Optimal Transport Based Dynamic Weighted Federated Aggregation to Mitigate Data Poisoning Attacks

Anonymous CVPR submission

Paper ID 9680

Abstract

Federated learning (FL) enables multi-agents to collaborate with a central server to build a global model without sharing private and sensitive information. While FL provides scalable privacy-preserving training capabilities, it is susceptible to adversarial poisoning attacks. Existing defense techniques, including Byzantine robust aggregation rules, against data poisoning attacks in FL, have several limitations. (i) Trade-off between precision and robustness, (ii) assumptions related to asymptotic optimal-bounds on error rates of parameters, (iii) i.i.d. data distributions, and (iv) strong-convex nature of optimization problem. To overcome these limitations, we propose federated learning optimal transport (FLOT), a dynamic weighted federated aggregation technique to mitigate data poisoning attacks. We leverage Wasserstein barycentric approach to obtain a global model from a given set of local models trained privately on client devices. Further, we propose loss function-based rejection (LFR) to suppress malicious updates and provide a set of weighted coefficients to the Wasserstein barycentric optimization function. We demonstrate the effectiveness of the proposed FLOT framework on three benchmark datasets, namely, GTSRB, KBTS, and CIFAR10. Experimental results show that the proposed FLOT aggregation outperforms existing baselines by ≈ 2% and ≈ 9% under single and multi-client attack settings, respectively. Further, we show that the performance of our FLOT is better than state-of-the-art by a significant margin on the datasets mentioned above.

1. Introduction

Multi-agent federated learning (FL) has attracted the attention of researchers because of its use in several applications, such as signal processing, mobile user personalization, speech recognition, and more [1–3]. Federated learning can simultaneously offload the computation and memory-intensive training work onto multiple low-end computation devices, referred to as clients [4, 5]. FL functions as a highly distributed decentralized system preserving data privacy [6] with limited communication, and computational capabilities [7].

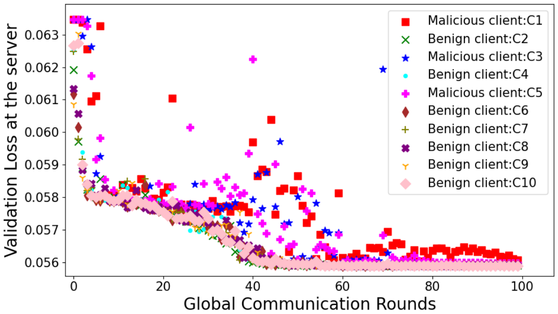

Figure 1. Validation losses of individual client model at the server for 100 global communication rounds under 33% multi-attack settings for KBTS dataset. Here, the global model is updated with the remaining good-performing client updates. For the next iteration, the clients train their local models using this new global model.

Despite the advantages of privacy and shared intelligence, FL with deep neural networks (DNN) faces unique challenges w.r.t. data and system heterogeneity, computation and memory constraints [8, 9] and adversarial poisoning attacks [10].

Past works [11–13] in FL have exposed its high vulnerability to adversarial attacks under the *white-box* setting. However, they are somewhat unrealistic as the attacker needs to have complete knowledge of the model structure and parameters distributed across all the clients. Our state-of-the-art study has shown that the FL research community is showing more interest in investigating the *black-box* adversarial attacks [14]. For example, in autonomous vehicles, where the FL setting is more relevant, these attacks can misdirect the vehicle controller, which might result in catastrophic events. In this paper, *we focus on untargeted data poisoning attack* in FL as it is the most common and relevant to production deployments [15]. Specifically, the attacker is interested in *generic misclassification (untargeted) rather than specific misclassification (targeted)*.

Existing defenses against FL data poisoning attacks are either based on anomaly detection or using a novel model aggregation to dampen the poisoned update effectiveness. The anomaly detection-based techniques detect malicious clients by grouping various aspects of client updates and removing those that are not part of a majority group. And these techniques vary according to different aspects of client updates [16]. Recently, Rieger *et al*. proposed DeepSight [17], which uses deep inspection of the updates in NN layers to detect anomalous updates. Further, the aggregation techniques such as Krum [18], trimmed mean [19], and median [19] claim to be Byzantine robust. However, they have some critical limitations in common. For example, they only provide asymptotic bounds that are far from practice. Specifically, they only offer the order-optimal bounds on error rates of parameters. However, even if such order-optimal bounds are given, there is no guarantee for classification performance on the learned global model. Further, they strongly assume that data is divided in an *i.i.d.* fashion, and the optimization problem is strongly convex, which is not practical in real-world scenarios. Hence, to overcome the above limitations, we propose federated learning optimal transport (FLOT), an OT-based dynamic weighted federated aggregation, to mitigate poisoning attacks. Our defense is based on the hypothesis that the updates from a malicious client doing data poisoning will differ from benign client updates in terms of loss of validation data at the server. Figure. 1 shows the validation loss of 10 clients under multi-client attack settings. We observe a clear dispersion in the malicious clients' loss values for initial rounds. Further, when we remove them for each round, the models eventually converge after 60 rounds.

Hence, our technique is based on loss function-based rejection (LFR) that suppresses updates from high-loss performing malicious updates and then applies OT optimization to smoothen the aggregated global model. Specifically, we leverage the benefits of Wasserstein barycenters in order to obtain a global model from a given set of local models. Further, the LFR provides the weighted coefficients for the Wasserstein barycentric function that helps in discarding the malicious updates. The main contributions of this paper are:

1. Explored optimal transport-based optimization to mitigate data poisoning attacks in federated learning.
2. Proposed a dynamic weighted federated aggregation method called FLOT for secure aggregation of gradient updates on a global server.
3. Presented the proposed FLOT time complexity as $\mathcal{O}(n.d)$ which is a significant improvement over $\mathcal{O}(n^2.d)$ of the Krum function [18].
4. The proposed FLOT method is evaluated on three baseline methods and three Byzantine robust aggregation rules under no attack, single-client attack, and multi-client attack (33% Byzantine) settings.
5. The experiments are demonstrated on three datasets, namely, GTSRB [20], KBTS [21], and CIFAR10 [22], that signify the potential of the proposed FLOT approach.

2. Related Work

This section reviews existing literature and relates to our proposed methodology from two perspectives: adversarial attacks in federated learning and optimal transport methods in machine learning.

2.1. Adversarial Attacks in Federated Learning

Adversarial attacks against ML models and DNN have received much attention [23–25] in recent years. There has been vast research interest in adversarial attacks for deep neural networks, which can cause potential security incidents even with small perturbations [23]. This interest has trickled into privacy-preserving federated learning [26, 27] as researchers have begun exploring it in adversarial settings [11]. Since clients in FL communicate local model updates to the central server, adversarial attacks in FL are usually performed either through client data (data poisoning) or model updates (model poisoning). Broadly, there are two major types of adversarial attacks in FL: targeted & untargeted data poisoning and model poisoning. Below, we discuss the current works on different kinds of FL attacks.

Data Poisoning Attacks: In this category, the attacker or the malicious client tries creating data (i.e., poisonous) that - through local model updates - leads to an incorrect or imprecise global model. The black-box adversarial attack in multi-agent communication was first propagated in [28] using a computationally expensive surrogate-based approach. Zhang *et al.* [12, 29], Hitaj *et al.* [30], Wang *et al.* [31] proposed a generative adversarial attack (GAN) based poisoning attack method in the context of federated learning. Tolpegin *et al.* [32] studied targeted data poisoning attacks against FL systems in which a malicious subset of the participants aim to poison the global model by sending model updates derived from mislabeled data. Wang *et al.* [33] proposed an edge-case backdoor attack that forces a model to misclassify on seemingly straightforward inputs that are, however, unlikely to be part of the training or test data and live on the tail of the input distribution.

Model Poisoning Attacks: In this second category, the attacker directly sends malicious updates [11, 34]. Bhagoji *et al.* [11, 34] have focused on targeted model poisoning as opposed to data poisoning of prior works. Bagdasaryan *et al.* [35] proposed a backdoor FL attack framework that trains on the backdoor data using a constraint-and-scale technique and sends the resulting corrupted model as an update to the central server. Shejwalkar *et al.* [36] proposed a general model poisoning attack by computing the malicious model update through maximally perturbing the benign reference aggregate in the malicious direction.

In this paper, we focus on defending untargeted data poisoning in FL as we find it to be significantly most common and relevant to production deployments [15]. Also, data poisoning attacks can affect a large population of FL clients and remain undetected for a longer period.

2.2. Optimal Transport in Machine Learning

Optimal Transport (OT) theory has been gaining significant attention from the machine learning community due to its efficiency in modeling various ML applications [37]. *Computer vision:* early works [38] used OT formulations (Wasserstein distance) in computer vision applications to find the dissimilarity measure between the images. Also, OT is used to perform the image-to-image color transfer, the color of a source image to match the color of a target image of the same scene [39, 40]. *GANs:* research has been done to improve generative adversarial networks (GANs) using OT [41–43]. Liu *et al.* [44] proposed WGAN-QC, a WGAN with quadratic transport cost (Optimal Transport Regularizer) to stabilize the training process of WGAN-QC and prove that it converges to a local equilibrium point with finite discriminator updates per generator update. *Semantic correspondence:* Liu *et al.* [45] tries to establish dense correspondences across semantically similar images by solving the many-to-one matching and background-matching issues using OT. *Domain adaptation:* early works [46, 47] introduced a regularized optimal transport model in an unsupervised way to align the representations in the source and target domains. *Graph matching:* Gromov *et al.* [48] proposed a novel Gromov-Wasserstein learning framework to jointly match (align) graphs and learn embedding vectors for the associated graph nodes. Finally, very few works used OT to improve the federated learning system [49, 50]. However, to the best of our knowledge, there is no explicit use of OT in FL to defend against data-poisoning attacks. We are the first to model a defense mechanism using the OT framework.

3. Proposed Approach

This section presents the proposed FLOT approach to mitigate data poisoning attacks as shown in Figure 2.

3.1. Overview of Federated Learning

Federated Learning involves bringing machine learning (ML) capabilities to local clients for building models with local datasets, ensuring their data privacy. It consists of a total of n clients, each with access to local data D_i, where i is the client's index, $i \in n$, and $1 \leq i \leq n$. Each client maintains its own copy of the data shard as private, such that $D_i = \{x^i_1 ... x^i_{l_i}\}$ and $|D_i| = l_i$ is not shared with the server.

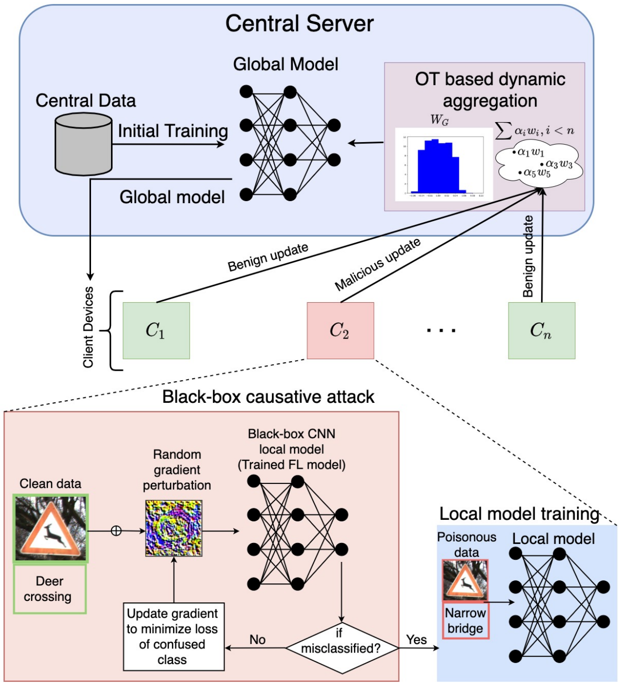

Figure 2. Overview of FLOT integrated into a FL system with n clients ($C_1, C_2, \ldots, C_n$). The malicious client (C_2) is poisoning the training data. The central server receives the gradients and performs FLOT to obtain the global model W_G.

Initialization: The server generates the initial global model by training on some amount of auxiliary data.

Client execution: At each global epoch t, every client: (i) tries to minimize the empirical loss over its data shard and trains the classification algorithm with a batch size of b_s for E local epochs with the initial global model w^t_G, (ii) after the completion of training phase with E local epochs, all client(s) calculate the local updates using $\Delta C^n_{t+1} = w^n_{t+1} - w^G_t$, and (iii) these individual client model updates are sent back to the central server for model aggregation.

Server execution: At each global epoch t, the central server: (i) sends the current version of the global model to update all n agents, (ii) receives the local client updates and performs global model aggregation using synchronous federated weighted averaging [9] as $w^G_{t+1} = w^G_t + \sum_{n \in n} \lambda_n \Delta C^n_{t+1}$, where, $\lambda_n = \frac{l_n}{\sum l_n}$ and $\sum_n \lambda_n = 1$, i.e., the updates are used to generate the 'aggregated' global model synchronously for t global epochs, and (iii) performs global model testing using the updated global model on the test data at the server.

Further, as federated weighted averaging is a naive aggregation rule that averages the local model parameters to obtain the global model parameters, it is widely used under non-adversarial settings [9,51]. However, federated averaging is not robust under adversarial settings as the attacker can manipulate the global model parameters arbitrarily for this mean aggregation rule when compromising only one client device [18, 19]. Hence, we take an optimal transport-based dynamic aggregation approach to improve federated averaging and mitigate data poisoning attacks in FL.

We ensure a *non-i.i.d.* (independent and identically distributed) dataset by splitting the dataset randomly among clients with the number of samples as $l_k >= \zeta$, where ζ is the minimum threshold to ensure proper FL protocol. An overview of the proposed FLOT in federated learning is shown in Figure 2.

3.2. Threat Model

In our threat model, the main goal of the adversary is to poison the global model so that it makes wrong predictions on clean test data. The threat model assumes a black-box attack scenario wherein the adversary has little to no knowledge about the model architecture and has no access to its gradients. The adversary can only access the local model's predicted class labels and probability scores to generate random gradient perturbations.

Our threat model also consists of a single (fixed attacker clients) and multi-client attack (random malicious clients) scenarios. The adversary can poison the central server only through the local model update that is poisoned using malicious data. Avoiding a single point of failure, the aggregation algorithm is considered beyond the attacker's control. In addition, malicious clients cannot directly poison other benign participants' learning phase or training data $\mathbb{D}_{benign}$ until and unless the node is explicitly specified as an adversary node. This implies that clients marked malicious must

3.3. Designing the Adversarial Attack

The attacker performs the following steps for attacking the FL setup: (i) generate adversarial samples based on the gradient-based black-box attack method, (ii) add these samples to the local training dataset, (iii) train the local model, (iv) finally transmitting the malicious updates to poison the global model.

Black-box attack method: In this paper, we consider the M-SimBA data poisoning attack proposed by Kumar *et al.* [52] as it is recent and powerful to other gradient-based black-box attack methods. In order to generate an adversarial image, a random gradient perturbation is added to the original image and is calculated as $I_{adv} = I_x + \epsilon * G_p$, where I_{adv} is the adversarial image, I_x is the original image, and G_p is the randomized gradient perturbation. The step size (ϵ) controls the intensity of perturbation. The algorithm repeats the above process until it generates the final adversarial image as per Eq. (3.3). In addition, it converges on the L2 norm such that $||I_{adv} - I_x||_2 < \theta$. Threshold parameter θ controls the deviation of adversarial image w.r.t. original image *without*, making it perceivable to the human eye. More details about the attack can be found in supplementary material.

3.4. Proposed Defense Mechanism

In this subsection, we discuss the background of optimal transport (OT), problem formulation, and proposed algorithm for dynamic model aggregation to discard the poisonous model updates.

Overview of optimal transport (OT): Gaspard Monge introduced OT [53], [54] to find the most efficient way to move a unit of mass between two distributions. The aim is to minimize the overall ground cost to move the unit mass from the source distribution to the target distribution. The optimization problem can be given as

$$\min_{t,\, t \neq \mu_s = \mu_t} \int C(a, t(a))\, d\mu_s(a), \tag{1}$$

where μ_s, μ_t correspond to source and target distributions, respectively. $C(.,.)$ is the ground cost of moving a unit mass between two positions x, $t(x)$. The constraint $t \neq \mu_s = \mu_t$ ensures that the source is completely transported to the target. In general, the OT solution is used in two main aspects, (i) to find the optimal value that measures the similarity between two distributions, also known as Wasserstein distance. (ii) To find the OT matrix, which is the optimal correspondence mapping between distributions.

Wasserstein Barycenters [55]: It is a distribution that minimizes the weighted sum of Wasserstein distance w.r.t all other distributions. It aims to find a distribution μ such that

$$\min_{\mu} \sum_{n} \alpha_n \mathbb{W}(\mu, \mu_n), \tag{2}$$

where α_i represent the weight of distribution μ_i, $\mathbb{W}(.,.)$ correspond to Wasserstein distance between distributions given by

$$\mathbb{W}(\mu, \mu_n) = \inf_{\gamma \in \Gamma_{\mu,\mu_n}} \mathop{\mathbb{E}}_{(\mathcal{X},\mathcal{Y} \sim \gamma)} ||\mathcal{X} - \mathcal{Y}||_2^2, \tag{3}$$

where inf is take over couplings between μ and μ_n.

Problem formulation: Let us assume we are at t^{th} communication round in federated learning such that the server receives the model updates from all the n clients, and $\mathbb{D}_v$ is the validation data at the server. Let $\{w_1^t, w_2^t, \ldots, w_n^t\}$ are model updates that correspond to $\{c_1, c_2, \ldots, c_n\}$ clients, respectively. Also, let us assume there are ρ unknown malicious client updates $\rho < n$. Now, the aim is to find a global model weight w_G^t that minimizes its weighted Wasserstein distance w.r.t other benign client model weights $w_{[1,2,\ldots,n]}$ after dynamically discarding the malicious updates as shown in Figure 3.

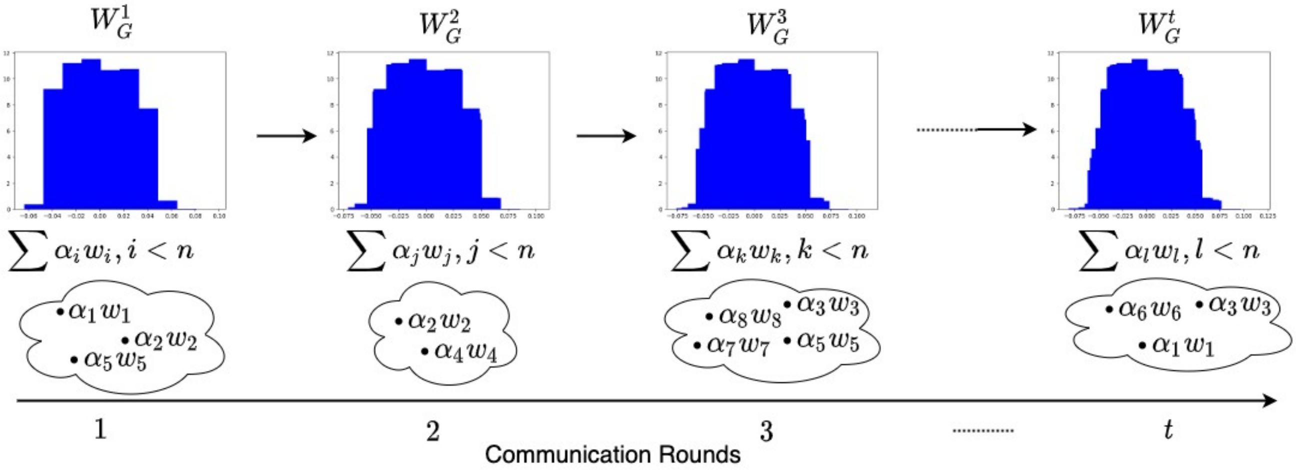
W_G^1
W_G^2
W_G^3
W_G^t
$\sum \alpha_i w_i, i < n$
$\sum \alpha_j w_j, j < n$
$\sum \alpha_k w_k, k < n$
$\sum \alpha_l w_l, l < n$
$\alpha_1 w_1$ $\alpha_2 w_2$ $\alpha_5 w_5$
$\alpha_2 w_2$ $\alpha_4 w_4$
$\alpha_8 w_8$ $\alpha_3 w_3$ $\alpha_7 w_7$ $\alpha_5 w_5$
$\alpha_6 w_6$ $\alpha_3 w_3$ $\alpha_1 w_1$
1
2
3
t
Communication Rounds

Figure 3. Visualization of global model W_G interpolation for t communication rounds. Our proposed FLOT framework helps in suppressing malicious clients by dynamically assigning $\alpha_i's$ to the local client models w_j at each communication round. The clients with $\alpha = 0$ are discarded otherwise weighted coefficients are assigned to their respective clients.

Different variations of OT optimization: Recent developments in OT have resulted in different variations of OT optimization. *(i) Regularized OT:* It is expressed as

$$\gamma^* = argmin_{\gamma \in \mathbb{R}_+^{mxn}} \sum_{i,j} \gamma i, j\mathbb{M}_{i,j} + \lambda\omega(\gamma)$$

$$s.t.\gamma 1 = a; \gamma^T 1 = b; \gamma \geq 0, \tag{4}$$

where $\mathbb{M} \in \mathbb{R}_+^{mxn}$ is the cost matrix to move mass from bin a_i to bin b_j, a, b are histograms that represent the weight of each sample in the source and target distributions. ω is the regularization term. *(i) Entropic Regularized OT:* Marco Cuturi [56] smooth the classic OT problem with an entropic regularization term and show that the resulting optimum is also a distance that can be computed through Sinkhorn's matrix scaling algorithm faster than that of transport solvers. It is expressed as $\gamma_\lambda^* = diag(u)\mathbb{K}diag(v)$, where u, v are vectors and $K = exp(-M/\lambda)$ and exp is taken componentwise. In addition, there are other regularizations such as *quadratic* $(\omega(\gamma) = \sum_{i,j} \gamma_{i,j}^2)$, that have a similar effect to entropic regularization yet keeps some sparsity that is lost when $\lambda > 0$ [57]. *Group lasso regularization* given by $(\omega(\gamma) = \sum_{j,G \in \zeta} ||\gamma_{G,\zeta}||_q^p)$, where ζ contains non-overlapping groups of lines in the OT matrix [58]. Further, there are other optimizations and problem formulations such as Wasserstein discriminant analysis [59], unbalanced OT [60], etc. Finally, our proposed optimization is intuition-based with respect to defending against data poisoning attacks in FL. We formulate our problem statement in terms of Wasserstein barycenter as per Eq. 2.

Definition 3.1 *($(\omega, \rho\chi)$ - Byzantine Resilience). Let $[\mathcal{N}] = \{w_1, \dots, w_n\}$ be any non-independent indentically distributed (non-i.i.d.) local clients models in $\mathbb{R}^d$. Let $[\mathcal{R}] = \{w_1, \dots, w_\rho\}$ be any non-i.i.d. Byzantine local clients models. Let $[\mathcal{X}] = \{w_1, \dots, w_\chi\}$ be any highly non-i.i.d. benign local clients models. An aggregation rule $\mathcal{A}$ is said to be $(\omega, \rho\chi)$-Byzantine Resilience) if for any $1 \leq \dots \leq i_1 \dots \leq i_\rho \dots \leq j_1 \leq \dots \leq j_\chi \leq \dots n$, vector*

$$\mathcal{A} = \mathcal{A}(w_1, \dots, w_1', \dots, w_\rho', \dots, w_1'', \dots, w_\chi'', \dots, w_n) \tag{5}$$

satisfies the following

$$\sum_{k \in ([\mathcal{N}] - [\mathcal{R}])} (\mathcal{L}(\mathbb{D}_v, w_k) \leq \sum_{k \in [\mathcal{N}]} (\mathcal{L}(\mathbb{D}_v), w_k), \tag{6}$$

$$\sum_{k \in ([\mathcal{N}] - [\mathcal{X}])} (\mathcal{L}(\mathbb{D}_v, w_k) \leq \sum_{k \in [\mathcal{N}]} (\mathcal{L}(\mathbb{D}_v), w_k), \tag{7}$$

$$\left\| \sum_{k \in [\mathcal{N}]} (\mathcal{L}(\mathbb{D}_v, w_k) - \sum_{k \in [\mathcal{N}']} (\mathcal{L}(\mathbb{D}_v, w_k) \right\| \geq \omega, \tag{8}$$

for some $\omega \geq 0$. Here, $\mathcal{N} = [\mathcal{N}'] - ([\mathcal{R}] + [\mathcal{X}])$, $\mathcal{L}(\mathbb{D}_v, w_k)$ denote the loss of w_i model on validation data $\mathbb{D}_v$.

Proposed optimization: We introduce FLOT, an OT-based $(\omega, \rho\chi)$ - Byzantine Resilient (as defined in Definition 3.1, we give the detailed proof in the supplementary material in Proposition 1) dynamic weighted federated aggregation rule to mitigate poisoning attacks. Blanchard *et al.* [18] prove that *no linear combination* of the vectors can tolerate a single Byzantine worker (refer Corollary 1 in supplementary material). Specifically, federated averaging [9] is not Byzantine resilient. Existing Byzantine robust algorithms like Krum [18] select the local model updates representative of most client models by computing the pairwise distances between individual models. However, when the data across the workers are highly non-*i.i.d.*, there is no 'representative' client model. The local client models show high variance with respect to each other as they compute their local gradient over vastly diverse local data. Hence, for convergence, it is crucial to not only select a good (non-Byzantine) local model but also ensure that each of the good models is selected with roughly equal frequency. Further, when applied to non-*i.i.d.* datasets, Krum performs poorly even without any attack [61]. This is because Krum primarily selects models from $n - c - 2$ (where c is the number of malicious clients), local models whose pairwise distances are closer to others. Hence, the robust aggregation rules may fail on realistic non-*i.i.d.* datasets.

To circumvent this issue, we consider LFR with OT optimization to develop Wasserstein barycentric aggregation rule (FLOT). In the end, through our experimental results, we show that our FLOT also serves as a robust client selection technique in discarding the benign clients that do not perform well on the validation data. This implies that dropping some less performing benign updates helps to improve the accuracy, which also supports the claims of the recent work, DivFL [62].

Now, we explain our FLOT framework. To start with, we find the optimal coefficients set of the client model weights α based on loss on validation data $\mathbb{D}_v$, i.e., $\mathcal{L}_v$ of every client model w_i. It can be formulated as

$$\alpha \leftarrow \mathcal{L}_v(w, \mathbb{D}_v), \tag{9}$$

$$\alpha' \leftarrow |\alpha - \max(\alpha)|. \tag{10}$$

Now, we define a set $\alpha'_0 = \alpha'$ and write

$$\beta_1 := \{b \in \alpha'_0 : b \leq a \,\forall\, a \in \alpha'_0\}. \tag{11}$$

Next, we define $\alpha'_1 := \alpha'_0 \setminus \beta_1$ which discards the highly malicious weight coefficient from the set α'_0. Further, we inductively write

$$\beta_k := \{b \in \alpha'_{k-1} : b \leq a \,\forall\, a \in \alpha'_{k-1}\}, \tag{12}$$

$$\alpha'_k := \alpha'_{k-1} \setminus \beta_k, \tag{13}$$

such that α'_k is the final set after discarding k malicious client updates whose $\alpha' = 0$[1]. Further, we normalize α'_k to $[0, 1]$ through the softmax of all weighting factors, which is defined as:

$$\alpha'_k = \frac{e^{\alpha'_k}}{\sum_{k=1}^{n} e^{\alpha'_k}}. \tag{14}$$

Now, our optimization problem can be formulated in terms of Wasserstein barycenter as per Eq. 2 as

$$FLOT(w_1^t, w_2^t, \ldots, w_n^t) \leftarrow \min_{w_G^t} \sum_k \alpha'_k \mathbb{W}(w_G^t, w_k), \tag{15}$$

where t is the global communication round.

Lemma 3.2 *The expected time complexity of our FLOT function $FLOT(w_1^t, w_2^t, \ldots, w_n^t)$, where, $w_1^t, w_2^t, \ldots, w_n^t$ are d-dimensional vectors is $\mathcal{O}(n.d)$.*

Proof. Firstly, the parameter server computes the maximum of loss values $(\alpha_1, \alpha_2, \ldots, \alpha_n)$ and updates all its elements $|\alpha - max(\alpha)|$ (time $\mathcal{O}(n.d)$). Then the server selects the loss that is less than a certain threshold (expected time $\mathcal{O}(nlog(n).d)$ with a binary search). Next, it computes the set difference to discard the highly malicious weight vector (time $\mathcal{O}(n.d)$). Finally, the server normalizes the remaining $n - k$ values (time $\mathcal{O}(n.d)$). Hence, adding all the times, we obtain the overall time complexity of $FLOT$ as $\mathcal{O}(n.d)$.

We report that **our proposed FLOT time complexity is $\mathcal{O}(n.d)$ which is a significant improvement over $\mathcal{O}(n^2.d)$ of the Krum function [18].**

4. Convergence Analysis

In this section, we analyze the convergence of global model aggregation for convex problems under assumptions of non-identically distributed data, full device participation, and local model updating. Our FLOT optimization function, as per Eq. 15 is given by

$$FLOT(w_1, w_2, \ldots, w_n) \leftarrow \min_{w_G} \sum_k \alpha'_k \mathbb{W}(w_G, w_k). \tag{16}$$

Rewriting it, we get the FLOT barycenter functional as

$$w_G^* \in \operatorname*{arg\,min}_{w \in \mathcal{P}_2(\mathbb{R}^d)} \alpha'_k \sum_{i=1}^{k} \mathbb{W}_2^2(w_G, w_k) =: 2FLOT(w_G)\text{[2]}, \tag{17}$$

(*from Wasserstein-2 spaces* ($\mathbb{W}_2^2$)*- it is the metric space of probability measures* $\mathcal{P}_2(\mathbb{R}^d)$*, equipped with Wasserstein distance as given in Eq. 3*). The aim is to minimize $FLOT(w_G)$. Further, we can write the Wasserstein gradient of the above formulation using Brenier map [63] as

$$\nabla FLOT(w_G) = -\alpha'_k \sum_{i=1}^{k} (\mathcal{T}_{w_G \to w_i} - \tau), \tag{18}$$

where $\mathcal{T}_{w_G \to w_i}$ is the Brenier map, τ is the identity that gives the displacement map of w_G. Finally, the gradient

[1] Since all the local models are trained on different amounts of non-*i.i.d.* data, all $\alpha'_i s$ are different, where $i \in [1, n]$.

[2] We scaled to one half so that when the derivate is taken the term 2 goes away.

descent of the global model over $\mathbb{W}$ metric space is given by

$$\begin{aligned} w_G^{t+1} &= (\tau - \eta_t \nabla FLOT(w_G))_{\#} w_G^t \\ &\implies w_G^t - (\tau - \eta_t \nabla FLOT(w_G)) \\ &= (\tau + \alpha'_k \sum_{i=1}^{k} (\mathcal{T}_{w_G \to w_i} - \tau)_{\#} w_G^t; (Eq. 18) \\ &= (1 - \eta_t) w_G^t + \eta_t \alpha'_k \sum_{i=1}^{k} \mathcal{T}_{w_G \to w_i}(w_G^t). \end{aligned} \tag{19}$$

Further, we apply the Polyak-Łojasiewicz (PL) inequality [64] given by

$$f(x) - \inf f \leq C||\nabla f(x)||^2, \forall x, \tag{20}$$

followed by smoothness of gradient [65] given by

$$f(y) - f(x) \leq \langle \nabla f(x), y - x \rangle + \frac{\beta}{2}||y - x||^2, \tag{21}$$

for some function $f(x)$, the derivative of f as $\nabla f(x)$ and constant C, to prove the linear rate (exponentially) of convergence for gradient descent. Finally, the linear rate of convergence of FLOT for gradient descent is given by

$$FLOT(w_G^{t+1}) - FLOT(w_G^t) \lessapprox e^{-\frac{\alpha'_k}{2C} t}. \tag{22}$$

5. Experiments

In this section, we demonstrate the efficacy of the proposed FLOT on the three datasets, namely, the German traffic sign recognition benchmark (GTSRB) [20], KUL Belgium traffic sign (KBTS) [27], and CIFAR10 [66]. We scale the three datasets to an average resolution of 150 × 150 for our experimentation. We set $\zeta = 900$ samples in local client data shards for three datasets. Based on the data availability, we set the total number of clients as 30, 10, and 30 for GTSRB, KBTS, and CIFAR10 datasets.

We build a custom 4-layer CNN architecture followed by two fully connected layers and treat this as a global model. The model is trained with images of size 150 × 150 using categorical cross-entropy as loss function optimized using Adam optimizer. During the training of the global classifier for 200 epochs through FL protocol, each client trains for $E = 5$ local epochs on the local data with a batch size $b_s = 64$ and with a learning rate of $l_r = 0.01$. All the clients are trained individually and sequentially at each global epoch. More details about the dataset, CNN architecture and the related source codes are given in the supplementary material.

5.1. Attack Configurations

We use a black-box and active data poisoning attack MSimBA [52] (3.3), a recent gradient-based technique to generate adequate poisoned data. We set $\epsilon = 0.7$ and the maximum iterations for MSimBA as 1000. We use two attacker settings, namely *single-client* and *multi-client*. For multi-client attack scenarios, we set 33% of randomly chosen clients as malicious (following the Byzantine client set up in [18]).

5.2. Baselines and Configurations

We use the following baselines and configurations of FLOT to evaluate its effectiveness:

1. *FL [9]:* Normal federated learning without any defense. Ideally, should perform similarly to this baseline under **no attack** scenarios.
2. *Random Sampling (RS) of the Clients:* This represents the FL system with random sampling, where the server randomly selects some updates for aggregation. As our FLOT involves generating loss function-based weighted coefficients that drop the malicious clients, followed by OT optimization, it should perform better than RS.
3. *Power-of-choice [67]:* In this work, the server selects the clients with the largest training losses.
4. *DivFL [62]:* This is a recent work that proposes a technique to perform FL by selecting a group of clients based on submodular optimization.
5. *FLOT Configurations:* We use two configurations of FLOT, namely, FLOT (our method) and RS+FLOT (our method includes random sampling for better results).

We use the following Byzantine Robust Aggregation approaches to perform a comparative evaluation:

1. *Krum [18]:* As explained in Section 3.4, Krum selects one local model updates that are representative of a majority of client models. We set $c = 10$ for GTSRB and CIFAR10 datasets and $c = 3$ for the KBTS dataset to handle the 33% malicious clients in our experimentation.
2. *TM [19]:* Trimmed mean (TM) aggregates each dimension of input updates separately and sorts the values along the i^{th}-dimension. Then it removes x largest and smallest values of that dimension and computes the average of the rest. We consider the suggested configuration of $x = 5$ for GTSRB, CIFAR10, and $x = 1$ for KBTS datasets to handle the 33% malicious clients in our experimentation.
3. *Median [19]:* The median aggregates each dimension of input updates separately and sorts the values of the i^{th}-dimension. Then it takes the median as the i^{th} parameter of the global model.

5.3. Effectiveness

Table 1 gives the performance of the proposed FLOT framework in comparison to the baselines and Byzantine

Table 1. Comparison of global test accuracy (%) with existing baseline methods and Byzantine aggregation rules. **Bold** result indicates the best result for setting. (c/n) represents the ratio of the number of malicious clients to the total number of clients.

		GTSRB [20]			**KBTS** [27]			**CIFAR10** [66]		
Comparison	**Method**	**No attack (0/30)**	**Single-client attack (1/30)**	**Multi-client attack (10/30)**	**No attack (0/10)**	**Single-client attack (1/10)**	**Multi-client attack (3/10)**	**No attack (0/30)**	**Single-client attack (1/30)**	**Multi-client attack (10/30)**
Baselines	FL [9]	**87.8**	83.24	70.63	**90.02**	83.26	73.14	91.23	85.03	73.83
	RS	86.68	84.45	65.45	87.92	84.24	70.53	90.54	82.98	75.33
	Power-of-choice [67]	87.56	81.29	63.72	88.05	80.27	69.38	92.64	73.86	70.84
	DivFL [62]	87.12	82.63	72.08	89.96	81.63	71.63	92.86	74.12	71.19
Byzantine Robust Aggregation	Krum [18]	86.72	85.80	79.98	89.97	84.29	77.72	91.46	85.12	81.33
	TM [19]	84.32	82.87	77.45	88.52	84.09	72.96	90.64	84.43	80.64
	Median [19]	85.23	83.39	78.64	88.27	84.97	75.26	89.91	83.36	81.62
Ours	FLOT	86.24	85.12	81.12	89.12	**85.94**	**79.94**	91.51	85.21	82.26
	RS+FLOT	87.01	**85.98**	**82.26**	89.36	85.02	78.02	**92.37**	**86.24**	**83.54**

robust aggregation methods on three benchmark datasets, namely, GTSRB, KBTS, and CIFAR10. Our FLOT configurations consistently outperform the other methods for all the datasets. Under the no-attack setting, our approach closely performed to that of the FL baseline with $< 1\%$ difference for GTSRB and KBTS dataset and outperformed for CIFAR10 dataset. This is due to a large number of classes with inter and intra-class variability in the GTSRB and KBTS dataset that led to the discarding of benign clients models with a slight difference in the loss values. Also, the FedAvg tries to achieve the local-optimum error rate when the objective function is strongly convex under no attack. On the contrary, given a good amount of data, our FLOT configuration was able to sample updates that improved performance under no attack on the CIFAR10 dataset.

Further, our FLOT outperformed all the baselines under single and multi-client attack settings. For GTSRB and CIFAR10 datasets with 30 clients, we observe that RS+FLOT performed 82.26% and 83.54%, respectively, which is better than FLOT. For single-client attacks, as the number of benign clients is one less than the total clients, they try to dampen the effect of the single malicious client. Hence, all baselines and Byzantine aggregation techniques are performed on a similar scale. Comparatively, our FLOT configurations outperformed other methods and rules by more than 0.5%. Power-of-choice and DivFL are effective client selection techniques under clean data settings; hence, their optimizations perform poorly under attack settings. Also, the non-*i.i.d.* data distribution among the client and strong data poisoning attack resulted in a performance drop of Krum as it is based on strong *i.i.d.* assumption [18]. In addition, trimming model updates after sorting client updates, including median, results in considering malicious updates for aggregation. For GTSRB and CIFAR10 datasets with 30 clients, we observed that RS+FLOT performed better than FLOT. This is due to the availability of a large number of clients, and RS already discards some clients that may be malicious. Hence, applying FLOT on top of RS is effective. On the contrary, applying RS on KBTS data with only 10 clients discards some clients, and further discarding using FLOT has resulted in lower performance. Hence, the aggregated global model performs poorly under higher attack percentages. Detailed performance plots are provided in the supplementary material.

In summary, our OT-based optimization using Wasserstein barycenters allows us to effectively interpolate between multiple client updates [68] by warping them using the loss-based weighted coefficients that are dynamically suppressing the malicious updates. FLOT configurations outperformed all the baselines under two different attack settings and are close to the FL baseline with less than 1% *difference under no attack for two datasets. Also, our FLOT configurations outperformed the existing Byzantine robust aggregation techniques by more than* 0.5% *and more than* 1% *under single-client and multi-client attacks. This implies that FLOT is Byzantine robust under non-i.i.d. data poisoning attacks.*

6. Conclusion

In this paper, we proposed an optimal transport-based dynamic weighted federated aggregation to mitigate untargeted data poisoning attacks in an FL framework. The proposed FLOT framework effectively interpolates the global model update using the proposed loss-based weighted coefficients. It leverages the OT optimization using Wasserstein barycenters to obtain smoothed global model after discarding the malicious updates. We show through our experimental results that the proposed FLOT configurations achieve better classification performance under single and 33% Byzantine workers compared to other methods, including Byzantine robust aggregation rules. Also, the time complexity analysis shows an improvement over the Krum aggregation rule by a factor of n, where n is the number of clients. In the future, we will explore different variations of OT optimizations, including regularization to account for higher maliciousness ($> 33\%$ attackers), higher order of non-*i.i.d.*ness, and other data and model poisoning attacks.

References

[1] J. Sun, T. Chen, G. B. Giannakis, Q. Yang, and Z. Yang, "Lazily aggregated quantized gradient innovation for communication-efficient federated learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 1

[2] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020. 1

[3] A. M. Elbir and S. Coleri, "Federated learning for vehicular networks," *arXiv preprint arXiv:2006.01412*, 2020. 1

[4] C. Fang, Y. Guo, Y. Hu, B. Ma, L. Feng, and A. Yin, "Privacy-preserving and communication-efficient federated learning in internet of things," *Computers & Security*, vol. 103, p. 102199, 2021. 1

[5] C. Fang, Y. Guo, N. Wang, and A. Ju, "Highly efficient federated learning with strong privacy preservation in cloud computing," *Computers & Security*, vol. 96, p. 101889, 2020. 1

[6] G. Xu, H. Li, S. Liu, K. Yang, and X. Lin, "Verifynet: Secure and verifiable federated learning," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 911–926, 2019. 1

[7] X. Guo, Z. Liu, J. Li, J. Gao, B. Hou, C. Dong, and T. Baker, "V eri fl: Communication-efficient and fast verifiable aggregation for federated learning," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 1736–1751, 2020. 1

[8] V. Smith, C.-K. Chiang, M. Sanjabi, and A. Talwalkar, "Federated multi-task learning," *arXiv preprint arXiv:1705.10467*, 2017. 1

[9] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial Intelligence and Statistics*, pp. 1273–1282, PMLR, 2017. 1, 3, 4, 5, 7, 8

[10] O. Ibitoye, M. O. Shafiq, and A. Matrawy, "Differentially private self-normalizing neural networks for adversarial robustness in federated learning," *Computers & Security*, vol. 116, p. 102631, 2022. 1

[11] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, "Analyzing federated learning through an adversarial lens," in *International Conference on Machine Learning*, pp. 634–643, ICML, 2019. 1, 2

[12] J. Zhang, J. Chen, D. Wu, B. Chen, and S. Yu, "Poisoning attack in federated learning using generative adversarial nets," in *2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, pp. 374–380, IEEE, 2019. 1, 2

[13] N. Rodríguez-Barroso, E. Martínez-Cámara, M. Luzón, G. G. Seco, M. Á. Veganzones, and F. Herrera, "Dynamic federated learning model for identifying adversarial clients," *arXiv preprint arXiv:2007.15030*, 2020. 1

[14] V. Mothukuri, R. M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, and G. Srivastava, "A survey on security and privacy of federated learning," *Elsevier Future Generation Computer Systems*, vol. 115, pp. 619–640, 2021. 1

[15] V. Shejwalkar, A. Houmansadr, P. Kairouz, and D. Ramage, "Back to the drawing board: A critical evaluation of poisoning attacks on production federated learning," in *2022 IEEE Symposium on Security and Privacy (SP)*, pp. 1354–1371, IEEE, 2022. 1, 3

[16] S. Shen, S. Tople, and P. Saxena, "Auror: Defending against poisoning attacks in collaborative deep learning systems," in *Proceedings of the 32nd Annual Conference on Computer Security Applications*, pp. 508–519, 2016. 2

[17] P. Rieger, T. D. Nguyen, M. Miettinen, and A.-R. Sadeghi, "Deepsight: Mitigating backdoor attacks in federated learning through deep model inspection," *arXiv preprint arXiv:2201.00763*, 2022. 2

[18] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," *Advances in Neural Information Processing Systems*, vol. 30, 2017. 2, 4, 5, 6, 7, 8

[19] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *International Conference on Machine Learning*, pp. 5650–5659, PMLR, 2018. 2, 4, 7, 8

[20] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "The German Traffic Sign Recognition Benchmark: A multi-class classification competition," in *IEEE International Joint Conference on Neural Networks*, pp. 1453–1460, 2011. 2, 7, 8

[21] M. Mathias, R. Timofte, R. Benenson, and L. Van Gool, "Traffic sign recognition — how far are we from the solution?," in *The 2013 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2013. 2

[22] A. Krizhevsky, V. Nair, and G. Hinton, "The cifar-10 dataset," 2017. 2

[23] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014. 2

[24] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 IEEE Symposium on Security and Privacy (sp)*, pp. 39–57, IEEE, 2017. 2

[25] A. Salem, R. Wen, M. Backes, S. Ma, and Y. Zhang, "Dynamic backdoor attacks against machine learning models," in *2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P)*, pp. 703–718, IEEE, 2022. 2

[26] L. Zhang, Y. Luo, Y. Bai, B. Du, and L.-Y. Duan, "Federated learning for non-iid data via unified feature learning and optimization objective alignment," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4420–4428, October 2021. 2

[27] M. Mathias, R. Timofte, R. Benenson, and L. Van Gool, "Traffic sign recognition — how far are we from the solution?," in *The 2013 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2013. 2, 7, 8

[28] J. Tu, T. Wang, J. Wang, S. Manivasagam, M. Ren, and R. Urtasun, "Adversarial attacks on multi-agent communication," *arXiv preprint arXiv:2101.06560*, 2021. 2

[29] J. Zhang, B. Chen, X. Cheng, H. T. T. Binh, and S. Yu, "Poisongan: Generative poisoning attacks against federated learning in edge computing systems," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3310–3322, 2020. 2

[30] B. Hitaj, G. Ateniese, and F. Perez-Cruz, "Deep models under the gan: information leakage from collaborative deep learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 603–618, 2017. 2

[31] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang, and H. Qi, "Beyond inferring class representatives: User-level privacy leakage from federated learning," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pp. 2512–2520, IEEE, 2019. 2

[32] V. Tolpegin, S. Truex, M. E. Gursoy, and L. Liu, "Data poisoning attacks against federated learning systems," in *European Symposium on Research in Computer Security*, pp. 480–501, Springer, 2020. 2

[33] H. Wang, K. Sreenivasan, S. Rajput, H. Vishwakarma, S. Agarwal, J.-y. Sohn, K. Lee, and D. Papailiopoulos, "Attack of the tails: Yes, you really can backdoor federated learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 16070–16084, 2020. 2

[34] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, "Model poisoning attacks in federated learning," in *Proc. Workshop Secur. Mach. Learn.(SecML) 32nd Conf. Neural Inf. Process. Syst.(NeurIPS)*, 2018. 2

[35] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *International Conference on Artificial Intelligence and Statistics*, pp. 2938–2948, PMLR, 2020. 2

[36] V. Shejwalkar and A. Houmansadr, "Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning," in *NDSS*, 2021. 3

[37] L. C. Torres, L. M. Pereira, and M. H. Amini, "A survey on optimal transport for machine learning: Theory and applications," *arXiv preprint arXiv:2106.01963*, 2021. 3

[38] Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," *International journal of computer vision*, vol. 40, no. 2, pp. 99–121, 2000. 3

[39] H. Alghamdi, M. Grogan, and R. Dahyot, "Patch-based colour transfer with optimal transport," in *2019 27th European Signal Processing Conference (EUSIPCO)*, pp. 1–5, IEEE, 2019. 3

[40] J. Rabin, S. Ferradans, and N. Papadakis, "Adaptive color transfer with relaxed optimal transport," in *2014 IEEE international conference on image processing (ICIP)*, pp. 4852–4856, IEEE, 2014. 3

[41] G. Avraham, Y. Zuo, and T. Drummond, "Parallel optimal transport gan," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4411–4420, 2019. 3

[42] T. Salimans, H. Zhang, A. Radford, and D. Metaxas, "Improving gans using optimal transport," *arXiv preprint arXiv:1803.05573*, 2018. 3

[43] J. Adler and S. Lunz, "Banach wasserstein gan," *Advances in neural information processing systems*, vol. 31, 2018. 3

[44] H. Liu, X. Gu, and D. Samaras, "Wasserstein gan with quadratic transport cost," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4832–4841, 2019. 3

[45] Y. Liu, L. Zhu, M. Yamada, and Y. Yang, "Semantic correspondence as an optimal transport problem," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4463–4472, 2020. 3

[46] N. Courty, R. Flamary, A. Habrard, and A. Rakotomamonjy, "Joint distribution optimal transportation for domain adaptation," *Advances in Neural Information Processing Systems*, vol. 30, 2017. 3

[47] S. P. Singh and M. Jaggi, "Model fusion via optimal transport," *Advances in Neural Information Processing Systems*, vol. 33, pp. 22045–22055, 2020. 3

[48] H. Xu, D. Luo, H. Zha, and L. C. Duke, "Gromov-wasserstein learning for graph matching and node embedding," in *International conference on machine learning*, pp. 6932–6941, PMLR, 2019. 3

[49] F. Farnia, A. Reisizadeh, R. Pedarsani, and A. Jadbabaie, "An optimal transport approach to personalized federated learning," *arXiv preprint arXiv:2206.02468*, 2022. 3

[50] H. Wang, M. Yurochkin, Y. Sun, D. Papailiopoulos, and Y. Khazaeni, "Federated learning with matched averaging," *arXiv preprint arXiv:2002.06440*, 2020. 3

[51] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, *et al.*, "Large scale distributed deep networks," *Advances in neural information processing systems*, vol. 25, 2012. 4

[52] K. N. Kumar, C. Vishnu, R. Mitra, and C. K. Mohan, "Black-box adversarial attacks in autonomous vehicle technology," in *2020 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*, pp. 1–7, IEEE, 2020. 4, 7

[53] G. Monge, "Mémoire sur la théorie des déblais et des remblais," *Histoire de l'Académie Royale des Sciences de Paris*, 1781. 4

[54] L. V. Kantorovich, "On the translocation of masses," *Journal of mathematical sciences*, vol. 133, no. 4, pp. 1381–1382, 2006. 4

[55] M. Agueh and G. Carlier, "Barycenters in the wasserstein space," *SIAM Journal on Mathematical Analysis*, vol. 43, no. 2, pp. 904–924, 2011. 4

[56] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," *Advances in neural information processing systems*, vol. 26, 2013. 5

[57] M. Blondel, V. Seguy, and A. Rolet, "Smooth and sparse optimal transport," in *International conference on artificial intelligence and statistics*, pp. 880–889, PMLR, 2018. 5

[58] R. Flamary, N. Courty, D. Tuia, and A. Rakotomamonjy, "Optimal transport for domain adaptation," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 1, 2016. 5

[59] R. Flamary, M. Cuturi, N. Courty, and A. Rakotomamonjy, "Wasserstein discriminant analysis," *Machine Learning*, vol. 107, no. 12, pp. 1923–1945, 2018. 5

[60] C. Frogner, C. Zhang, H. Mobahi, M. Araya, and T. A. Poggio, "Learning with a wasserstein loss," *Advances in neural information processing systems*, vol. 28, 2015. 5

[61] L. He, S. P. Karimireddy, and M. Jaggi, "Byzantine-robust learning on heterogeneous datasets via resampling," 2020. 6

[62] R. Balakrishnan, T. Li, T. Zhou, N. Himayat, V. Smith, and J. Bilmes, "Diverse client selection for federated learning via submodular maximization," in *International Conference on Learning Representations*, 2021. 6, 7, 8

[63] L. Ambrosio, N. Gigli, and G. Savaré, *Gradient flows: in metric spaces and in the space of probability measures*. Springer Science & Business Media, 2005. 6

[64] H. Karimi, J. Nutini, and M. Schmidt, "Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition," in *Joint European conference on machine learning and knowledge discovery in databases*, pp. 795–811, Springer, 2016. 7

[65] V. Mai and M. Johansson, "Convergence of a stochastic gradient method with momentum for non-smooth non-convex optimization," in *International conference on machine learning*, pp. 6630–6639, PMLR, 2020. 7

[66] A. Krizhevsky, G. Hinton, *et al.*, "Learning multiple layers of features from tiny images," 2009. 7, 8

[67] Y. J. Cho, J. Wang, and G. Joshi, "Client selection in federated learning: Convergence analysis and power-of-choice selection strategies," *arXiv preprint arXiv:2010.01243*, 2020. 7, 8

[68] J. Lacombe, J. Digne, N. Courty, and N. Bonneel, "Learning to generate wasserstein barycenters," *Journal of Mathematical Imaging and Vision*, pp. 1–17, 2022. 8