt}^* is bounded by the sum of three terms on the RHS of Equation 7: (1) the averaged optimal local loss $\frac{1}{K}\sum_{k=1}^{K}\left[L_{\mathcal{D}^{(k)}}\left(h^{*(k)}\right)\right]$; (2) the intra-discrepancy of the source domain $\overline{d}_{\mathcal{H}}(\mathcal{D}_S^{fed})$; (3) the discrepancy between the source and target domains, $d_{\mathcal{G},\mathcal{F}}(\mathcal{D}_S^{fed}, \mathcal{D}_T)$. Therefore, the bound on the optimal target loss can be tightened by making the optimal source loss smaller, and by lowering the intra-source and source-target discrepancy. The later two terms can be controlled through regularization as detailed below.

Tightening the Bound via Regularization over $\mathcal{F}$. Given two feature extractor function classes $\mathcal{F}_1$ and $\mathcal{F}_2$, if $\mathcal{F}_1 \subset \mathcal{F}_2$, we have $\mathcal{H}_1 \subset \mathcal{H}_2$ where $\mathcal{H}_1 = \mathcal{G} \times \mathcal{F}_1$ and $\mathcal{H}_2 = \mathcal{G} \times \mathcal{F}_2$. From Definition 2 and 3, it is straightforward to see that $\overline{d}_{\mathcal{H}_1}(\mathcal{D}_S^{fed}) \leq \overline{d}_{\mathcal{H}_2}(\mathcal{D}_S^{fed})$ and $d_{\mathcal{G},\mathcal{F}_1}(\mathcal{D}_S^{fed}, \mathcal{D}_T) \leq d_{\mathcal{G},\mathcal{F}_2}(\mathcal{D}_S^{fed}, \mathcal{D}_T)$. This indicates that any general regularizer on $\mathcal{F}$ can lead to a decrease in the latter two terms of the RHS of Theorem 1. However, shrinking the expressive power of $\mathcal{F}$ will inevitably increase the optimal source loss, which is the first term on the RHS of the expression in Theorem 1. Therefore, using general regularization, one has to trade-off the *optimal source loss*, *TFL-specific cross-client discrepancy*, and *TFL-specific source-target discrepancy*.

Based on Theorem 1 and the follow-up discussion, we aim to answer the question: how should one design a *practical regularizer* that can inherently tighten our bound on TFL? We give an answer to this question in Section 4.2. There, Theorem 2 shows that a transferability-boosting pretraining regularization should decrease the *cross-client Jacobian variance* while at the same time increase the *cross-client averaged Jacobian norm*.

https://doi.org/10.52202/079017-1233

4.2 Practical Bounds Based on Cross-Client Statistics

The goal of FL is to estimate the optimal solution f^* by updating the global model through multiple federated rounds [1]. Denote the total number of rounds by P and the output global model after round P by f_P. Denote the target loss under f_P instead of f^* as $\widehat{L}^*_{tgt}$ (which is an estimate of L^*_{tgt}). At the end of local training during any round $p \leq P$, let $h_p^{(k)}$ with weight $w_p^{(k)}$ represent the model of client k, and let h_p with weight w_p represent the global model. Denote the Jacobian (gradient) of the loss l w.r.t the model weights at domain k as $J^{(k)}(w) = \mathbb{E}_{\mathcal{D}_S^{(k)}} \nabla_{w_h} l(h(x), y)|_{w_h = w}$.

Throughout the remainder of the manuscript, we use $J_p^{(k)}$ as shorthand for $J^{(k)}(w_p)$. Furthermore, we denote the learning rate of agent k at round p by $\lambda_p^{(k)}$. We make a single-step local update assumption (Assumption 4.2) and use the definitions for cross-client statistics (Definition 4) to derive bounds exploiting the cross-client statistics from Lemma 4.1 and Theorem 2).

Assumption 4.2 (Single-Step Local Update)**.** *During local training, all clients perform one step of gradient descent (GD) to update their model for transmission,* $w_{p+1}^{(k)} = w_p - \lambda_{p+1}^{(k)} \cdot J^{(k)}(w_p)$*. This is a common assumption in the FL literature [30, 32].* [2]

Definition 4 (Cross-Client Statistics)**.** At federated round p, given K clients with local Jacobians $\{J_p^{(k)}\}_{k\in[K]}$, we define the *cross-client averaged Jacobian norm* $\|J_p\|_2$ and the *cross-client Jacobian variance*, respectively, as

$$\|J_p\|_2 = \left\| \frac{1}{K} \sum_k J_p^{(k)} \right\|_2, \text{ and } \sigma_p^2 = \frac{1}{K} \sum_k \left\| J_p^{(k)} \right\|_2^2 - \left\| \frac{1}{K} \sum_k J_p^{(k)} \right\|_2^2. \tag{8}$$

Note that we assumed the loss function to have a α-Lipschitz continuous gradient. When the gradient is large, α is also large, and to make $\beta_1(\lambda) \geq 0$, λ has to be close to 0. In this case, the absolute value of the second term can be small and that of the third term can be large.

Lemma 4.1 (Loss Bound Using Cross-Client Statistics)**.** *Under Assumptions 4.2 and 4.1, and the cross-client statistics defined in Definition 4, after P rounds of federated pretraining one has*

$$\widehat{L}^*_{tgt} \leq L_{src}(h_0) - \sum_{p=0}^{P-1} \beta_1(\lambda_{p+1}) \|J_p\|_2^2 + \sum_{p=0}^{P-1} \beta_2(\lambda_{p+1}) \sigma_p^2 + d_{\mathcal{G},\mathcal{F}}(\mathcal{D}_S^{fed}, \mathcal{D}_T), \tag{9}$$

where h_0 is the initial global model and $\beta_2(\lambda) = \frac{\alpha\lambda^2}{2}$, $\beta_1(\lambda) = \lambda - \beta_2(\lambda)$.

Note that *during the training process*, a large Jacobian norm can promote transferability (the Jacobian is expected to be small at the end of training).

Interpretation of Key Terms. Lemma 4.1 shows that the *target loss* of the finetuned pretrained model (LHS) is bounded by a sum (RHS) involving four key terms: 1) $L_{src}(h_0)$, the *initial source loss*; 2) $\|J_p\|_2$, the *cross-client averaged Jacobian norm*; 3) σ_p^2, the *cross-client Jacobian variance*; 4) $d_{\mathcal{G},\mathcal{F}}(\mathcal{D}_S^{fed}, \mathcal{D}_T)$, the *TFL-specific source-target domain divergence*. Since $L_{src}(h_0)$ is fixed throughout the pretraining process, and $d_{\mathcal{G},\mathcal{F}}(\mathcal{D}_S^{fed}, \mathcal{D}_T)$ can be reduced using general regularization, we focus on analyzing the two tunable cross-client statistics, $\|J_p\|_2$ and σ_p^2.

Influence of the Cross-Client Statistics on the Bound. We note that during pretraining, both the coefficients $\beta_1(\lambda_{p+1})$ and $\beta_2(\lambda_{p+1})$ have to be positive (see how to ensure these constraints in Appendix D). Therefore, a larger $\|J_p\|_2$ and a smaller σ_p^2 tighten the upper bound, indicate a lower target loss, and thus suggest better model transferability.

Coefficients Quadratic w.r.t. the Learning Rates. Lemma 4.1 involves coefficients $\beta_1(\lambda)$ and $\beta_2(\lambda)$ that are quadratic in the learning rate λ; thus, we can further tighten the bound by optimizing the learning rates across different rounds (Assumption 4.3). The tightened bound is given in Theorem 2.

[1] One federated round comprises local model initialization, local training, local model transmission, global model aggregation, and global model broadcasting

[2] We briefly discuss the Stochastic Gradient Descent approach in Appendix F.

Assumption 4.3 (Optimal Learning Rates Across Rounds)**.** *In each round p ($2 \leq p \leq P$), we use an optimal learning rate for local training $\lambda_p^* = \frac{K \cdot \|J_{p-1}\|_2^2}{\alpha \cdot \sum_k \|J_{p-1}^{(k)}\|_2^2}$. A similar analysis and assumptions on optimal learning rate has also been used in [7, 10, 13, 30, 49].*

Theorem 2 (Tightened Bound Based on Cross-Client Statistics)**.** *By optimizing the bound in Lemma 4.1 with respect to the learning rates λ as governed by Assumption 4.3, and under Assumption 4.1 (Convexity and Smoothness), the estimated **optimal** target loss is bounded as follows:*

$$\widehat{L}_{tgt}^* \leq L_{src}(h_0) - \sum_{p=0}^{P-1} \frac{2\|J_p\|_2^2}{\alpha(1 + \sigma_p^2 \|J_p\|_2^{-2})} + d_{\mathcal{G},\mathcal{F}}(\mathcal{D}_S^{fed}, \mathcal{D}_T).$$

Semantic Interpretation. Theorem 2 indicates that by optimizing the learning rates at each round, the bound for $\widehat{L}_{tgt}^*$ can be made smaller through 1) a smaller source loss $L_{src}(h_0)$; 2) a larger cross-client average Jacobian norm $\|J_p\|_2$; 3) a smaller cross-client Jacobian variance σ_p^2, and 4) a smaller source-target domain divergence $d_{\mathcal{G},\mathcal{F}}(\mathcal{D}_S^{fed}, \mathcal{D}_T)$. These are consistent with Lemma 4.1.

Intuitive Explanation. **(a)** *Increasing the Cross-Client Averaged Jacobian Norm $\|J_p\|$.* In early training, it is crucial to avoid a small J_p, as it can "trap" clients in local minima or cause overfitting of local distributions. Note that increasing J_p will not induce an excessively large second term (i.e., a term that $\to \infty$) in Theorem 2, as increments are constrained through the Lipschitz condition $\|J_1 - J_2\| \leq \alpha \|x_1 - x_2\|$. In the final stages of training, $\|J_p\|$ naturally decreases due to convergence. **(b)** *Decreasing the Cross-Client Jacobian Variance σ_p.* A small σ_p promotes domain similarity, preventing local overfitting and enhancing global model transferability. **(c)** *Trade-Off Between Increasing $\|J_p\|$ and Decreasing σ_p.* A larger $\|J_p\|$ induces more substantial local updates but can increase variance σ_{p+1}. Thus, Theorem 2 underscores balancing this trade-off: enlarging $\|J_p\|$ while maintaining a small σ_p.

Challenges in Regularization of the Jacobians. Certain regularization of the Jacobians can induce local Jacobian alignment and reduce σ_p^2 (e.g., the regularization of the alignment between local gradients and the global gradient proposed in FedIIR [12]). However, local regularization can naturally *impede* the growth of local Jacobian norms $\{J_p^{(k)}\}_{k \in [K]}$ and subsequently prevent $\|J_p\|_2$ from increasing. Thus, such prior regularization may not necessarily improve transferability. As an example, consider the extreme case where $\sigma_p^2 = 0$. Then, the bound in Theorem 2 takes the form $\widehat{L}_{tgt}^* \leq L_{src}(h_0) - \sum_{p=0}^{P-1} \frac{2}{\alpha} \|J_p\|_2^2 + d_{\mathcal{G},\mathcal{F}}(\mathcal{D}_S^{fed}, \mathcal{D}_T)$, where a small $\|J_p\|_2$ can severely degrade the bound. This indicates that boosting transferability by prior regularization, such as controlling σ_p^2 itself, has limitations, and tuning $\|J_p\|$ during the pretraining stage is of crucial importance. We provide a solution to this problem in Section 5.

5 Our Algorithm

Algorithmic Solution for the Theoretical Challenges. From Lemma 4.1 and Theorem 2, we can see that certain round-wise FL-specific statistics, the *cross-client averaged Jacobian norm* $\|J_p\|_2$ and *cross-client Jacobian variance* σ_p^2 control the bound on the target loss. The challenge is that, while σ_p^2 can be reduced using straightforward techniques (i.e., such as gradient alignment from FedIIR [12]), such techniques unavoidably prevent J_p from increasing properly. We therefore propose our FedGTST approach which reduces σ_p^2 while enlarging $\|J_p\|_2$. A round-wise description of FedGTST is given in Algorithm 5.

Tuning the Cross-Client Jacobian Variance σ_p^2. The cross-client Jacobian variance is controlled via regularization at the local client level (Line 5 of Algorithm 5). The local clients, upon receiving a *guide norm* γ from the server (explanation deferred to subsequent sections), implement a regularized local training protocol. While client k during standard local training uses the objective $L_{\mathcal{D}^{(k)}}(h)$, during regularized local training he/she/they use

$$L_{\mathcal{D}_S^{(k)}}(h) + \xi \cdot \left(\|J^{(k)}(h)\|_2 - \gamma \right)^2 \tag{10}$$

instead, where ξ is the penalty coefficient for the regularization term. This type of regularization intuitively aligns each local Jacobian norm $\|J^{(k)}\|$ with the guide norm γ, which results in a σ_J^2 that is smaller than that of standard FL (i.e., FedAVG).

https://doi.org/10.52202/079017-1233

Algorithm 1 FedGTST (Round p)

1: Randomly select a set $\phi \subset [K]$ for regular training.
2: **while** $k \leq K$, Client k should **do**
3: Receive the *guide* norm γ_{p-1} and global model h_{p-1} from the server.
4: Initialize the local model to h_{p-1}.
5: Update the local model to $h_p^{(k)}$ by training with $L_{\mathcal{D}_S^{(k)}}(h) + \xi \cdot \left(\|J^{(k)}(h)\|_2 - \gamma\right)^2$.
6: Calculate the surrogate Jacobian norm $\gamma_p^{(k)} = \gamma_p^{reg,(k)} = \|J_p^{(k)}\|_2$.
7: **if** $k \in \phi$ **then**
8: Train with $L_{\mathcal{D}^{(k)}}(h)$ to obtain Jacobian norm $\gamma_p^{std,(k)}$.
9: Update the surrogate Jacobian norm $\gamma_p^{(k)} = \max\left(\gamma_p^{reg,(k)}, \gamma_p^{std,(k)}\right)$.
10: **end if**
11: Transmit the model $h_p^{(k)}$ and norm $\gamma_p^{(k)}$ to the server.
12: **end while**
13: The server aggregates the client models into a global model h_p and sets the guide norm to $\gamma_p = \max_k(\gamma_p^{(k)})$.
14: The server broadcasts h_p and γ_p to all clients.

Tuning the Cross-Client Averaged Jacobian Norm $\|J_p\|_2$. The cross-client averaged Jacobian norm can be increased via an exchange protocol that includes: 1) clients calculating surrogate norms for transmission; 2) the server computing and broadcasting a guide norm; 3) clients performing local alignment using the guide norm.

Surrogate Norms. A decrease in the local Jacobian norm $\|J^{(k)}\|$ prevents the cross-client averaged Jacobian norm $\|J\|$ from growing. We mitigate such a decrease by forcing a small portion of the clients to implement both regularized training and standard training (Line 1), to generate a pair of Jacobian norms $\gamma^{reg,(k)}$ and $\gamma^{std,(k)}$. The client than chooses the larger norm, $\gamma^{(k)} = max(\gamma^{reg,(k)}, \gamma^{std,(k)})$ as a surrogate norm for transmission (Lines 6 to 10).

Server Guide Norm. The clients send their local Jacobian norms $\gamma^{(k)} := \|J^{(k)}\|_2$ to the server, and the server broadcasts the largest norm received as its guide norm $\gamma := \max_k \gamma^{(k)}$ (Line 13).

Local alignment. The local regularizer from Equation (10) reduces the variance, but can also force an increase of the averaged Jacobian norm. It forces all local Jacobian norms to align with the guide norm γ (Line 5). Since γ has been boosted both at the clients' and server levels, the alignment leads to larger local Jacobian norms.

Communication Cost. Most TFL methods, as already described, require large communication overheads between the clients and the server. For example, FedIIR requires exchanging Jacobians, which have the same dimension as the model weights; FedGTST only requires exchanging norms.

6 Experiments

6.1 Experimental Setting

Transfer tasks. We investigate three transfer tasks utilizing fully-annotated data: a) MNIST [9] to MNIST-M [11], b) CIFAR-10 [19] to SVHN [33], and c) cross-domain transfer in DomainNet [37]. All transfer tasks have been used as benchmarks in existing TL research [3, 8, 11, 22, 28, 50] (also, see Appendix G). Moreover, DomainNet is a standard large-scale dataset for TFL studies, also used by FedSR.

Non-iid Distributed Source (Local) Domains. The pretraining phase is conducted via FL on source (local) domains. Marginal distribution shift is an important phenomenon in FL [27] since the access to classes (or categories) is not the same for all participating entities. However, some federated datasets tested by existing transferable FL methods do not reflect marginal distribution shifts when constructing source local domains (i.e., in the Rotated-MNIST benchmark in FedSR[34], all clients have access to all classes). To address this issue, we employ the following methods for constructing non-iid local domains:

- For MNIST or CIFAR-10, unless otherwise specified, we follow the approach in [27] and let each client have access to only a subset of the categories. The category selection rule and data sampling method is described in Appendix H. Besides, we run additional experiments for Dirichlet sampling, a method commonly used in FL [1, 20, 23, 24, 29, 45, 46, 54].
- For DomainNet, which compromises six distinct domains, we follow the leave-one-out strategy used in FedSR: one domain is designated as the target domain, while the remaining five domains serve as source domains, with each assigned to an individual client.

Federated System Size. Large system sizes are inherent to FL systems, where the number of clients can be as large as 100 [25, 27]. In such a case, it is more challenging for the global model to achieve good performance. However, existing TFL methods typically use a very small number of domains (≤ 5 for FedSR and StableFDG). In contrast, we allow the system size to cover a broad range of values, including 10 (small), 50 (medium), and 100 (large) clients.

Evaluation matrix and backbones. We measure domain transferability via acc_{tgt}, the accuracy of a pretrained model finetuned on the target domain. For backbone selection, we use both LeNet [21] and ResNet18[14] to represent different levels of backbone complexity.

Baselines. We consider the following TFL algorithms as our main baselines: FedAVG, FedSR, and FedIIR. Furthermore, we also compare our findings to Scaffold [16], an advanced FL approach used to address data heterogeneity. We do not report results for FedADG, FedCDG, FedGTST, and StableFDG since they do not ensure privacy and/or underperform compared to the main baseline.

Settings for Pretraining (FL on source local domains). 1) *Local epochs*. To conform with our theoretical assumptions, unless specified otherwise, we set the local epoch of each client to 1. We also investigate the system performance with 10 local epochs in Appendix C. 2) *Number of participants per round*. We allow 50% of the clients to participate in each round (e.g., if $K = 50$, there are 25 participants per round). 3) *Number of clients performing standard local training*. To boost the Jacobian norm, a subset of clients conducts standard local training (we set the subset size to 10% of the total number of clients).

Additional Settings. Unless specified otherwise, for local training on the source datasets we use the Adam [17] optimizer with coefficients $(\beta_1, \beta_2) = (0.9, 0.999)$ (note that these β values are not to be confused with the coefficients from our theoretical analysis). Our initial learning rate equals 0.01 and then decays by a factor of 10 per 50 rounds, with an early stop trigger of 10 rounds. We apply the standard cross-entropy loss. We also set the pretraining batch size to 256 for MNIST $\rightarrow$ MNIST-M and to 128 for CIFAR10 $\rightarrow$ SVHN. For both tasks, we use a finetuned learning rate 0.005, with weight decay 0.0001. All results reported in Section 6.2 are averaged over three runs.

6.2 Results

Table 1: Target accuracy (%) of the finetuned model pretrained on a small number of clients ($K = 10$). FedGTST outperforms other methods across both tasks and both backbones; for the example MNIST to MNIST-M with a LeNet backbone, FedGTST outperforms FedIIR and FedSR by around 4%.

Method	MNIST$\rightarrow$ MNIST-M		CIFAR10 $\rightarrow$ SVHN		Average
	LeNet	ResNet	LeNet	ResNet	
FedAVG	73.8±0.7	81.6±0.2	64.4±0.5	72.0±1.0	73.0
FedSR	75.0±0.9	80.6±0.1	65.9±0.6	71.3±0.4	73.2
FedIIR	74.5±0.3	**82.7±0.7**	66.2±1.0	73.8±0.2	74.3
Fed-GTST	**76.2±0.9**	82.3±0.5	**70.1±0.8**	**74.5±0.3**	**75.8**

Transferability results. FedGTST exhibits **significantly improved transfer performance** when compared to baselines across a range of tasks, system sizes, and backbone architectures. For the example of CIFAR10$\rightarrow$SVHN with 100 clients and a LeNet backbone, FedGTST outperforms FedIIR by 7.6% and FedSR by 9.8%. The results for a small, medium and large federated system ($K = 10$, $K = 50$ and $K = 100$) are reported in Tables 1, 2 and 3, respectively.

Discussion. Besides the significant performance gain of FedGTST over baseline methods, we also observe that 1) for transfer tasks in which the backbone and method are fixed, transferability generally decreases with the system size. Importantly, FedGTST improves the baselines more significantly

https://doi.org/10.52202/079017-1233

Table 2: Target accuracy (%) of the finetuned model pretrained on a medium number of clients ($K = 50$). FedGSTS outperforms other methods across both tasks and both backbones; for both the examples of MNIST→MNIST-M and CIFAR10→SVHN with a ResNet18 backbone, FedGTST outperforms FedIIR and FedSR by around 5%.

Method	MNIST→ MNIST-M		CIFAR10 → SVHN		Average
	LeNet	ResNet	LeNet	ResNet	
FedAVG	63.5 ± 0.3	72.4±0.2	59.1±0.5	65.5±0.1	65.1
FedSR	64.0±0.6	73.1±0.3	58.9±0.1	66.8±0.5	65.7
FedIIR	64.7±0.2	73.9±0.4	59.7±0.3	66.2±0.1	66.1
Fed-GTST	**69.0±0.8**	**79.2±0.4**	**63.5±0.1**	**71.1±0.2**	**70.7**

Table 3: Target accuracy (%) of the finetuned model pretrained on a large number of clients ($K = 100$). FedGTST outperforms other methods across both tasks and for both backbones; for CIFAR10→SVHN with a ResNet18 and LeNet backbone, FedGTST outperforms FedIIR and FedSR by 7%.

Method	MNIST→ MNIST-M		CIFAR10 → SVHN		Average
	LeNet	ResNet	LeNet	ResNet	
FedAVG	48.7±0.1	61.1±0.3	41.2±0.2	51.7±0.5	50.7
FedSR	49.2±0.2	59.8±0.3	42.6±0.1	52.0±0.3	50.9
FedIIR	51.9±0.4	61.3±0.1	44.8±0.4	55.8±0.2	53.4
Fed-GTST	**57.5**±0.3	**67.6**±0.2	**52.4**±0.1	**63.1**±0.2	**60.2**

for large system sizes; 2) when the system size, transfer task and the method are fixed, the more "complex" the backbone (ResNet18 vs LeNet), the better the transferability.

Additional results. We defer reporting and discussing additional results in Appendix C, which include (a) an investigation on *hyper-parameter sensitivity* the *convergence speed*, and *cross-client statistics*, (b) results regarding *DomainNet*, *Dirichlet Sampling*, and *Scaffold*.

7 Limitations

Increased Local Computational Cost. While FedGTST only induces a negligible communication overhead among clients and the server, we note that at the level of a *small subset of clients* we need to conduct both regularized training and standard training to boost $\|J_p\|_2$. This increases the local computational burden but to a very small extent. Nevertheless, in future works, we will explore algorithms with lower local computational costs.

Potentially Loose Bounds. Although existing studies have added to our understanding of transferable federated learning, our work is the first to derive a bound on a direct measure of transferability (the target loss). We believe that the bound can be tightened.

8 Conclusion

We introduced FedGTST, a federated learning algorithm aimed at enhancing global transferability. Inspired by theoretical insights, FedGTST integrates cross-client information on averaged Jacobian norms and Jacobian variance. Our work addresses key challenges in existing methods, such as privacy violations and an overemphasis on local transferability. Experimental results demonstrate significant performance improvements over baseline models.

Acknowledgement

This work was supported by the Air Force Office of Scientific Research under award number FA9550-23-1-0107, the NSF CAREER Award under Grant No. EPCN-1944403 as well as the NSF Awards CCF 1956384 and 2402815, and SVCF CZI 2022-249120.

References

[1] Durmus Alp Emre Acar, Yue Zhao, Ramon Matas Navarro, Matthew Mattina, Paul N Whatmough, and Venkatesh Saligrama. Federated learning based on dynamic regularization. In *International Conference on Learning Representations (ICLR)*, 2021. URL https://openreview.net/forum?id=B7v4QMR6Z9w.

[2] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, David Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konečnỳ, Stefano Mazzocchi, H Brendan McMahan, et al. Towards federated learning at scale: System design. *arXiv preprint arXiv:1902.01046*, 2019.

[3] Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain separation networks. *Advances in neural information processing systems*, 29, 2016.

[4] Junming Chen, Meirui Jiang, Qi Dou, and Qifeng Chen. Federated domain generalization for image recognition via cross-client style transfer. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 361–370, 2023.

[5] Corinna Cortes and Mehryar Mohri. Domain adaptation and sample bias correction theory and algorithm for regression. *Theoretical Computer Science*, 519:103–126, 2014. ISSN 0304-3975. doi: https://doi.org/10.1016/j.tcs.2013.09.027. URL https://www.sciencedirect.com/science/article/pii/S0304397513007184. Algorithmic Learning Theory.

[6] Corinna Cortes, Mehryar Mohri, and Andrés Munoz Medina. Adaptation based on generalized discrepancy. *Journal of Machine Learning Research*, 20(1):1–30, 2019.

[7] Laizhong Cui, Xiaoxin Su, Yipeng Zhou, and Jiangchuan Liu. Optimal rate adaption in federated learning with compressed communications. In *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*, pages 1459–1468. IEEE, 2022.

[8] Bharath Bhushan Damodaran, Benjamin Kellenberger, Rémi Flamary, Devis Tuia, and Nicolas Courty. Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 447–463, 2018.

[9] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[10] Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. Adaptive personalized federated learning. *arXiv preprint arXiv:2003.13461*, 2020.

[11] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of machine learning research*, 17(59):1–35, 2016.

[12] Yaming Guo, Kai Guo, Xiaofeng Cao, Tieru Wu, and Yi Chang. Out-of-distribution generalization of federated learning via implicit invariant relationships. In *International Conference on Machine Learning*, pages 11905–11933. PMLR, 2023.

[13] Farzin Haddadpour and Mehrdad Mahdavi. On the convergence of local descent methods in federated learning. *arXiv preprint arXiv:1910.14425*, 2019.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[15] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjuneta Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

[16] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5132–5143. PMLR, 2020.

 https://doi.org/10.52202/079017-1233

[17] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[18] Jakub Konečný, H Brendan McMahan, Daniel Ramage, Peter Richtárik, Ananda Theertha Suresh, et al. Federated optimization: Distributed optimization beyond the datacenter. *arXiv preprint arXiv:1511.03575*, 2016.

[19] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[20] Fan Lai, Yinwei Dai, Sanjay Singapuram, Jiachen Liu, Xiangfeng Zhu, Harsha Madhyastha, and Mosharaf Chowdhury. Fedscale: Benchmarking model and system performance of federated learning at scale. In *International Conference on Machine Learning (ICML)*, 2022. URL `https://proceedings.mlr.press/v162/lai22a.html`.

[21] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

[22] Seunghun Lee, Sunghyun Cho, and Sunghoon Im. Dranet: Disentangling representation and adaptation networks for unsupervised cross-domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15252–15261, 2021.

[23] Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10713–10722, 2021. doi: 10.1109/CVPR46437.2021.01057. URL `https://openaccess.thecvf.com/content/CVPR2021/papers/Li_Model-Contrastive_Federated_Learning_CVPR_2021_paper.pdf`.

[24] Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *CVPR*, pages 10713–10722, 2021. doi: 10.1109/CVPR46437.2021.01057. URL `https://openaccess.thecvf.com/content/CVPR2021/papers/Li_Model-Contrastive_Federated_Learning_CVPR_2021_paper.pdf`.

[25] Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10713–10722, June 2021.

[26] Tianqing Li, Anup Sahu, Ameet S Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020.

[27] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. *arXiv preprint arXiv:1907.02189*, 2019.

[28] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pages 97–105. PMLR, 2015.

[29] Mi Luo, Fei Chen, Dapeng Hu, Yifan Zhang, Jian Liang, and Jiashi Feng. No fear of heterogeneity: Classifier calibration for federated learning with non-iid data. In *Advances in Neural Information Processing Systems*, volume 34, pages 6072–6084, 2021.

[30] Evelyn Ma, Praneet Rathi, and S Rasoul Etesami. Local environment poisoning attacks on federated reinforcement learning. *arXiv preprint arXiv:2303.02725*, 2023.

[31] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation: Learning bounds and algorithms. *arXiv preprint arXiv:0902.3430*, 2009.

[32] 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.

[33] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[34] A Tuan Nguyen, Philip Torr, and Ser Nam Lim. Fedsr: A simple and effective domain generalization method for federated learning. *Advances in Neural Information Processing Systems*, 35:38831–38843, 2022.

[35] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[36] Jungwuk Park, Dong-Jun Han, Jinho Kim, Shiqiang Wang, Christopher Brinton, and Jaekyun Moon. Stablefdg: Style and attention based learning for federated domain generalization. *Advances in Neural Information Processing Systems*, 36, 2024.

[37] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1406–1415, 2019.

[38] Mohammad Pezeshki, Chengxu Le Lan, Thibaut Durand, Guillaume Lajoie, Simon Lacoste-Julien, Aaron Courville, and David Lopez-Paz. Understanding hessian alignment for domain generalization. *arXiv preprint arXiv:2308.11778*, 2023.

[39] Alexandre Rame, Corentin Dancette, and Matthieu Cord. Fishr: Invariant gradient variances for out-of-distribution generalization. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 18347–18377. PMLR, 17–23 Jul 2022. URL `https://proceedings.mlr.press/v162/rame22a.html`.

[40] Sebastian Ruder, Ivan Vulic, and Anders Søgaard. A survey of cross-lingual word embedding models. *Journal of Artificial Intelligence Research*, 65:569–631, 2019.

[41] Hadi Salman, Andrew Ilyas, Logan Engstrom, Ashish Kapoor, and Aleksander Madry. Do adversarially robust imagenet models transfer better? *Advances in Neural Information Processing Systems*, 33, 2020. URL `https://par.nsf.gov/biblio/10283431`.

[42] Yan Shen, Jian Du, Han Zhao, Zhanghexuan Ji, Chunwei Ma, and Mingchen Gao. Fedmm: A communication efficient solver for federated adversarial domain adaptation. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, pages 1808–1816, 2023.

[43] Yuge Shi, Jeffrey Seely, Philip H. S. Torr, Siddharth Narayanaswamy, Awni Y. Hannun, Nicolas Usunier, and Gabriel Synnaeve. Gradient matching for domain generalization. In *International Conference on Learning Representations (ICLR)*, 2022. URL `https://openreview.net/forum?id=_7w5JXy8VJM`.

[44] Virginia Smith. Federated learning: strategies for improving communication efficiency. *IEEE Signal Processing Magazine*, 34(6):36–43, 2017.

[45] Zhenheng Tang, Yonggang Zhang, Shaohuai Shi, Xin He, Bo Han, and Xiaowen Chu. Virtual homogeneity learning: Defending against data heterogeneity in federated learning. *arXiv preprint arXiv:2206.02465*, 2022.

[46] Zhenheng Tang, Yonggang Zhang, Shaohuai Shi, Xinmei Tian, Tongliang Liu, Bo Han, and Xiaowen Chu. Fedimpro: Measuring and improving client update in federated learning. In *The Twelfth International Conference on Learning Representations (ICLR)*, Vienna, Austria, May 7-11 2024. OpenReview.net.

[47] Lisa Torrey and Jude Shavlik. Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, pages 242–264. IGI global, 2010.

[48] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big data*, 3(1):1–40, 2016.

 https://doi.org/10.52202/079017-1233

[49] Xidong Wu, Feihu Huang, Zhengmian Hu, and Heng Huang. Faster adaptive federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 10379–10387, 2023.

[50] Xiaojun Xu, Jacky Y Zhang, Evelyn Ma, Hyun Ho Son, Sanmi Koyejo, and Bo Li. Adversarially robust models may not transfer better: Sufficient conditions for domain transferability from the view of regularization. In *International Conference on Machine Learning*, pages 24770–24802. PMLR, 2022.

[51] Qiang Yang, Yang Liu, Tianjian Chen, Yu Tong, et al. Federated optimization: Distributed optimization beyond the datacenter. *IEEE Signal Processing Magazine*, 36(4):49–59, 2019.

[52] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *Advances in Neural Information Processing Systems*, 27, 2014.

[53] Xinhui Yu, Dan Wang, Martin McKeown, and Z Jane Wang. Contrastive-enhanced domain generalization with federated learning. *IEEE Transactions on Artificial Intelligence*, 2023.

[54] Jie Zhang, Chen Chen, Bo Li, Lingjuan Lyu, Shuang Wu, Shouhong Ding, Chunhua Shen, and Chao Wu. DENSE: Data-free one-shot federated learning. In *Advances in Neural Information Processing Systems*, volume 35, pages 19112–19126, 2022. URL `https://papers.nips.cc/paper_files/paper/2022/hash/868f2266086530b2c71006ea1908b14a-Abstract-Conference.html`.

[55] Liling Zhang, Xinyu Lei, Yichun Shi, Hongyu Huang, and Chao Chen. Federated learning with domain generalization. *arXiv preprint arXiv:2111.10487*, 2021.

[56] Han Zhao, Shanghang Zhang, Guanhang Wu, José MF Moura, Joao P Costeira, and Geoffrey J Gordon. Adversarial multiple source domain adaptation. In *Advances in Neural Information Processing Systems*, pages 8568–8579, 2018.

[57] Han Zhao, Remi Tachet Des Combes, Kun Zhang, and Geoffrey Gordon. On learning invariant representations for domain adaptation. In *International conference on machine learning*, pages 7523–7532. PMLR, 2019.

[58] Han Zhao, Chen Dan, Bryon Aragam, Tommi S Jaakkola, Geoffrey J Gordon, and Pradeep Ravikumar. Fundamental limits and tradeoffs in invariant representation learning. *The Journal of Machine Learning Research*, 23(1):15356–15404, 2022.

A Additional preliminaries

Definition 5 ($\mathcal{H}$-discrepancy [6, 57])**.** Given a model function class $\mathcal{H}$ and two data distributions $\mathcal{D}_S$, $\mathcal{D}_T$, the $\mathcal{H}$-discrepancy between $\mathcal{D}_S, \mathcal{D}_T$ is defined as

$$d_{\mathcal{H}}(\mathcal{D}_S, \mathcal{D}_T) := \sup_{h \in \mathcal{H}} |L_{\mathcal{D}_S}(h) - L_{\mathcal{D}_T}(h)|. \tag{11}$$

Remark. This type of discrepancy, as well as the related $\mathcal{H}$-divergence, have been frequently used in the analysis of domain adaptation [5, 31, 57], and we follow this trend.

B Proof of our bounds

Lemma B.1 (Theorem 2.5 in [50])**.** *Suppose we perform pretraining on the source domain $\mathcal{D}_S$ to obtain f and g at the server, and fine-tune the model on the target domain $\mathcal{D}_T$ to obtain a new classifier g_T. We then have*

$$L_{\mathcal{D}_T}(g_T^* \circ f^*) \leq L_{src}(g^* \circ f^*) + d_{\mathcal{G},\mathcal{F}}(\mathcal{D}_S, \mathcal{D}_T). \tag{12}$$

Equation (12) shows that the transfer loss can be upper bounded by the sum of the loss on the source domain and the divergence between two domains.

Lemma B.2. *With Assumption 4.1 (Convexity and Smoothness) and Definition 2 (Cross-Client Divergence), we have*

$$L_{src}(h^*) \leq \frac{1}{K} \sum_{k=1}^{K} \left[L_{\mathcal{D}^{(k)}} \left(h^{*(k)} \right) \right] + \overline{d}_{\mathcal{H}}(\mathcal{D}_S^{fed}). \tag{13}$$

Proof. The loss function l is assumed to be convex w.r.t. the parameters of the model h, where the parameters are denoted by w_h. Based on the aggregation rule in Section 3, we have $w_h = \frac{1}{K} \sum_k w_h^{(k)}$, where w_h and $w_h^{(k)}$ stands for the weights for global model h and that for local model $h^{(k)}$, respectively. We therefore have

$$\begin{aligned}
L_{src}(h^*) &= \frac{1}{K} \sum_{k_1} L_{\mathcal{D}^{(k_1)}} \left(h^* \right) \\
&= \frac{1}{K} \sum_{k_1} \mathbb{E}_{\mathcal{D}^{(k_1)}} \, l \left(h \left(x, w_h^* \right), y \right) \\
&= \frac{1}{K} \sum_{k_1} \mathbb{E}_{\mathcal{D}^{(k_1)}} \, l \left(h \left(x, \frac{1}{K} \sum_{k_2} w_h^{*(k_2)} \right), y \right) \\
&\overset{(a)}{\leq} \frac{1}{K^2} \sum_{k_1, k_2} \mathbb{E}_{\mathcal{D}^{(k_1)}} \, l \left(h \left(x, w_h^{*(k_2)} \right), y \right) \\
&= \frac{1}{K^2} \sum_{k_1, k_2} L_{\mathcal{D}^{(k_1)}} \left(h^{*(k_2)} \right) \\
&\overset{(b)}{\leq} \frac{1}{K^2} \sum_{k_1, k_2} \left[L_{\mathcal{D}^{(k_1)}} \left(h^{*(k_1)} \right) + d_{\mathcal{H}} \left(\mathcal{D}^{(k_1)}, \mathcal{D}^{(k_2)} \right) \right] \\
&= \frac{1}{K} \sum_{k_1} L_{\mathcal{D}^{(k_1)}} \left(h^{*(k_1)} \right) + \frac{1}{K^2} \sum_{k_1, k_2} d_{\mathcal{H}} \left(\mathcal{D}^{(k_1)}, \mathcal{D}^{(k_2)} \right) \\
&\leq \frac{1}{K} \sum_{k_1} L_{\mathcal{D}^{(k_1)}} \left(h^{*(k_1)} \right) + \overline{d}_{\mathcal{H}}(\mathcal{D}_S),
\end{aligned}$$

where $\frac{1}{K^2} \sum_{k_1, k_2} d_{\mathcal{H}} \left(\mathcal{D}^{(k_1)}, \mathcal{D}^{(k_2)} \right) \leq \frac{1}{K(K-1)} \sum_{k_1 \neq k_2} d_{\mathcal{H}} \left(\mathcal{D}^{(k_1)}, \mathcal{D}^{(k_2)} \right) = \overline{d}_{\mathcal{H}}(\mathcal{D}_S)$ as in Definition 2. Here (a) follows from the convexity assumption for l w.r.t the parameters, while (b) follows

https://doi.org/10.52202/079017-1233

from Lemma B.1 based on the argument below:

$$\begin{aligned} L_{\mathcal{D}^{(k_1)}}\left(h^{*(k_2)}\right) &= L_{\mathcal{D}^{(k_1)}}\left(h^{*(k_2)}\right) - L_{\mathcal{D}^{(k_2)}}\left(h^{*(k_2)}\right) + L_{\mathcal{D}^{(k_2)}}\left(h^{*(k_2)}\right) \\ &\leq \left|L_{\mathcal{D}^{(k_1)}}\left(h^{*(k_2)}\right) - L_{\mathcal{D}^{(k_2)}}\left(h^{*(k_2)}\right)\right| + L_{\mathcal{D}^{(k_2)}}\left(h^{*(k_2)}\right) \\ &\leq d_{\mathcal{H}}\left(\mathcal{D}^{(k_1)}, \mathcal{D}^{(k_2)}\right) + L_{\mathcal{D}^{(k_2)}}\left(h^{*(k_2)}\right). \end{aligned}$$

□

Theorem 3 (Theorem 1)**.** *Under Assumptions 4.1 (Convexity and Smoothness), the optimal target loss is bounded by*

$$L_{tgt}^{*} \leq \frac{1}{K}\sum_{k=1}^{K}\left[L_{\mathcal{D}^{(k)}}\left(h^{*(k)}\right)\right] + \overline{d}_{\mathcal{H}}(\mathcal{D}_S^{fed}) + d_{\mathcal{G},\mathcal{F}}(\mathcal{D}_S^{fed}, \mathcal{D}_T), \tag{14}$$

where $h^{*(k)}$ denotes the optimal local model of client k (see Section 3).

Proof. Since

$$L_{tgt}^{*} = L_{\mathcal{D}_T}\left(g_T^{*} \circ f^{*}\right) \leq L_{\mathcal{D}^{(k)}}\left(g^{*} \circ f^{*}\right) + d_{\mathcal{G},\mathcal{F}}(\mathcal{D}^{(k)}, \mathcal{D}_T),$$

we have

$$\begin{aligned} L_{tgt}^{*} &\leq \frac{1}{K}\sum_{k}\left[L_{\mathcal{D}^{(k)}}\left(g^{*} \circ f^{*}\right) + d_{\mathcal{G},\mathcal{F}}(\mathcal{D}^{(k)}, \mathcal{D}_T)\right] \\ &= L_{src}(h^{*}) + d_{\mathcal{G},\mathcal{F}}(\mathcal{D}_S^{fed}, \mathcal{D}_T). \end{aligned}$$

Using Lemma B.2, we have

$$\begin{aligned} L_{tgt}^{*} &\leq L_{src}(h^{*}) + d_{\mathcal{G},\mathcal{F}}(\mathcal{D}_S^{fed}, \mathcal{D}_T) \\ &\leq \frac{1}{K}\sum_{k=1}^{K}\left[L_{\mathcal{D}^{(k)}}\left(h^{*(k)}\right)\right] + \overline{d}_{\mathcal{H}}(\mathcal{D}_S^{fed}) + d_{\mathcal{G},\mathcal{F}}(\mathcal{D}_S^{fed}, \mathcal{D}_T). \end{aligned}$$

□

Lemma B.3 (Bound on round-wise source loss)**.** *Suppose the learning rates of all clients at round p are the same, i.e., $\lambda_p^{(k)} = \lambda_p, \forall k \in [K], p \in [P]$. Under Assumptions 4.2 and 4.1, we have that*

$$L_{src}\left(h_{p+1}\right) \leq L_{src}\left(h_p\right) - \beta_1(\lambda_{p+1})\|J_p\|_2^2 + \beta_2(\lambda_{p+1})\sigma_p^2 \tag{15}$$

where $J_p = \frac{1}{K}\sum_k J_p^{(k)}$, $\sigma_p^2 = \frac{1}{K}\sum_k \left\|J_p^{(k)}\right\|_2^2 - \left\|\frac{1}{K}\sum_k J_p^{(k)}\right\|_2^2$, and $\beta_1(\lambda) = \lambda - \beta_2(\lambda)$, $\beta_2(\lambda) = \frac{\alpha\lambda^2}{2}$.

Proof. Following the same proof idea as for Lemma B.2, we have

$$L_{src}\left(h_{p+1}\right) \leq \frac{1}{K^2}\sum_{k_1,k_2} L_{\mathcal{D}^{(k_1)}}\left(h_{p+1}^{(k_2)}\right) \tag{16}$$

By definition, we also have

$$L_{\mathcal{D}^{(k_1)}}\left(h_{p+1}^{(k_2)}\right) = \mathbb{E}_{\mathcal{D}^{(k_1)}} l\left(h\left(x, w_{p+1}^{(k_2)}\right), y\right) \tag{17}$$

From the update rule of GD, we can write

$$\begin{aligned} l\left(h\left(x, w_{p+1}^{(k_2)}\right), y\right) &= l\left(h\left(x, w_p - \lambda_{p+1} \cdot \left[\nabla_{w_h} L_{\mathcal{D}^{(k_2)}}\left(h\right)\Big|_{w_h = w_p}\right]\right), y\right) \\ &= l\left(h\left(x, w_p - \lambda_{p+1} \cdot \left[\nabla_{w} \mathbb{E}_{\mathcal{D}^{(k_2)}} l\left(h(x_i, w), y_i\right)\Big|_{w = w_p}\right]\right), y\right) \qquad (18) \\ &= l\left(h\left(x, w_p - \lambda_{p+1} \cdot J_p^{(k_2)}\right), y\right) \qquad (19) \end{aligned}$$

In (18), (x_i, y_i) are sampled from $\mathcal{D}^{(k_2)}$, and not to be confused with (x, y) sampled from $\mathcal{D}^{(k_1)}$.

Define $\Delta := -\lambda_{p+1} \cdot J_p^{(k_2)}$, we can upper bound the loss in (19) by

$$\begin{aligned} l\left(h\left(x, w_{p+1}^{(k_2)}\right), y\right) &= l\left(h\left(x, w_p + \Delta\right), y\right) \\ &\leq l\left(h\left(x, w_p\right), y\right) + \left[\nabla_w l\left(h\left(x, w\right), y\right)\big|_{w=w_p}\right]^\top \cdot \Delta + \frac{\alpha}{2}\Delta^\top\Delta. \end{aligned} \tag{20}$$

The inequality holds since l is assumed to have α-Lipschitz continuous gradient. Combining (17) and (20), we have

$$\begin{aligned} L_{\mathcal{D}^{(k_1)}}\left(h_{p+1}^{(k_2)}\right) &\leq \mathbb{E}_{\mathcal{D}^{(k_1)}}\left[l\left(h\left(x, w_p\right), y\right) + \left[\nabla_w l\left(h\left(x, w\right), y\right)\big|_{w=w_p}\right]^\top \cdot \Delta + \frac{\alpha}{2}\Delta^\top\Delta\right] \\ &= L_{\mathcal{D}^{(k_1)}}\left(h_p\right) + \left[J_p^{(k_1)}\right]^\top \cdot \Delta + \frac{\alpha}{2}\Delta^\top\Delta \end{aligned} \tag{21}$$

Next, by (16) and (21), we have

$$\begin{aligned} L_{src}\left(h_{p+1}\right) &\leq \frac{1}{K^2}\sum_{k_1,k_2} L_{\mathcal{D}^{(k_1)}}\left(h_{p+1}^{(k_2)}\right) \\ &\leq \frac{1}{K^2}\sum_{k_1,k_2}\left[L_{\mathcal{D}^{(k_1)}}\left(h_p\right) + \left[J_p^{(k_1)}\right]^\top \cdot \Delta + \frac{\alpha}{2}\Delta^\top\Delta\right] \\ &= \frac{1}{K^2}\sum_{k_1,k_2}\left[L_{\mathcal{D}^{(k_1)}}\left(h_p\right) - \lambda_{p+1} \cdot \left[J_p^{(k_1)}\right]^\top \cdot J_p^{(k_2)}\right. \\ &\quad \left. + \frac{\alpha}{2} \cdot \left(\lambda_{p+1}\right)^2 \cdot \left\|J_p^{(k_2)}\right\|_2^2\right] \\ &= L_{src}\left(h_p\right) - \lambda_{p+1}\|J_p\|_2^2 + \frac{\alpha}{2K} \cdot \left(\lambda_{p+1}\right)^2 \cdot \sum_{k_2}\left\|J_p^{(k_2)}\right\|_2^2. \end{aligned} \tag{22}$$

The last equality follows from the definition of J_p.

Define $\sigma_p^2 := \frac{1}{K}\sum_{k=1}^K \left\|J_p^{(k)}\right\|_2^2 - \|J_p\|_2^2$. Then, we have

$$\begin{aligned} L_{src}\left(h_{p+1}\right) &\leq L_{src}\left(h_p\right) - \lambda_{p+1}\|J_p\|_2^2 + \frac{\alpha}{2} \cdot \left(\lambda_{p+1}\right)^2 \cdot \left(\sigma_p^2 + \|J_p\|_2^2\right) \\ &= L_{src}\left(h_p\right) - \left(\lambda_{p+1} - \frac{\alpha \cdot \left(\lambda_{p+1}\right)^2}{2}\right)\|J_p\|_2^2 + \frac{\alpha \cdot \left(\lambda_{p+1}\right)^2}{2}\sigma_p^2 \\ &= L_{src}\left(h_p\right) - \beta_1(\lambda_{p+1})\|J_p\|_2^2 + \beta_2(\lambda_{p+1})\sigma_p^2, \end{aligned} \tag{23}$$

which completes the proof. □

Lemma B.4 (Lemma 4.1)**.** *Under Assumptions 4.2 and 4.1, and the cross-client statistics defined in Definition 4, after P rounds of federated pretraining one has*

$$\widehat{L}_{tgt}^* \leq L_{src}\left(h_0\right) - \sum_{p=0}^{P-1}\beta_1(\lambda_{p+1})\|J_p\|_2^2 + \sum_{p=0}^{P-1}\beta_2(\lambda_{p+1})\sigma_p^2 + d_{\mathcal{G},\mathcal{F}}(\mathcal{D}_S^{fed}, \mathcal{D}_T), \tag{24}$$

where h_0 is the initial global model and $\beta_2(\lambda) = \frac{\alpha\lambda^2}{2},\ \beta_1(\lambda) = \lambda - \beta_2(\lambda)$.

Proof. From Lemma B.3, we have

$$\begin{aligned} L_{src}\left(h_P\right) &\leq L_{src}\left(h_{P-1}\right) - \beta_1(\lambda_P)\|J_{p-1}\|_2^2 + \beta_2(\lambda_P)\sigma_{P-1}^2 \\ &\leq L_{src}\left(h_0\right) - \sum_{p=0}^{P-1}\beta_1(\lambda_{p+1})\|J_p\|_2^2 + \sum_{p=0}^{P-1}\beta_2(\lambda_{p+1})\sigma_p^2. \end{aligned}$$

https://doi.org/10.52202/079017-1233

Following the same proof as the one outlined for Theorem 1, we obtain

$$L_{tgt}(h_P) \leq L_{src}(h_P) + d_{\mathcal{G},\mathcal{F}}(\mathcal{D}_S^{fed}, \mathcal{D}_T).$$

Therefore,

$$\begin{aligned}\widehat{L}_{tgt}^{*} = L_{tgt}(h_P) &\leq L_{src}(h_P) + d_{\mathcal{G},\mathcal{F}}(\mathcal{D}_S^{fed}, \mathcal{D}_T) \\ &\leq L_{src}\left(h_0\right) - \sum_{p=0}^{P-1} \beta_1(\lambda_{p+1})\|J_p\|_2^2 + \sum_{p=0}^{P-1} \beta_2(\lambda_{p+1})\sigma_p^2 + d_{\mathcal{G},\mathcal{F}}(\mathcal{D}_S^{fed}, \mathcal{D}_T).\end{aligned}$$

□

C Additional Experimental Results

Computing resources. We used an NVIDIA GeForce RTX 3090 Ti GPU with a memory of 24247MiB. One run of experimental evaluation takes approximately 6 hours for CIFAR10 → SVHN for the small client setting.

Fraction of Clients Participating in FL Rounds. Besides the results generated using 20% of the participating clients reported in the main text, we also report the results pertaining to 10% and 100% participating clients in Table 4. A larger fraction of participants helps with improving transferability, i.e., as expected, 100% participation outperforms the setting with 10% participation. However, we note that for 20% of participating clients we already reach a performance comparable to that involving 100% of the participants.

Method	Coefficient	10% participants	100% participants
FedAVG	NA	0.52 ± 0.01	0.56 ± 0.03
FedGTST	1e-4	0.52 ± 0.02	0.58 ± 0.01
	5e-4	$\mathbf{0.58 \pm 0.04}$	$\mathbf{0.63 \pm 0.05}$
	1e-3	0.53 ± 0.01	0.60 ± 0.06

Table 4: Transferability versus the fraction of participating clients in each round. We report the results for CIFAR10 → SVHN on the LeNet backbone, with $K = 100$.

Convergence Results. Convergence results are plotted in Figure 1. We observe that a model pretrained via FedGTST not only offers better transferability than baselines, but also tends to converge faster during the finetuning stage (i.e., the green lines in all plots always converge faster than the grey dashed lines).

Local Epochs. Besides the results for single local client epochs presented in main text, here we also report the results for multiple local epochs in Table 5. A smaller number of local epochs tends to improve transferability more than a larger number of epochs, which is consistent with the intuition that FL pretraining can avoids local overfitting when using fewer epochs.

	Coeffient	1 local epoch	10 local epochs
FedAVG	NA	0.54 ± 0.05	0.52 ± 0.07
FedGTST	1e-4	$\mathbf{0.60 \pm 0.01}$	0.52 ± 0.04
	5e-4	0.56 ± 0.06	$\mathbf{0.58 \pm 0.02}$
	1e-3	0.58 ± 0.05	0.53 ± 0.02

Table 5: Transferability v.s. local number of epochs. We report results for CIFAR10→SVHN with the LeNet backbone, for 10% of participating clients and $K = 100$.

Cross-Client Statistics. We plot the cross-client averaged Jacobian norm $\|J_p\|_2$ and the cross-client Jacobian variance σ_p^2 in Figure 2. The observation is that FedGTST leads to a significantly larger $\|J_p\|_2$ and a significantly smaller σ_p^2 compared to FedAVG. A more detailed explanation of the findings (i.e., the reason for truncating the x-axis in the left plot, the FedGTST coefficient selection approach and faster convergence results in the right plot) is available in the caption of Figure 2.

Additional Dataset: DomainNet. Following FedSR, we apply a leave-one-out strategy, where one domain is treated as the target domain and the other five are source domains, allocated to five

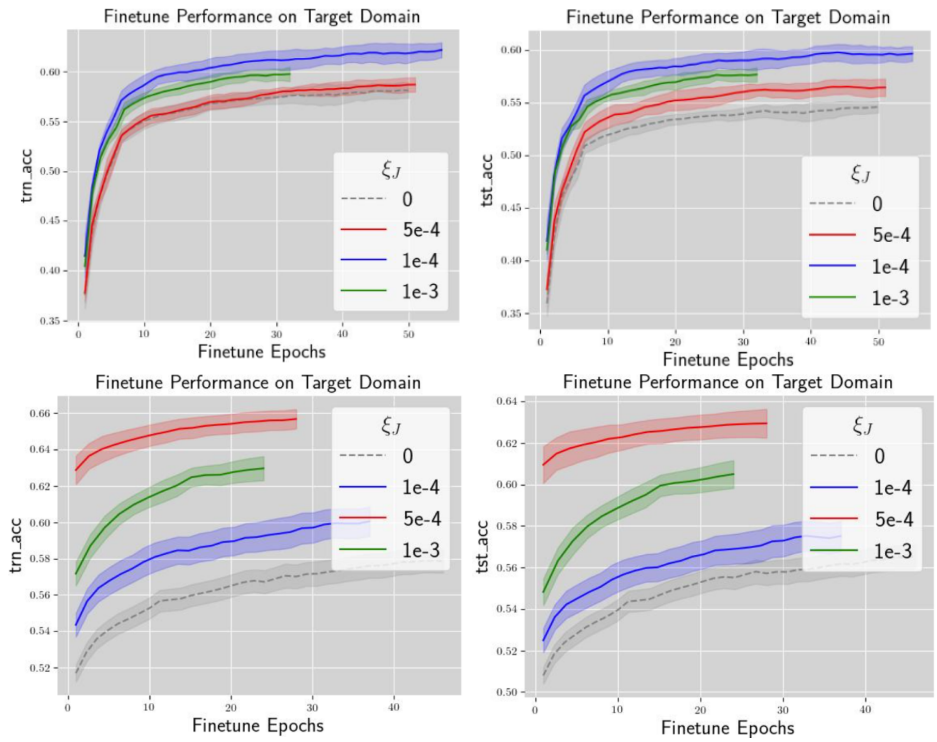

Figure 1: Visualization of Convergence Results. We use CIFAR10 $\rightarrow$ SVHN with $K = 100$ as an example. The top two plots correspond to a fraction of 10% of participating clients, while the bottom two plots correspond to 100% participation. We report the training and test accuracy along with finetuned epochs for both settings. The grey dashed lines represent FedAVG, where the coefficient for the regularizer term is set to 0. Other lines represent FedGTST with tuned coefficients.

individual clients. Results are reported in Table 6, where the target domains listed are: C (Clipart), I (Infograph), P (Painting), Q (Quickdraw), R (Real), and S (Sketch).

Model	Leave-One-Out Domain					Average
	C	I	P	Q	R	
FedAVG	59.3±0.7	16.5±0.9	44.2±0.7	10.8±1.8	57.2±0.8	39.6
FedSR	61.0±0.6	18.6±0.4	45.2±0.5	13.4±0.6	57.6±0.2	41.3
FedGTST	**63.9±0.5**	**20.7±0.3**	**47.8±0.4**	**15.2±0.5**	**59.5±0.6**	**43.6**

Table 6: Comparison of different federated models on intra-domain transfer tasks of DomainNet. FedGTST consistently outperforms the other methods.

Additional Baseline: Scaffold. FedGTST consistently outperforms Scaffold, as presented in Table 7 and 8.

Method	MNIST$\rightarrow$ MNIST-M		CIFAR10 $\rightarrow$ SVHN		Average
	LeNet	ResNet	LeNet	ResNet	
Scaffold	75.6 $\pm$ 0.8	80.8 $\pm$ 0.3	66.0 $\pm$ 0.5	71.1 $\pm$ 0.4	73.3
FedGTST	**76.2** $\pm$ 0.9	**82.3** $\pm$ 0.5	**70.1** $\pm$ 0.8	**74.5** $\pm$ 0.3	**75.8**

Table 7: Comparison between Scaffold and FedGTST. Number of clients is 10.

Dirichlet Sampling. We set the concentration parameter to 0.5 and the number of parties to 10 by default. The results in Table 9 and 10 indicate that FedGTST still outperforms others when individual domains are constructed via Dirichlet sampling.

https://doi.org/10.52202/079017-1233

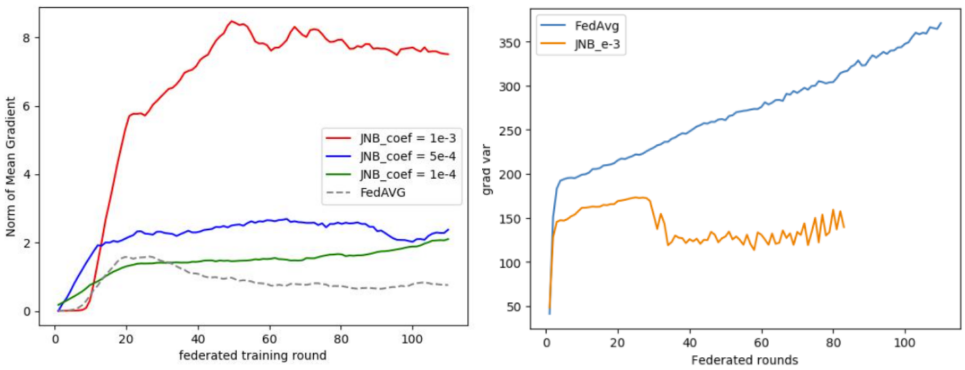

Figure 2: Cross-client statistics tuning via FedGTST. We use CIFAR10→SVHN with $K = 100$ as an example. The left plot reports the global Jacobian (gradient) norm versus the index of the federated round. The grey dashed line represents FedAVG, while other lines correspond to FedGTST with different coefficients. We truncate the plot to only capture the results of the first 100 rounds, since at the end of training the gradient norm should drop to a value close to 0 due to convergence, and we are only interested in observing the behaviour of Jacobian norms during relative early pretraining stages. We select the best-performing setup from the left plot (the red line with coefficient $1e-3$), and then in the right plot, compare its variance during a federated round with that of FedAVG. The blue line represents FedAVG and the yellow line corresponds to FedGTST. The yellow line terminated earlier since all experiments are averaged over 3 runs and aligned with the run that converges the fastest.

Method	MNIST→ MNIST-M		CIFAR10 → SVHN		Average
	LeNet	ResNet	LeNet	ResNet	
Scaffold	52.3 ± 0.5	63.1 ± 0.3	45.5 ± 0.1	55.5 ± 0.3	54.1
FedGTST	**57.5** ± 0.3	**67.6** ± 0.2	**52.4** ± 0.1	**63.1** ± 0.2	**60.2**

Table 8: Comparison between Scaffold and FedGTST. Number of clients is 100.

D Discussion of the Positivity of Coefficients Requirement for Cross-Client Statistics

Lemma D.1 (Bound on round-wise source loss)**.** *Suppose the learning rates of all clients at round p are the same: $\lambda_p^{(k)} = \lambda_p, \forall k \in [K], p \in [P]$. When Assumption 4.2 and 4.1 hold, we have that*

$$L_{src}\left(h_{p+1}\right) \leq L_{src}\left(h_p\right) - \beta_1(\lambda_{p+1})\|J_p\|_2^2 + \beta_2(\lambda_{p+1})\sigma_p^2 \tag{25}$$

where $J_p = \frac{1}{K}\sum_k J_p^{(k)}$, $\sigma_p^2 = \frac{1}{K}\sum_k \left\|J_p^{(k)}\right\|_2^2 - \left\|\frac{1}{K}\sum_k J_p^{(k)}\right\|_2^2$, and $\beta_1(\lambda) = \lambda - \beta_2(\lambda)$, $\beta_2(\lambda) = \frac{\alpha\lambda^2}{2}$.

Remark. Lemma D.1 provides an upper bound of *current-round cross-client loss* (the proof is given in Appendix B), indicating that a smaller *current-round cross-client loss* (LHS) may be influenced by factors such as a small *last-round cross-client loss*, a large *cross-client average Jacobian norm*, and a small *cross-client Jacobian variance* (RHS). More precisely, the LHS is the *cross-client current-round loss*, which is upper bounded by three terms on the RHS,

1) *The Loss Term*: $L_{src}\left(h_p^{(0)}\right)$ is the *last-round cross-client loss*.

2) *The Variance Term*: σ_p^2 measuring the variance of local gradients across all nodes.

3) *The Norm Term*: $\|J_p\|_2^2$ measures the squared *cross-client average Jacobian norm*.

Positivity of the Coefficients. It is straightforward to see that $\beta_2(\lambda_{p+1}) = \frac{\alpha(\lambda_{p+1})^2}{2}$, the coefficient in front of σ_p^2, is always positive. Therefore, we focus on $\beta_2(\lambda_{p+1})$, the coefficient in front of $\|J_p\|_2$. Denote the RHS of Equation 25 by $\mathcal{B}$, so that $L_{src}(h_{p+1}) \leq \mathcal{B}$. To allow such a bound to be used as an indicator that the source loss is decreasing with the number of training rounds, we require

Method	MNIST→ MNIST-M		CIFAR10 → SVHN		Average
	LeNet	ResNet	LeNet	ResNet	
FedAVG	74.1±0.6	82.2±0.3	65.2±0.4	72.8±0.9	73.5
FedSR	75.5±0.8	82.0±0.2	66.1±0.5	72.1±0.3	73.9
FedIIR	75.8±0.2	**82.8**±0.6	66.5±0.9	74.6±0.1	74.9
Fed-GTST	**77.3**±0.8	82.7±0.4	**70.8**±0.7	**75.2**±0.2	**76.5**

Table 9: Comparison of target accuracy (%) across different methods on MNIST→MNIST-M and CIFAR10→SVHN tasks. 10 individual domains are constructed by Dirichlet sampling.

Method	MNIST→ MNIST-M		CIFAR10 → SVHN		Average
	LeNet	ResNet	LeNet	ResNet	
FedAVG	50.4±0.1	63.0±0.3	43.3±0.2	54.6±0.5	52.8
FedSR	52.7±0.2	62.9±0.3	44.7±0.1	56.5±0.3	54.2
FedIIR	54.2±0.4	64.1±0.1	47.4±0.4	58.7±0.2	56.1
Fed-GTST	**59.5**±0.3	**69.2**±0.2	**55.1**±0.1	**65.6**±0.2	**62.4**

Table 10: Comparison of target accuracy (%) across different methods on MNIST→MNIST-M and CIFAR10→SVHN tasks. 100 individual domains are constructed by Dirichlet sampling.

$\mathcal{B} \leq L_{src}(h_p)$, since only in this way can the bound give $L_{src}(h_{p+1}) \leq \mathcal{B} \leq L_{src}(h_p)$. Thus, we require $\beta_1(\lambda_{p+1})$ to be positive, since otherwise $\mathcal{B} \leq L_{src}(h_p)$ cannot be meet. We describe in what follows that a positive $\beta_1(\lambda_{p+1})$ is indeed possible in practice.

Realistic scenarios in which $\beta_1(\lambda_{p+1}) > 0$. To require $\beta_1(\lambda_{p+1}) > 0$ is equivalent to require $\lambda_{p+1} < \frac{2}{\alpha}$. This requirement can be easily met since for any model of a known mathematical form based on a second-order differentiable loss, we can easily get the lower bound for α once we observed all training data. Using linear regression as an example, where $l(x, y; w) = (wx - y)^2$, we have $\frac{d^2 l}{dw^2} = 2x$, therefore, in this case we can simply control $\lambda_{p+1} \leq \frac{1}{\max_{x \in \mathcal{X}} \|x\|}$ to approximately guarantee $\lambda_{p+1} < \frac{2}{\alpha}$. This then ensures $\beta_1(\lambda_{p+1}) > 0$.

E Optimal Learning Rate

Choosing the Optimal Learning Rate. Lemma B.3 shows that the upper bound of federated loss at round p is a quadratic function w.r.t the learning rate λ_p. Therefore, a good learning rate at each round needs to be chosen to minimize the upper bound. For simplicity of notation, we use $B_{p+1}(\lambda_{p+1})$ as a shorthand for the upper bound shown in (15).

The following two observations are in place for $B_{p+1}(\lambda_{p+1})$:

- When $0 < \lambda_{p+1} < \frac{2K \cdot \|J_p\|_2^2}{\alpha \cdot \sum_k \|J_p^{(k)}\|_2^2} \leq \frac{2}{\alpha}$, it holds that $L_{src}\left(h_{p+1}^{(0)}\right) \leq B_{p+1}(\lambda_{p+1}) < L_{src}\left(h_p^{(0)}\right)$, indicating that the federated loss is decreasing with the number of rounds.
- By minimizing $B_{p+1}(\lambda_{p+1})$ w.r.t. λ_{p+1}, we have

$$\lambda_{p+1}^* = \frac{K \cdot \|J_p\|_2^2}{\alpha \cdot \sum_k \|J_p^{(k)}\|_2^2}, \quad B_{p+1}(\lambda_{p+1}^*) = L_{src}\left(h_p^{(0)}\right) - \frac{K \cdot \|J_p\|_2^4}{2\alpha \cdot \sum_k \|J_p^{(k)}\|_2^2}.$$

F Stochastic Gradient Descend

Assumption 4.2 on one step of gradient descent can be extended to stochastic learning with batch sampling. The additional randomness in sampling would require incorporating the variance of batch sampling into the generalization bound. This variance term, being independent of the algorithm design, was omitted in our theoretical analysis.

https://doi.org/10.52202/079017-1233

G Dataset Description

MNIST comprises $60,000$ 28×28 grayscale images of handwritten digits (0 through 9); MNIST-M is created by combining MNIST digits with the patches randomly extracted from color photos of BSDS500 as their background, containing $59,001$ training images. The CIFAR-10 dataset consists of $60,000$ 32×32 colour images from 10 classes. SVHN contains $73,257$ 32×32 colored digits obtained from house numbers in Google Street View images.

H Non-iid FL Models

For the source domains MNIST and CIFAR-10, we only allow each local client to have access to two out of ten classes (e.g., for the digit dataset (0-9), one client may only have access to say digits 3 and 6). We let each client randomly chooses their labels and samples following a uniform distribution. See the example in Fig. 3

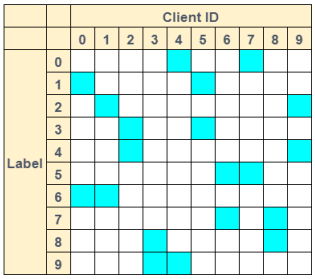

Figure 3: An example for constructing a non-iid marginal distribution for the *Cifar10* dataset allocated to 10 clients. Each client has access to only two labels. We also make sure that no samples are used by more than one client.

I Additional Related Work

I.1 Gradient Matching in Transfer Learning

Discussion. Gradient matching in transfer learning focuses on aligning gradients between source and target domains to facilitate better domain adaptation. Shi et al. [43] demonstrate that matching gradients across domains enhances domain generalization by making the learned representations more resilient to domain shifts. Extending this idea, Rame et al. [39] introduce the concept of invariant gradient variances, which helps maintain generalization performance even for out-of-distribution settings. The work in Pezeshki et al. [38] further highlights the impact of Hessian alignment, showing that aligning the Hessians of the source and target domains can significantly boost generalization in gradient-based methods.

Challenges. Applying gradient alignment techniques directly to FL presents several challenges: (1) *Privacy leakage*—these methods can potentially compromise data privacy by necessitating access to the target domain from source domains; (2) *Local overfitting*—clients train their models on local data, which can lead to overfitting within their specific domains, reducing the global model's generalization capabilities.

Our contribution. To address the issues described above, we propose communication schemes that ensure data privacy by preventing unrestricted access to client data. Furthermore, our approach promotes global transferability by focusing on improving generalization across all clients rather than solely enhancing local domain performance.

I.2 Distinctions and Connections between Generalization and Transferability in FL

Both approaches address the challenge of non-iid data, and improving transferability may potentially enhance generalization across diverse local domains. However, they employ distinct models and

evaluation datasets: (a) Generalization of FL targets performance on heterogeneous *source testing datasets*, while transferable FL aims for strong performance on a *target dataset* that may significantly differ from *source training datasets*. (b) Heterogeneous FL utilizes the pretrained model for evaluation, whereas Transferable FL assesses the finetuned model.

The methodologies also diverge. Although reducing cross-client variance is linked to better generalization, our method distinguishes itself from traditional heterogeneous FL by enforcing a large average Jacobian norm $|J_p|$ in the early stages. While a larger $|J_p|$ may hinder generalization due to increased local updates and model variance, it enhances transferability by preventing premature local convergence.

Additionally, adversarially robust models offer an example where improving transferability may come at the expense of generalization. As shown in [41], adversarially robust models tend to have better transferability. However, since these models are designed to perform well against adversarial examples or perturbations, they do not necessarily exhibit lower generalization error on clean test data. In this scenario, transferability can be unrelated to or even conflict with generalization.

 https://doi.org/10.52202/079017-1233

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 abstract and introduction reflect the facts that 1) we theoretically established that certain cross-client statistics can boost transferability of federated learning, and that 2) based on our theoretical findings, we proposed a novel transferable federated learning method which significantly outperforms baselines.

   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 discuss the limitations of our work in a separate section, Limitations.

   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 provided necessary assumptions need for our theoretical analysis in the main text but deferred the proofs to the Appendix due to space limitations. We nevertheless discussed the intuition behind our findings in the main text.

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 datasets are existing open-source benchmarks, and we have included our code in the Supplementary materials document.

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.

https://doi.org/10.52202/079017-1233

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

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

   Answer: [Yes]

   Justification: The datasets are existing open-source benchmarks, and we have attached our code in a Supplementary materials document.

   Guidelines:

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

6. **Experimental Setting/Details**

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

   Answer: [Yes]

   Justification: Details of the experimental work are discussed in the Experimental Setting in the Experiments section.

   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: Our experimental results are test accuracies, and we reported the standard deviations.

   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: Sufficient information about the compute resources (type of compute workers, memory, time of execution) is provided in the Experiments section.

   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 conducted to produce the paper conformed, in every respect, 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: [NA]

    Justification: There is neither a negative or positive societal impact of the work performed. It still may have a technological and academic impact based on its results and findings. There should be no problems with the proper use of the pertinent methods/software.

https://doi.org/10.52202/079017-1233

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 risk in this domain.

    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: The original papers that produced the dataset used in our studies are properly cited.

    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 will not lead to the release of 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.

https://doi.org/10.52202/079017-1233

- 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.