OTPA: FEDERATED MODEL POISONING VIA OPTIMAL TRANSPORT ALIGNMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated defenses often assess client behavior through update geometry, but parameter proximity need not imply benign functional influence. Motivated by neural permutation symmetry, Optimal Transport Poisoning Attack (OTPA) transports a stored poisoned source toward a clean, round-specific anchor before interpolation. Its entropic maps combine approximate feature alignment with mixing, without honest-client updates or explicit amplification. We specify transport propagation for convolutional, residual, vision-transformer, and BERT-style architectures. The primary vision benchmark compares nine baselines while distinguishing attacker-local from oracle information. OTPA reduces accuracy from 66.0% to 14.9% on Tiny-ImageNet and from 66.7% to 39.4% on CIFAR-10 while its updates remain close to the centre of the submitted population. On all three datasets it is Pareto-optimal among the evaluated attacker-local attacks, reducing the median update distance by 70% to 90% relative to matched unaligned interpolation at no more than 2.5 accuracy points of difference. Further evaluations cover architecture transfer, language, heterogeneous data, and partial participation. OTPA yields the lowest defended accuracy under six robust aggregators, while history-based filtering exposes highly consistent malicious directions. These results establish transport-based source construction as a poisoning mechanism: proximity and directional consistency reveal different aspects of an update, so neither coordinate alignment nor low Euclidean distance establishes benign behavior.

1 INTRODUCTION

Federated learning (FL) trains a shared model without collecting clients' raw data (McMahan et al., 2017). Because participation is open by design, any client can submit whatever parameters it chooses, and a server that cannot inspect training data must judge trustworthiness from the submission alone. This leaves the server to assess the behavior of submitted parameters. Robust aggregation therefore relies on update norms, coordinates, pairwise distances, or shared directions (Bagdasaryan et al., 2020; Bhagoji et al., 2019; Fang et al., 2020; Shejwalkar & Houmansadr, 2021; Blanchard et al., 2017; Yin et al., 2018), while attacks seek impact under these geometric constraints.

These defenses treat parameter coordinates as directly comparable, but neural representations are not unique: permuting a hidden feature and inversely permuting the adjacent layer preserves the network function. Functionally compatible models can therefore place corresponding features in different coordinates, inflating distance and mixing mismatched features during interpolation. Benign model fusion addresses this mismatch through alignment (Singh & Jaggi, 2020; Imfeld et al., 2024). This symmetry motivates OTPA's correspondence search. Its dense entropic maps perform approximate alignment together with contraction and mixing, rather than an exact function-preserving reparameterization.

OTPA combines a persistent poisoned source with a clean anchor conditioned on the current round. When compromised client k participates in round t, it trains B_k^t from the current broadcast, aligns its stored source M_k with that anchor, and submits

$$F_k^t = (1-\alpha)B_k^t + \alpha A(M_k; B_k^t), \tag{1}$$

where A transports compatible hidden coordinates and α controls the source contribution. The anchor follows the current training state, while transport couples source features to anchor coordinates

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

before interpolation. We propagate transport maps through convolutional channels, residual paths, embeddings, attention projections, and multilayer perceptron (MLP) coordinates without changing the server protocol or observing honest submissions. Figure 1 summarizes the pipeline.

The paper makes three contributions:

- We introduce poisoning by aligning a persistent adversarial source with a clean, round-conditioned anchor before constructing each submission.
- We derive a unified OT procedure for convolutional neural networks (CNNs), residual networks, vision transformers, and BERT-style encoders, and analyze its components, sensitivity, descriptor error, and computational cost.
- We compare OTPA with nine baselines on three vision datasets, separating attacker-local and oracle information assumptions, test SST-2, heterogeneous data, partial participation, and four architecture families, and evaluate all ten attack conditions under six robust aggregators, DnC, FoolsGold, and FLAME.

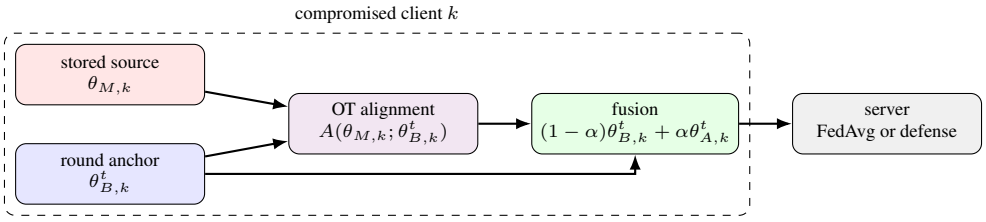

Figure 1: OTPA attack flow. Each compromised client aligns its stored adversarial source with a clean round anchor and fuses them before submission.

2 Related Work

Federated poisoning. Model replacement scales an update to move aggregation toward an attacker model (Bagdasaryan et al., 2020), while constrained attacks optimize against honest-update geometry or the aggregation rule (Baruch et al., 2019; Fang et al., 2020; Shejwalkar & Houmansadr, 2021). Other attacks couple malicious directions across rounds (Xie et al., 2025), concentrate changes in selected parameters or subnetworks (Zhang et al., 2022; Zhuang et al., 2024; Guo et al., 2026), or manipulate local sampling (Liu et al., 2024). OT-aligned surrogates have also been combined with adaptive replacement timing and budget (Kumar et al., 2025). OTPA constructs every submission from a persistent source and a current-round clean anchor, without honest updates.

Robust aggregation. Krum uses pairwise distances (Blanchard et al., 2017); coordinatewise median and trimmed mean use robust coordinates (Yin et al., 2018); Divide-and-Conquer (DnC) filters projected outliers (Shejwalkar & Houmansadr, 2021); robust federated aggregation (RFA) computes a geometric median (Pillutla et al., 2022); and FLTrust scores updates against a trusted direction (Cao et al., 2021). FoolsGold suppresses coordinated clients through history-based similarity (Fung et al., 2020), while FLAME combines clustering, clipping, and noise (Nguyen et al., 2022). These defenses use different statistics, so proximity alone does not predict defended accuracy.

Alignment and model fusion. OT model fusion aligns hidden-feature coordinates before averaging parameters. Singh & Jaggi (2020) developed this for multilayer perceptrons, convolutional networks, and residual networks, and Imfeld et al. (2024) extended it to vision transformers and BERT; entropic regularization enables soft couplings through Sinkhorn iterations (Cuturi, 2013), with Peyré & Cuturi (2019) giving broader background. Related benign methods include Bayesian matching, FedMA, and permutation-based weight matching (Yurochkin et al., 2019; Wang et al., 2020; Ainsworth et al., 2023), while OTBR (Lin et al., 2025), BadMerging and Defense-Aware Merging (Zhang et al., 2024; Yang et al., 2025) address offline repair and merge security. OTPA instead operates inside a malicious FL client, refreshes its clean anchor at participation, and compares directly with unaligned interpolation (Table 6).

3 SETTING AND THREAT MODEL

At round t, the server broadcasts w^t and samples participants $\mathcal{S}_t$. Client k returns a model w_k^{t+1}, or equivalently $\Delta_k^t = w_k^{t+1} - w^t$. FedAvg forms

$$w^{t+1} = w^t + \sum_{k \in \mathcal{S}_t} p_k^t \Delta_k^t, \qquad p_k^t \geq 0, \quad \sum_{k \in \mathcal{S}_t} p_k^t = 1. \tag{2}$$

The experiments use equal client weights, including under heterogeneous partitions where local sample counts may differ.

Table 1: Threat model and attacker assumptions.

Aspect	Assumption
Goal and type	Untargeted model poisoning that reduces evaluation accuracy, without explicit aggregation-aware amplification.
Compromise	A fixed malicious subset: 25/100 clients on CIFAR, 10/50 on Tiny-ImageNet, and 5/20 on SST-2.
Attacker access	The architecture, broadcast model, and own data; persistent local state; source and anchor training; modification of compromised submissions.
Attacker exclusions	No honest-client data or updates, private defense thresholds, or other clients' state; no control over selection, honest training, aggregation, or server state.
Server	Synchronous FL with individual client submissions available to FedAvg or the evaluated server defense.

Before FL, each compromised client stores a noisy-label source M_k and, when selected, trains a clean anchor B_k^t from the current broadcast. Each source is trained from its own independent random initialization, distinct from the global model's, so a nontrivial feature correspondence exists between source and anchor. We evaluate impact by vision test accuracy or SST-2 validation accuracy and report each malicious update's distance from the mean and coordinatewise median of all submissions.

4 OTPA

4.1 FROM A BENIGN ANCHOR TO AN ALIGNED SUBMISSION

For compromised client k, let $\theta_{M,k}$ denote its stored poisoned source and let $\theta_{B,k}^t = \mathrm{LocalTrain}(w^t, D_k)$ denote the clean anchor trained from the current broadcast. OTPA aligns compatible feature axes of the source with those of the anchor, obtaining $\theta_{A,k}^t = A(\theta_{M,k}; \theta_{B,k}^t)$, and returns

$$\theta_{F,k}^t = (1-\alpha)\theta_{B,k}^t + \alpha\theta_{A,k}^t, \qquad \alpha \in [0,1]. \tag{3}$$

The round-specific anchor ties the submission to the current global state, while the stored source carries the adversarial objective across rounds. The fusion weight α is fixed before each run and chosen without access to a private server-side detector. The matched Linear baseline instead returns $\theta_{L,k}^t = (1-\alpha)\theta_{B,k}^t + \alpha\theta_{M,k}$. Consequently, $\theta_{F,k}^t - \theta_{B,k}^t = \alpha(\theta_{A,k}^t - \theta_{B,k}^t)$ and $\theta_{L,k}^t - \theta_{B,k}^t = \alpha(\theta_{M,k} - \theta_{B,k}^t)$. The source, anchor, fusion weight, and schedule are matched, so the comparison isolates the effect of applying the transport operator.

Why align before fusion? Independently trained networks can represent the same computation with different hidden-unit orderings. For two consecutive layers $h = \sigma(W_1 x)$ and $f(x) = W_2 h$, any permutation matrix P gives

$$f(x) = W_2 P^\top \sigma(P W_1 x), \tag{4}$$

because elementwise activations commute with permutation. This exact symmetry motivates correspondence search, but OTPA's dense maps also mix and contract features. Applying them along adjacent axes is therefore a soft construction, not a guarantee of functional equivalence. The matched Linear comparison isolates the complete transport step, not correspondence independently of contraction.

4.2 Entropic transport maps and dense layers

For a dense component with incoming map T_{in}, OTPA first places the source input axis in the anchor basis and then constructs output-feature descriptors:

$$W'_M = W_M T_{\text{in}}, \qquad x_i = \text{row}_i(W'_M), \qquad y_j = \text{row}_j(W_B). \tag{5}$$

For a convolution, the corresponding input-channel contraction is applied before each filter is flattened across its channel and spatial coordinates; an embedding table instead uses hidden-coordinate columns as descriptors. Source and anchor share an architecture, so each paired feature axis has cardinality n. Let $X_M = (x_1, \ldots, x_n)$ and $X_B = (y_1, \ldots, y_n)$; define $\bar{x}_i = x_i/(\|x_i\|_2 + \varepsilon)$ and $\bar{y}_j = y_j/(\|y_j\|_2 + \varepsilon)$. The squared Euclidean cost is rescaled to $[0, 1]$:

$$C_{ij} = \|\bar{x}_i - \bar{y}_j\|_2^2, \qquad \widetilde{C} = C/(\max_{ij} C_{ij} + \varepsilon), \qquad \varepsilon = 10^{-8}. \tag{6}$$

With uniform marginals $a = b = \frac{1}{n}\mathbf{1}$ and $\lambda > 0$, OTPA solves

$$\Pi_\lambda(\widetilde{C}) = \underset{\Pi\mathbf{1}=a,\ \Pi^\top\mathbf{1}=b,\ \Pi\geq 0}{\arg\min} \left\{ \langle \Pi, \widetilde{C}\rangle + \lambda \sum_{ij} \Pi_{ij}(\log \Pi_{ij} - 1) \right\}. \tag{7}$$

Here Π_{ij} is the mass assigned from source feature i to anchor feature j. Thus OT produces a globally coupled soft correspondence rather than an independent nearest-neighbor choice for every feature. We compute Π_λ with the Python Optimal Transport (POT) Sinkhorn solver, using at most 1000 iterations for vision and 5000 for the language experiment, with $\lambda = 0.06$ unless stated otherwise. Scaling the coupling gives the soft alignment map $T = n\Pi_\lambda$. Sinkhorn produces a numerically doubly stochastic map whose row and column sums equal one up to solver tolerance.

The map convention is the same throughout the paper: T_{ij} transports source coordinate i toward anchor coordinate j, so a source feature vector q_M is written in anchor coordinates as $\widetilde{q}_M = T^\top q_M$. For an affine layer $W_M \in \mathbb{R}^{n_{\text{out}} \times n_{\text{in}}}$, separate maps act on its output and input axes. We define the two-sided alignment operator

$$\mathcal{L}_{T_{\text{out}}, T_{\text{in}}}(W_M, b_M) := \left(T_{\text{out}}^\top W_M T_{\text{in}}, T_{\text{out}}^\top b_M\right) = (\widetilde{W}_M, \widetilde{b}_M). \tag{8}$$

For a hard permutation, Eq. (8) exactly re-expresses the same affine map in the anchor basis. Entropic OT replaces the permutation with a dense doubly stochastic map, which does not preserve the represented function but places the source in the anchor's coordinate system. The outgoing map of one component becomes the incoming map of the next. Thus an MLP with $W_{1,M} \in \mathbb{R}^{m \times d}$ and $W_{2,M} \in \mathbb{R}^{d_{\text{out}} \times m}$ follows

$$\begin{aligned} (\widetilde{W}_1, \widetilde{b}_1) &= \mathcal{L}_{T_h, T_x}(W_{1,M}, b_{1,M}), \\ (\widetilde{W}_2, \widetilde{b}_2) &= \mathcal{L}_{T_y, T_h}(W_{2,M}, b_{2,M}), \end{aligned} \tag{9}$$

where T_x, T_h, T_y match the input, hidden, and output widths. This map chain also covers nonsquare layers.

4.3 Alignment across neural architectures

OTPA aligns parameter feature axes, not examples, pixels, tokens, or labels. Each linear, convolutional, embedding, or pooler component that defines a new feature basis solves an output map with Eq. (7). Normalization and semantic tensors inherit a compatible map, residual joins reconcile branch maps, and classifier-label axes remain fixed. Below, a tilde denotes an aligned source tensor.

Convolutional and residual networks. For a convolution, OTPA first applies T_{in} to the input-channel axis, then flattens each resulting filter across channels and spatial coordinates to form the descriptors that determine T_{out}. For a source kernel $K_M \in \mathbb{R}^{c_{\text{out}} \times c_{\text{in}} \times h \times w}$, the aligned kernel and bias use $T_{\text{in}} \in \mathbb{R}^{c_{\text{in}} \times c_{\text{in}}}$ and $T_{\text{out}} \in \mathbb{R}^{c_{\text{out}} \times c_{\text{out}}}$:

$$[\widetilde{K}_M]_{ajuv} = \sum_{o=1}^{c_{\text{out}}} \sum_{i=1}^{c_{\text{in}}} [T_{\text{out}}]_{oa} [K_M]_{oiuv} [T_{\text{in}}]_{ij}, \qquad \widetilde{b}_M = T_{\text{out}}^\top b_M. \tag{10}$$

The first convolution uses $T_{\text{in}} = I_3$ on the RGB channel axis; spatial indices u, v remain fixed. A learnable normalization vector $z \in \{\gamma, \beta\}$ on the output stream follows the same channel map, $\widetilde{z}_M = T_{\text{out}}^\top z_M$. For a residual block, OTPA propagates the main and shortcut paths according to

$$T_{\text{skip}} = \begin{cases} T_{\text{in}}, & \text{same-width identity shortcut,} \\ T_{\text{proj}}, & \text{projection shortcut,} \end{cases} \qquad T_{\text{join}} = \tfrac{1}{2}(T_{\text{main}} + T_{\text{skip}}). \tag{11}$$

Here T_{main} is the final main-path map and T_{proj} is obtained from the shortcut convolution using Eq. (10). Their average is doubly stochastic and becomes the input map of the next block. At the final classifier, the semantic class axis is fixed, giving $\widetilde{W}_C = W_{C,M} T_{\text{final}}$ and $\widetilde{b}_C = b_{C,M}$. The bias is not transported because its coordinates are class labels; it is still interpolated with the anchor by Eq. (3).

Vision transformers. Let d, m, s, N, L denote hidden width, MLP width, patch size, patch count, and encoder depth. Flattening the patch kernel gives $W_{\text{patch},M} \in \mathbb{R}^{d \times 3s^2}$; its input axis is the fixed RGB-pixel basis, while its output map $T_e \in \mathbb{R}^{d \times d}$ also aligns the class token $c_M \in \mathbb{R}^d$. A separate map $T_p \in \mathbb{R}^{d \times d}$ aligns the hidden columns of $E_{\text{pos},M} \in \mathbb{R}^{(N+1) \times d}$. Thus the input to the first block is

$$\begin{aligned} \widetilde{W}_{\text{patch}} &= T_e^\top W_{\text{patch},M}, \quad \widetilde{b}_{\text{patch}} = T_e^\top b_{\text{patch},M}, \quad \widetilde{c} = T_e^\top c_M, \\ \widetilde{E}_{\text{pos}} &= E_{\text{pos},M} T_p, \qquad\qquad T_0 = \tfrac{1}{2}(T_e + T_p). \end{aligned} \tag{12}$$

Within block ℓ, let T_ℓ denote the incoming residual-stream map, $W_Q, W_K, W_V, W_O \in \mathbb{R}^{d \times d}$ the combined-head projections, $W_1 \in \mathbb{R}^{m \times d}$, and $W_2 \in \mathbb{R}^{d \times m}$. The maps $T_\ell, T_Q, T_K, T_V, T_O, T_2 \in \mathbb{R}^{d \times d}$ and $T_1 \in \mathbb{R}^{m \times m}$ give

$$\begin{aligned} (\widetilde{W}_r, \widetilde{b}_r) &= \mathcal{L}_{T_r, T_\ell}(W_{r,M}, b_{r,M}), && r \in \{Q, K, V\}, \\ (\widetilde{W}_O, \widetilde{b}_O) &= \mathcal{L}_{T_O, T_V}(W_{O,M}, b_{O,M}), && T_\ell^{\text{att}} = \tfrac{1}{2}(T_\ell + T_O), \\ (\widetilde{W}_1, \widetilde{b}_1) &= \mathcal{L}_{T_1, T_\ell^{\text{att}}}(W_{1,M}, b_{1,M}), && (\widetilde{W}_2, \widetilde{b}_2) = \mathcal{L}_{T_2, T_1}(W_{2,M}, b_{2,M}), \\ T_{\ell+1} &= \tfrac{1}{2}(T_\ell^{\text{att}} + T_2). \end{aligned} \tag{13}$$

For $z \in \{\gamma, \beta\}$, the two pre-LayerNorm pairs obey $\widetilde{z}_{\ell,1} = T_\ell^\top z_{\ell,1,M}$ and $\widetilde{z}_{\ell,2} = (T_\ell^{\text{att}})^\top z_{\ell,2,M}$. Query and key maps terminate at the attention product, while the value map feeds the input axis of W_O; the output and MLP maps then merge with their residual streams. For $Q_h, K_h \in \mathbb{R}^{n_{\text{tok}} \times d_h}$, applying the same head-respecting permutation $P_h \in \mathbb{R}^{d_h \times d_h}$ gives the exact identity

$$(Q_h P_h)(K_h P_h)^\top = Q_h P_h P_h^\top K_h^\top = Q_h K_h^\top \tag{14}$$

OTPA instead optimizes dense query and key maps separately over the concatenated hidden dimension; these maps generally change the attention logits in Eq. (14). The final head fixes the semantic class axis, so $\widetilde{W}_C = W_{C,M} T_L$ and $\widetilde{b}_C = b_{C,M}$; its bias is then interpolated directly by Eq. (3).

BERT-style sequence classifiers. For $s \in \{\text{word}, \text{position}, \text{type}\}$, let $E_{s,M} \in \mathbb{R}^{N_s \times d}$ and $T_s \in \mathbb{R}^{d \times d}$. OTPA keeps the semantic row indices fixed, sets $\widetilde{E}_s = E_{s,M} T_s$, and initializes the encoder stream with $T_0 = (T_{\text{word}} + T_{\text{position}} + T_{\text{type}})/3$. The attention and MLP weights then follow Eq. (13); because the encoder uses post-residual LayerNorm, its attention-output and feedforward-output pairs receive T_ℓ^{att} and $T_{\ell+1}$, respectively; the embedding LayerNorm receives T_0. For pooler $W_{P,M} \in \mathbb{R}^{d_p \times d}$, map $T_P \in \mathbb{R}^{d_p \times d_p}$, and classifier $W_{C,M} \in \mathbb{R}^{C \times d_p}$,

$$(\widetilde{W}_P, \widetilde{b}_P) = \mathcal{L}_{T_P, T_L}(W_{P,M}, b_{P,M}), \qquad (\widetilde{W}_C, \widetilde{b}_C) = (W_{C,M} T_P, b_{C,M}). \tag{15}$$

Thus T_L enters the pooler, its solved output map T_P enters the classifier, and vocabulary, position, token-type, and class-label coordinates retain their semantics. The classifier bias is interpolated directly in Eq. (3) without a transport map. Appendix C gives the complete tensor routing.

Analytical properties. The fusion rule gives α direct control over the aligned source contribution: the fused model's deviation from its anchor is exactly α times the difference between the aligned source and anchor. Appendix D develops this property further, bounds OTPA's contribution to a FedAvg round, establishes stability of the entropic transport plan, and relates transport cost to barycentric descriptor error.

5 EXPERIMENTAL DESIGN

Benchmarks and models. The primary CIFAR-10 and CIFAR-100 experiments (Krizhevsky, 2009) use 100 clients, 25 compromised clients, a 3.20M-parameter Vision Transformer (ViT) (Dosovitskiy et al., 2021), and 100 full-participation rounds with independent and identically distributed (IID) data. Additional 100-round CIFAR-10 experiments study a three-layer CNN, ResNet-50, heterogeneous partitions, partial participation, parameter sensitivity, seven robust aggregation or outlier-filtering mechanisms, and two explicit client filters. Tiny-ImageNet uses 50 clients, 10 compromised clients, and a ViT-Base-style model over 30 rounds (Stanford CS231N, 2015); SST-2 uses 20 clients, five compromised clients, and a six-layer BERT-style encoder over 10 rounds (Wang et al., 2018; Devlin et al., 2019). In the CIFAR studies, conditions share the initialization, partition, participation schedule, and compromised clients, and source-based conditions also share their source-training configuration. A matched compromise-rate study uses 1, 5, 10, 15, 20, and 25 compromised clients, and partial-participation studies sample 10, 20, or 50 of 100 clients uniformly without replacement each round without forcing malicious clients to participate. Here p denotes the probability that an accessed source-training label is replaced by a newly sampled incorrect class; the primary settings use $p = 1.0$ for CIFAR-10 and $p = 0.8$ for CIFAR-100 and Tiny-ImageNet.

Comparators and measurements. The primary benchmark includes clean FL, noisy-label training, norm-matched noise, Min-Max and Min-Sum (Shejwalkar & Houmansadr, 2021), sign flip, PoisonedFL (Xie et al., 2025), BadSampler (Liu et al., 2024), direct source submission, unaligned interpolation, and OTPA; both CIFAR datasets and Tiny-ImageNet use this eleven-condition comparison, while the CNN and ResNet-50 transfer study isolates Clean, Direct, Linear, and OTPA. We report each malicious update's Euclidean distance from the mean and coordinatewise median of all submitted updates as D_{mean} and D_{med}, with the clean control averaging the same quantities over clean clients. Each defense comparison matches its FedAvg attack run in initialization, partition, and schedule. Appendix E gives the full optimization, corruption, architecture, metric, and defense settings.

6 RESULTS

We report attack impact through test accuracy and update geometry through D_{mean} and D_{med}; lower values indicate stronger untargeted impact and closer placement, respectively. We compare the primary attacks within their information classes, then examine transfer, defended accuracy, and construction choices. Distance describes proximity, not a detector's ability to identify a malicious client.

6.1 TRANSPORT PRESERVES IMPACT WHILE REDUCING MEDIAN DISTANCE

Across the ten attack conditions in Table 2 and Figure 2, OTPA is Pareto-optimal among attacks that do not observe current-round honest updates: on none of the three datasets does such an attack attain both lower accuracy and lower D_{med}. It strictly dominates noisy-label training, sign flip, and PoisonedFL on all three datasets, and BadSampler on CIFAR-10 and Linear on CIFAR-100 as well. On CIFAR-10 and Tiny-ImageNet no attack attains both, including those that do observe honest updates. BadSampler is the only other attacker-local attack whose median distance is comparably small, and OTPA lowers accuracy by 18.4–43.6 additional points at comparable or smaller distance.

The controlled source-based comparison isolates the transport step, since Linear shares OTPA's source, anchor, fusion weight, and schedule while omitting transport and Direct provides the unscaled-source reference. OTPA stays within 2.5 accuracy points of Linear and 6.4 of Direct while reducing D_{med} by 70%–90% relative to matched Linear across the three datasets.

Min-Max, Min-Sum, and norm-matched noise receive current-round honest updates as oracle information, so their results provide context beyond OTPA's information class rather than a matched-knowledge comparison. Against PoisonedFL, which is attacker-local, OTPA lowers Tiny-ImageNet accuracy by 25.92 points and median distance by 77.6%.

Table 2: Primary vision results after 100 rounds on CIFAR and 30 rounds on Tiny-ImageNet. Reported uncertainties are two-sided 95% Student t confidence intervals. Oracle-access attacks (†) use current-round honest updates; all other attacks use attacker-local information. Lower accuracy indicates greater attack impact, and lower distance indicates closer placement. *Blue bold values mark the favorable impact or distance side of a highlighted attack tradeoff; red values mark its opposing extreme. Highlighting covers the full reported entry. Colors do not denote a global ranking.*

	CIFAR-10 ($p = 1.0$)			**CIFAR-100** ($p = 0.8$)			**Tiny-ImageNet** ($p = 0.8$)		
Method	Acc. ↓	D_{mean} ↓	D_{med} ↓	Acc. ↓	D_{mean} ↓	D_{med} ↓	Acc. ↓	D_{mean} ↓	D_{med} ↓
Clean	66.69 ± 0.87	8.37 ± 0.03	8.39 ± 0.03	37.45 ± 1.81	8.43 ± 0.02	8.45 ± 0.02	66.02 ± 0.03	38.56 ± 0.01	39.03 ± 0.02
Noisy-label training	58.85 ± 4.43	8.87 ± 0.33	8.95 ± 0.25	35.25 ± 1.55	8.94 ± 0.13	9.01 ± 0.25	59.86 ± 0.04	42.29 ± 0.02	42.71 ± 0.01
Norm-matched noise	64.79 ± 0.35	8.26 ± 0.02	8.29 ± 0.02	34.45 ± 1.04	8.33 ± 0.01	8.35 ± 0.01	56.63 ± 0.01	39.13 ± 0.01	39.54 ± 0.03
Min-Max	41.98 ± 1.18	**5.32 ± 0.20**	**5.15 ± 0.27**	14.09 ± 1.11	**6.00 ± 0.12**	**5.75 ± 0.14**	59.44 ± 0.03	**12.25 ± 0.02**	**7.73 ± 0.04**
Min-Sum	44.06 ± 0.87	**4.87 ± 0.17**	**4.51 ± 0.21**	15.13 ± 1.06	**5.75 ± 0.07**	**5.39 ± 0.06**	61.14 ± 0.05	**12.31 ± 0.03**	**7.86 ± 0.02**
Sign flip	57.96 ± 1.08	8.94 ± 0.03	8.98 ± 0.03	25.13 ± 0.50	9.27 ± 0.02	9.30 ± 0.03	44.69 ± 0.04	43.20 ± 0.02	43.62 ± 0.03
PoisonedFL	44.42 ± 0.02	9.16 ± 0.03	11.22 ± 0.03	20.67 ± 0.02	7.86 ± 0.01	7.82 ± 0.03	40.82 ± 0.02	93.34 ± 0.05	107.38 ± 0.04
BadSampler	63.05 ± 1.96	6.49 ± 0.04	6.48 ± 0.04	34.13 ± 1.26	6.57 ± 0.02	6.58 ± 0.02	58.47 ± 0.03	23.18 ± 0.01	23.56 ± 0.04
Direct	**35.81 ± 2.02**	60.93 ± 0.02	61.06 ± 0.02	**15.33 ± 1.68**	61.64 ± 0.07	61.86 ± 0.07	**8.53 ± 0.02**	85.12 ± 0.04	87.64 ± 0.03
Linear	37.39 ± 1.32	54.86 ± 0.02	55.00 ± 0.01	17.43 ± 1.69	55.52 ± 0.06	55.74 ± 0.05	12.40 ± 0.05	77.48 ± 0.03	80.03 ± 0.01
OTPA	**39.41 ± 0.43**	**5.26 ± 0.10**	**5.47 ± 0.14**	**15.78 ± 1.31**	**6.22 ± 0.11**	**6.64 ± 0.15**	**14.90 ± 0.03**	**22.79 ± 0.05**	**24.06 ± 0.02**

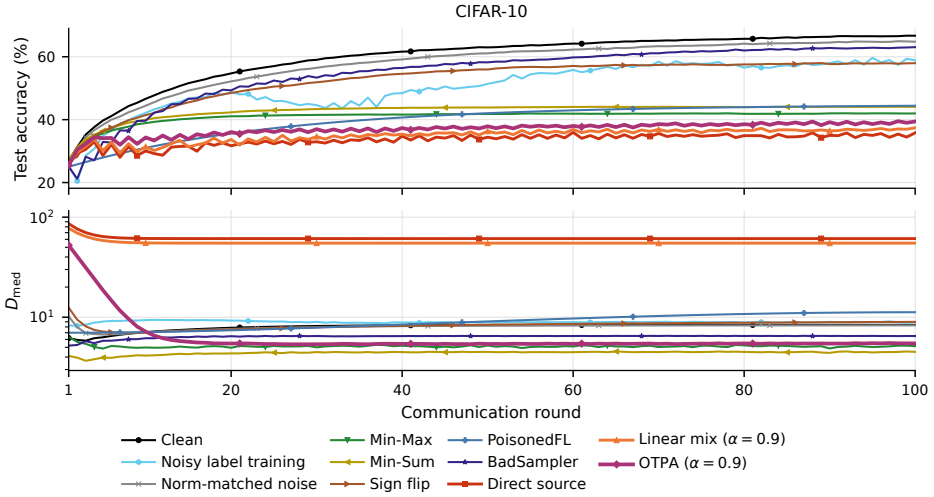

Figure 2: CIFAR-10 test accuracy (top) and median update distance (bottom) over 100 rounds for clean training and all ten attack conditions. D_{med} is displayed on a logarithmic axis.

Table 3: Round-100 transfer across data regimes and architectures.

(a) Data and participation

Setting	Method	Acc. (%)	D_{med}
IID/full	Clean	66.69 ± 0.87	8.39 ± 0.03
	Linear	37.39 ± 1.32	55.00 ± 0.01
	OTPA	39.41 ± 0.43	5.47 ± 0.14
Dir.(0.1)	Clean	59.31 ± 2.45	7.90 ± 0.33
	Linear	27.25 ± 6.66	54.98 ± 0.23
	OTPA	27.87 ± 8.23	4.73 ± 0.32
IID/20%	Clean	65.69 ± 1.18	8.44 ± 0.03
	Linear	32.07 ± 7.65	55.26 ± 2.31
	OTPA	36.54 ± 2.62	5.18 ± 0.75

(b) Architecture transfer

Model	Method	Acc. (%)	D_{med}
CNN	Clean	60.08 ± 0.99	0.98 ± 0.04
	Direct	28.01 ± 0.64	11.27 ± 0.19
	Linear	28.96 ± 0.91	10.28 ± 0.10
	OTPA	30.48 ± 3.48	6.79 ± 0.49
ResNet-50	Clean	65.28 ± 2.22	17.61 ± 0.47
	Direct	27.56 ± 9.97	215.41 ± 0.62
	Linear	30.54 ± 6.82	194.06 ± 0.55
	OTPA	24.53 ± 2.87	21.54 ± 2.50

6.2 Transfer across data regimes and architectures

Under Dirichlet heterogeneity and 20% participation, OTPA remains within 0.7 and 4.5 accuracy points of Linear while reducing D_{med} by over 90% (Table 3). It has the smallest source-based distance on both the CNN and ResNet-50. On ResNet-50 it is also the most damaging source-based construction, 3.0 and 6.0 accuracy points below Direct and Linear respectively, while reducing

distance by approximately 89%–90% relative to them. The trade-off therefore holds across data heterogeneity, participation rate, and the CNN and ResNet architectures beyond the primary ViT.

6.3 Robust aggregation

We evaluate all ten attack conditions with Krum, Multi-Krum, trimmed mean, coordinatewise median, RFA, FLTrust, and DnC. Table 4 reports the resulting defended accuracy for every attack and mechanism. OTPA leaves the lowest accuracy under the first six mechanisms, at 43.41% to 45.81% against 50.66% for the strongest baseline in any of those columns, whereas PoisonedFL is stronger under DnC at 54.92% against OTPA's 61.81%. The advantage is therefore defense-dependent rather than universal evasion.

Table 4: Absolute CIFAR-10 accuracy (%) after 100 rounds under FedAvg and seven defenses. Lower values indicate greater remaining attack impact; bold marks the lowest defended accuracy in each column. Norm-matched noise, Min-Max, and Min-Sum use current-round honest updates; all other attacks are attacker-local.

Attack	FedAvg	Krum	Multi-Krum	Trimmed mean	Median	RFA	FLTrust	DnC
Noisy-label training	58.85	71.45	76.65	77.45	67.55	75.05	79.55	74.15
Norm-matched noise	64.79	84.49	76.09	73.89	80.59	83.29	80.79	70.49
Min-Max	41.98	59.38	64.58	62.88	60.28	53.38	55.58	61.88
Min-Sum	44.06	66.26	59.56	60.26	67.16	50.66	52.96	61.76
Sign flip	57.96	72.96	78.06	71.76	78.56	78.86	64.36	70.86
PoisonedFL	44.42	54.72	53.22	55.92	55.42	53.42	55.62	**54.92**
BadSampler	63.05	70.95	76.75	69.75	76.45	76.85	81.45	71.15
Direct	35.81	62.71	60.71	61.31	61.21	61.51	58.81	62.71
Linear	37.39	61.99	64.69	60.59	65.19	60.69	62.89	66.69
OTPA	39.41	**45.01**	**45.71**	**43.61**	**45.81**	**43.51**	**43.41**	61.81

Explicit client filters expose a different signal. FoolsGold assigns every OTPA malicious client zero weight at round 100, with AUC 1.00 and no benign false positives; accuracy reaches 64.01% versus 65.88% in its clean control. FLAME also rejects every OTPA malicious client, reaching 58.67% versus 58.66% in its clean control, while rejecting 24 of 75 benign clients. Section 7 discusses this contrast with Euclidean proximity.

6.4 Dissecting the construction

Table 5: Matched CIFAR-10 alignment ablation after 100 rounds. Lower accuracy indicates stronger attack impact; lower distance indicates closer placement. All five conditions are single runs sharing one configuration and seed, so they are mutually comparable; they differ slightly from the seed-averaged entries of Table 2.

Construction	Acc. (%) ↓	D_{mean} ↓	D_{med} ↓
Linear (unaligned)	37.20	54.87	55.00
Hard assignment	44.26	40.59	40.91
Attention-restricted OT	37.05	13.55	13.70
MLP-restricted OT	35.65	22.13	22.29
Full OTPA	38.83	**5.22**	**5.44**

Restricting transport-map estimation to attention or MLP coordinates preserves attack impact but raises D_{med} from 5.44 for full OTPA to 13.70 and 22.29, respectively. Unaligned fusion reaches 37.20% accuracy at $D_{\text{med}} = 55.00$, whereas full OTPA reaches 38.83% at 5.44, a 90.1% distance reduction. The complete soft map-flow construction also improves both metrics over the tested hard-assignment variant.

6.5 Sensitivity to fusion and transport parameters

Increasing α strengthens both attacks, but Linear moves progressively farther from the update median while OTPA remains close. Across the tested λ values, OTPA simultaneously increases impact and reduces D_{med}, so the advantage is not tied to one setting: raising α from 0.1 to 0.9 moves Linear from $D_{\text{med}} = 9.23$ to 55.00 while OTPA stays between 6.89 and 5.47, and raising λ from 0.06 to 0.30 lowers OTPA's accuracy from 39.41% to 33.55% while D_{med} falls from 5.47 to 4.33.

6.6 TRANSFER TO A LANGUAGE TRANSFORMER

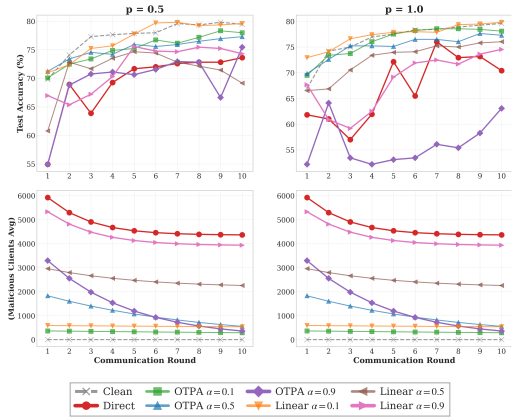

Figure 3: SST-2 accuracy and D_{mean} over 10 rounds at two corruption levels.

At both corruption levels, aligned fusion traces a lower-distance trajectory than matched Linear. The separation is strongest at full corruption, where larger α increases OTPA's impact while retaining substantially smaller D_{mean} than the unaligned controls. This behavior transfers the same alignment principle from vision transformers to BERT-style tensor routing.

7 DISCUSSION AND LIMITATIONS

OTPA separates Euclidean proximity from malicious functional influence, not from every form of detectability. Under FoolsGold, malicious classifier-head histories have mean pairwise cosine similarity 0.997, compared with -0.039 for Direct and -0.038 for Linear. This is consistent with transport reducing coordinate diversity while creating a shared malicious direction that temporal aggregation can identify. These histories come from defended runs, whereas the low-distance result comes from FedAvg; the comparison does not establish simultaneous low distance and high directional similarity on one trajectory. Together with FLAME's rejection and DnC's recovery, it shows why evaluating several geometric statistics matters. The matched Linear control isolates the transport step but does not separate learned correspondence from contraction and mixing. Our scope is untargeted, synchronous FL with persistent state and matched architectures; targeted and asynchronous settings, state resets, and larger language models remain open. Dense OT requires quadratic plan memory, and individual-update inspection is incompatible with secure aggregation (Bonawitz et al., 2017).

8 CONCLUSION

OTPA constructs poisoned submissions by transporting a persistent source toward a clean, round-specific anchor before fusion, without honest updates or explicit amplification. In the primary three-dataset benchmark, it is Pareto-optimal among the evaluated attacker-local attacks, reducing the median distance by 70% to 90% relative to matched Linear with at most 2.5 accuracy points of difference, and architecture and data-regime studies extend this beyond the benchmark. OTPA leaves the lowest accuracy under six robust aggregators, yet its consistent malicious histories are exposed by FoolsGold, while FLAME and DnC also counter its impact. The conclusion is therefore about which geometry reveals an attack, and it motivates evaluating functional impact alongside distance, temporal consistency, and defended accuracy.

AI USE STATEMENT

Generative AI tools assisted with manuscript editing and help in code implementation; the authors reviewed and take responsibility for all content.

ETHICS STATEMENT

The poisoning technique studied here has both offensive and defensive implications. Publishing its construction could inform attacks on federated learning systems, but the resulting geometric and spectral behavior can also support defense evaluation. The experiments use public benchmarks in simulated FL and involve no human subjects, private user data, or production systems. Alignment attacks should be studied only in isolated, authorized environments, with robust aggregation and subspace monitoring considered alongside distance metrics.

REPRODUCIBILITY STATEMENT

Section 5 describes the experimental protocol and distinguishes the primary studies from the additional evaluations. Appendix E lists the model, attack, optimization, solver, metric, and implementation details. Appendix D contains the complete proofs. The supplementary artifact provides executable implementations for the core OTPA method and experiment drivers for CIFAR, Tiny-ImageNet, CNN, ResNet-50, SST-2, heterogeneous partitions, partial participation, robust aggregation, FoolsGold, and FLAME.

REFERENCES

Samuel K. Ainsworth, Jonathan Hayase, and Siddhartha Srinivasa. Git Re-Basin: Merging models modulo permutation symmetries. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=CQsmMYmlP5T.

Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 2938–2948. PMLR, 2020. URL https://proceedings.mlr.press/v108/bagdasaryan20a.html.

Gilad Baruch, Moran Baruch, and Yoav Goldberg. A little is enough: Circumventing defenses for distributed learning. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/ec1c59141046cd1866bbbcdfb6ae31d4-Abstract.html.

Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. Analyzing federated learning through an adversarial lens. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 634–643. PMLR, 2019. URL https://proceedings.mlr.press/v97/bhagoji19a.html.

Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. In *Advances in Neural Information Processing Systems*, volume 30, pp. 119–129, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/f4b9ec30ad9f68f89b29639786cb62ef-Abstract.html.

Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1175–1191. ACM, 2017. doi: 10.1145/3133956.3133982.

Xiaoyu Cao, Minghong Fang, Jia Liu, and Neil Zhenqiang Gong. FLTrust: Byzantine-robust federated learning via trust bootstrapping. In *Network and Distributed System Security Symposium*. Internet Society, 2021. doi: 10.14722/ndss.2021.24434.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, volume 26, pp. 2292–2300, 2013. URL https://proceedings.neurips.cc/paper/2013/hash/af21d0c97db2e27e13572cbf59eb343d-Abstract.html.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186. Association for Computational Linguistics, 2019. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423/.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. Local model poisoning attacks to byzantine-robust federated learning. In *29th USENIX Security Symposium*, pp. 1605–1622. USENIX Association, 2020. URL https://www.usenix.org/conference/usenixsecurity20/presentation/fang.

Clement Fung, Chris J. M. Yoon, and Ivan Beschastnikh. The limitations of federated learning in sybil settings. In *23rd International Symposium on Research in Attacks, Intrusions and Defenses*, pp. 301–316. USENIX Association, 2020. URL https://www.usenix.org/conference/raid2020/presentation/fung.

Hanxi Guo, Hao Wang, Tao Song, Tianhang Zheng, Yang Hua, Haibing Guan, and Xiangyu Zhang. Poisoning with a pill: Circumventing detection in federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 21432–21440, 2026. doi: 10.1609/aaai.v40i26.39290.

Moritz Imfeld, Jacopo Graldi, Marco Giordano, Thomas Hofmann, Sotiris Anagnostidis, and Sidak Pal Singh. Transformer fusion with optimal transport. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=LjeqMvQpen.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

K. Naveen Kumar, C. Krishna Mohan, and Linga Reddy Cenkeramaddi. Federated learning minimal model replacement attack using optimal transport: An attacker perspective. *IEEE Transactions on Information Forensics and Security*, 20:478–487, 2025. doi: 10.1109/TIFS.2024.3516555. URL https://doi.org/10.1109/TIFS.2024.3516555.

Weilin Lin, Li Liu, Jianze Li, and Hui Xiong. Fusing pruned and backdoored models: Optimal transport-based data-free backdoor mitigation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 26299–26307, 2025. doi: 10.1609/aaai.v39i25.34828. URL https://ojs.aaai.org/index.php/AAAI/article/view/34828.

Yi Liu, Cong Wang, and Xingliang Yuan. BadSampler: Harnessing the power of catastrophic forgetting to poison byzantine-robust federated learning. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1944–1955. ACM, 2024. doi: 10.1145/3637528.3671879.

H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pp. 1273–1282. PMLR, 2017. URL https://proceedings.mlr.press/v54/mcmahan17a.html.

Thien Duc Nguyen, Phillip Rieger, Huili Chen, Hossein Yalame, Helen Möllering, Hossein Fereidooni, Samuel Marchal, Markus Miettinen, Azalia Mirhoseini, Shaza Zeitouni, Farinaz Koushanfar, Ahmad-Reza Sadeghi, and Thomas Schneider. FLAME: Taming backdoors in federated learning. In *31st USENIX Security Symposium*, pp. 1415–1432. USENIX Association, 2022. URL https://www.usenix.org/conference/usenixsecurity22/presentation/nguyen.

Gabriel Peyré and Marco Cuturi. Computational optimal transport. *Foundations and Trends in Machine Learning*, 11(5–6):355–607, 2019. doi: 10.1561/2200000073.

Krishna Pillutla, Sham M. Kakade, and Zaid Harchaoui. Robust aggregation for federated learning. *IEEE Transactions on Signal Processing*, 70:1142–1154, 2022. doi: 10.1109/TSP.2022.3153135.

Virat Shejwalkar and Amir Houmansadr. Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning. In *Network and Distributed System Security Symposium*. Internet Society, 2021. doi: 10.14722/ndss.2021.24498.

Sidak Pal Singh and Martin Jaggi. Model fusion via optimal transport. In *Advances in Neural Information Processing Systems*, volume 33, pp. 22045–22055, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/fb2697869f56484404c8ceee2985b01d-Abstract.html.

Stanford CS231N. Tiny imagenet challenge. Project page, 2015. URL https://cs231n.stanford.edu/2015/project.html.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP*, pp. 353–355. Association for Computational Linguistics, 2018. doi: 10.18653/v1/W18-5446. URL https://aclanthology.org/W18-5446/.

Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. Federated learning with matched averaging. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=BkluqlSFDS.

Yueqi Xie, Minghong Fang, and Neil Zhenqiang Gong. Model poisoning attacks to federated learning via multi-round consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15454–15463, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Xie_Model_Poisoning_Attacks_to_Federated_Learning_via_Multi-Round_Consistency_CVPR_2025_paper.html.

Jinluan Yang, Anke Tang, Didi Zhu, Zhengyu Chen, Li Shen, and Fei Wu. Mitigating the backdoor effect for multi-task model merging via safety-aware subspace. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=dqMqAaw7Sq.

Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 5650–5659. PMLR, 2018. URL https://proceedings.mlr.press/v80/yin18a.html.

Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan Greenewald, Nghia Hoang, and Yasaman Khazaeni. Bayesian nonparametric federated learning of neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 7252–7261. PMLR, 2019. URL https://proceedings.mlr.press/v97/yurochkin19a.html.

Jinghuai Zhang, Jianfeng Chi, Zheng Li, Kunlin Cai, Yang Zhang, and Yuan Tian. BadMerging: Backdoor attacks against model merging. In *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security*, pp. 4450–4464. ACM, 2024. doi: 10.1145/3658644.3690284.

Zhengming Zhang, Ashwinee Panda, Linyue Song, Yaoqing Yang, Michael W. Mahoney, Prateek Mittal, Ramchandran Kannan, and Joseph E. Gonzalez. Neurotoxin: Durable backdoors in federated learning. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 26429–26446. PMLR, 2022. URL https://proceedings.mlr.press/v162/zhang22w.html.

Haomin Zhuang, Mingxian Yu, Hao Wang, Yang Hua, Jian Li, and Xu Yuan. Backdoor federated learning by poisoning backdoor-critical layers. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=AJBGSVSTT2.

Appendix guide. Appendix C gives the complete client procedure and architecture-specific map routing. Appendix D develops the theoretical analysis and provides all proofs. Appendix E records the reported experimental configuration, comparator implementations, metrics, and defense settings. Appendix F reports the absolute operating points underlying the defense comparison in the main paper and the explicit client-filter results. Appendix B reports the compromise-rate endpoints and construction profile.

A Positioning Against Alignment and Merging Methods

Table 6: Representative alignment and model-merging methods.

Work	Protocol / role	Objects combined	Construction	Purpose
OT fusion (Singh & Jaggi, 2020; Imfeld et al., 2024)	Offline / benign	Independently trained neural models	Align compatible axes with OT, then average	Model reuse
FedMA (Wang et al., 2020)	Iterative FL / benign	Client layer representations	Match and average each layer	FL utility
FL-MMR (Kumar et al., 2025)	Iterative FL / attack	Surrogate poisoned and benign models	OT alignment with adaptive replacement map, timing, and budget	Poisoning with a small budget
OTBR (Lin et al., 2025)	Offline / defense	Pruned and backdoored models	Model fusion with OT	Repair without data
BadMerging / DAM (Zhang et al., 2024; Yang et al., 2025)	Offline merge / attack, defense	Contributed task models	Backdoor designed for merging; constrained safety subspace	Merge security
OTPA (ours)	Iterative FL / attack	Stored source and a fresh clean anchor	OT alignment across compatible axes, then dense interpolation	Untargeted poisoning

B Additional Experimental Results

B.1 Sensitivity sweeps

Figure 4 plots the two sweeps summarized in Section 6.

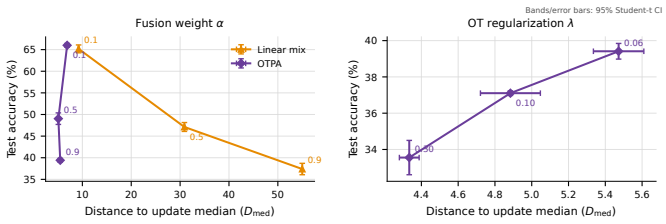

Figure 4: Round-100 CIFAR-10 sensitivity to fusion weight α (left) and OT regularization λ (right).

B.2 COMPROMISE-RATE SWEEP

Table 7: CIFAR-10 endpoints across the compromise-rate sweep. Methods use the same initialization, clients, source states, and training schedule at each compromise rate. Lower values indicate stronger impact and closer placement.

	Direct		Linear		OTPA	
Compromised	Acc.	D_{med}	Acc.	D_{med}	Acc.	D_{med}
1%	66.72	44.87	67.00	43.15	66.51	35.95
5%	61.83	57.73	63.15	52.16	62.30	14.51
10%	53.47	59.84	54.30	53.97	53.59	9.85
15%	45.93	60.55	47.85	54.55	48.60	7.77
20%	40.30	60.87	42.71	54.86	44.33	6.33
25%	35.81	61.06	37.39	55.00	39.41	5.47

OTPA has the smallest D_{med} at every evaluated compromise rate while remaining within 4.03 accuracy points of Direct and 2.02 points of Linear. Relative to the closer of these matched source-based controls, its distance reduction grows from 16.7% with one compromised client to 85.8% with fifteen, 88.5% with twenty, and 90.1% with twenty-five.

B.3 ADDITIONAL DATA-REGIME STRESS TESTS

Table 8: Round-100 CIFAR-10 endpoints under additional heterogeneity and participation settings. Lower accuracy indicates greater attack impact; lower distance indicates closer placement.

Setting	Method	Acc. (%) ↓	D_{med} ↓
Dirichlet (0.5), full participation	Clean	63.44	8.50
	Linear	35.51	55.04
	OTPA	34.82	**5.03**
Dirichlet (1.0), full participation	Clean	64.66	8.39
	Linear	36.80	55.02
	OTPA	37.26	**5.08**
IID, 10% participation	Clean	63.27	8.44
	Linear	32.90	55.10
	OTPA	35.27	**5.94**
IID, 50% participation	Clean	65.19	8.40
	Linear	36.93	54.87
	OTPA	39.65	**5.68**

OTPA preserves the impact of matched unaligned fusion while sharply reducing its update distance in every setting. Relative to Linear, D_{med} falls by 89.2%–90.9%, while the accuracy difference remains at most 2.72 percentage points.

B.4 CONSTRUCTION PROFILE

Table 9: Compact-ViT CPU post-training construction microbenchmark. Times are medians over five measured repetitions after two warmups. The benchmark uses prepared in-memory states and excludes source training, local training, aggregation, and file I/O. Map volume is cumulative materialized tensor size, not peak memory.

Construction	Malicious construction (ms)	Map tensors (MiB)
Linear interpolation	48.3	0.0
Attention-restricted OT	176.5	6.0
MLP-restricted OT	208.1	7.5
Full OTPA	337.1	14.0
Hard assignment	339.1	12.5

Full OTPA requires 337.1 ms and materializes 14.0 MiB of transport maps in this construction-only profile, placing its runtime between the restricted OT variants and hard assignment. These measurements do not include training, aggregation, file I/O, or peak accelerator memory.

C Alignment Across Architectures

Throughout this appendix, $\theta_{M,k}$ is client k's stored poisoned source, $\theta^t_{B,k}$ is its clean round-t anchor, $\theta^t_{A,k} = A(\theta_{M,k}; \theta^t_{B,k})$ is the aligned source, and $\theta^t_{F,k} = (1-\alpha)\theta^t_{B,k} + \alpha\theta^t_{A,k}$ is the submitted model. A transport entry T_{ij} maps source feature i into anchor coordinate j; an output axis is consequently transformed by $T^\top$ and an input axis by T. Semantic axes such as RGB channels, spatial kernel coordinates, token identities, sequence positions, and class labels remain fixed. Hard permutations that satisfy the architectural constraints, including shared maps at residual joins and head-consistent query/key maps, preserve the represented function. OTPA applies the same routing with dense entropic maps and averaged residual joins. These maps are doubly stochastic rather than permutations, so the routing is shared but the exactness is not; the construction is uniform across convolutional, residual, vision-transformer, and BERT-style architectures.

C.1 Client procedure

Algorithm 1 OTPA submission at compromised client k

Require: Global model w^t, local clean data D_k, stored source $\theta_{M,k}$, fusion weight α, regularizer λ

$\theta^t_{B,k} \leftarrow \mathrm{LocalTrain}(w^t, D_k)$
Initialize maps on fixed semantic input axes to the identity
for each aligned component in a traversal of the architecture graph **do**
  if the component defines a new feature basis **then**
    Transform its source input axis by the incoming map
    Form output descriptors and construct Eq. (6)
    Solve Eq. (7); set $T_{\mathrm{out}} = n_{\mathrm{out}}\Pi_\lambda$
    Align the output axis and propagate T_{out}
  else if the component joins residual branches **then**
    Reconcile compatible branch maps using the prescribed average
  else
    Inherit the compatible stream map or fixed semantic identity
  end if
end for
Assemble $\theta^t_{A,k}$ and fuse each homologous floating tensor once using Eq. (3)
return submitted model $\theta^t_{F,k}$

OTPA completes the architecture-wide alignment pass before the final interpolation. Consequently, every descriptor cost compares source and anchor features along the architecture-defined map flow, and each homologous tensor is fused exactly once.

Map propagation. For $W_M, W_B \in \mathbb{R}^{n_{\mathrm{out}} \times n_{\mathrm{in}}}$, OTPA first forms $W'_M = W_M T_{\mathrm{in}}$. It row-normalizes W'_M and W_B, solves the resulting $n_{\mathrm{out}} \times n_{\mathrm{out}}$ transport problem, and sets $\widetilde{W}_M = T_{\mathrm{out}}^\top W'_M$ and $\widetilde{b}_M = T_{\mathrm{out}}^\top b_M$. The output map is propagated to the next component or reconciled with a compatible map at a graph join. At an equal-width two-branch addition, OTPA uses $T_{\mathrm{join}} = (T_{\mathrm{main}} + T_{\mathrm{skip}})/2$; an identity shortcut carries T_{in}, whereas a projection shortcut carries its solved output map. Because the branch maps are doubly stochastic, their average preserves the same marginal constraints and yields one map for the joined stream. Map propagation follows matching feature widths throughout the architecture graph.

Computational cost. For n descriptors of width d, cost construction and K_{it} Sinkhorn iterations require $O(n^2 d + K_{\mathrm{it}} n^2)$ time and $O(n^2)$ transport-plan memory. For a dense layer with n_{in} inputs and n_{out} outputs, applying the input and output maps costs $O(n_{\mathrm{out}} n_{\mathrm{in}}^2 + n_{\mathrm{out}}^2 n_{\mathrm{in}})$.

Table 10: Transport-map routing for the supported neural components.

Component	Aligned source tensor	Map propagation and fixed axes
Linear / MLP	$T_{\mathrm{out}}^{\top} W_M T_{\mathrm{in}}$; $T_{\mathrm{out}}^{\top} b_M$	Propagate T_{out}; classifier label rows remain fixed.
Convolution	T_{out} on filters and bias, T_{in} on input channels	Propagate T_{out}; spatial kernel coordinates and RGB inputs remain fixed.
Normalization	$T^{\top} \gamma_M$ and $T^{\top} \beta_M$	Use the map of the associated feature stream; no semantic axis is reordered.
Embedding table	$E_M T$ along hidden columns	Propagate the hidden map; token, position, and type indices remain fixed.
Query / key / value	$T_r^{\top} W_{r,M} T_{\mathrm{in}}$ and $T_r^{\top} b_{r,M}$	Query and key maps terminate at attention; the value map enters the output projection.
Attention output	$T_O^{\top} W_{O,M} T_V$ and $T_O^{\top} b_{O,M}$	Reconcile T_O with the residual-stream map.
Residual addition	No parameter tensor	Average compatible main and shortcut maps and propagate T_{join}.
Pooler	$T_P^{\top} W_{P,M} T_L$ and $T_P^{\top} b_{P,M}$	Propagate T_P to the sequence-classification head.
Classifier	$W_{C,M} T_{\mathrm{in}}$; $b_{C,M}$ is not transported	Hidden input coordinates are aligned; class-label coordinates remain fixed.

Convolutional and residual networks. For a convolutional layer, each output filter is flattened across its input channels and spatial coordinates to form a descriptor. The incoming map acts on input channels and the newly solved outgoing map acts on filters and biases. The outgoing map then becomes the input channel map for the following layer. Pointwise activations, pooling, and global average pooling pass the channel map forward without introducing a new one. In a ResNet bottleneck, OTPA applies this rule to the convolutions and learnable normalization parameters on the main path and to any projection shortcut. Compatible branch maps are averaged at residual addition, after which the resulting map is propagated to the next block. The reconciled map provides one soft correspondence for the next block and is carried through normalization and pointwise nonlinearities as part of the architecture-wide alignment. The final map transforms the classifier input axis, while output class rows remain fixed.

Vision transformer inputs and residual paths. The patch projection defines T_e over the hidden dimension and this map also acts on the class token. A separately solved position-embedding map T_p acts only on hidden columns; the position indices remain fixed. Their average $T_0 = (T_e + T_p)/2$ initializes the encoder map. In each block, the incoming map transforms the LayerNorm parameters and the input axes of the query, key, and value projections. Separate soft output maps are solved for the three projections. The query and key maps terminate at the attention computation, the value map becomes the input map for the attention output projection, and its output map is reconciled with the residual stream. At each residual join, the branch maps are averaged. Because a convex combination of doubly stochastic maps remains doubly stochastic, this operation preserves the marginal constraints and provides a single soft map for the next block.

Attention symmetry. For one attention head, using the same within-head permutation P_h on query and key features preserves its logits:

$$(Q_h P_h)(K_h P_h)^{\top} = Q_h P_h P_h^{\top} K_h^{\top} = Q_h K_h^{\top}.$$

This exact shared-permutation symmetry motivates OTPA's dense query and key correspondences over the concatenated hidden dimension, extending correspondence search beyond hard one-to-one assignments. Section 6 demonstrates that the resulting alignment transfers to both vision and language transformers.

MLP and classifier. The map entering the block aligns the input axis of the first MLP matrix. An OT map then aligns the hidden MLP axis and is applied to the input axis of the second matrix. The resulting map is merged at the residual join. The final normalization and classifier input use the last hidden map, while the output class coordinates remain fixed.

Sequence classifier with a BERT-style encoder. The language experiment uses post-residual normalization within its encoder blocks. The word, position, and token-type embeddings are aligned separately along their hidden columns, and the three maps are averaged to initialize the encoder stream; vocabulary rows, position indices, and token-type identities remain fixed. Attention and MLP matrices use the same two-sided transport flow as above, while each normalization vector uses

the map at its own residual stream. The last encoder map T_L aligns the pooler's input, the pooler's solved map T_P aligns its output and becomes the classifier input map, and the two label coordinates remain fixed. Equation (15) gives the full pooler and classifier transformation.

D Theoretical Analysis and Proofs

The analysis characterizes three components of OTPA. Propositions D.1 and D.2 quantify the effect of affine fusion at the client and in one FedAvg round. Proposition D.3 establishes stability of the exact entropic plan to changes in the rescaled cost, and Lemma D.4 bounds barycentric error for the normalized descriptors. The experiments evaluate the resulting functional impact and update geometry.

D.1 Anchor results and the effect of one FedAvg round

Proposition D.1 (Exact anchor deviation)**.** *For a fixed client k and round t, let $\Delta^t_{F,k} = \theta^t_{F,k} - w^t$ and $\Delta^t_{B,k} = \theta^t_{B,k} - w^t$. Then*

$$\|\Delta^t_{F,k} - \Delta^t_{B,k}\|_2 = \alpha \|\theta^t_{A,k} - \theta^t_{B,k}\|_2. \tag{16}$$

Proof of Proposition D.1. Subtracting $\theta^t_{B,k}$ from Eq. (3) yields

$$\theta^t_{F,k} - \theta^t_{B,k} = \alpha(\theta^t_{A,k} - \theta^t_{B,k}).$$

The same identity holds for $\Delta^t_{F,k} - \Delta^t_{B,k}$ because the common w^t cancels. Taking the Euclidean norm and using $\alpha \geq 0$ proves the claim. □

Here and below, the parameter norm is the Euclidean norm after concatenating all floating parameter tensors. Proposition D.1 holds for every aligned source and isolates the linear control that α exerts over displacement from the anchor.

Proposition D.2 (FedAvg perturbation in one round)**.** *Let $\mathcal{M}$ be the fixed set of compromised clients and $\mathcal{M}_t = \mathcal{M} \cap \mathcal{S}_t$ the compromised participants in round t. Hold the broadcast w^t, participants, aggregation weights, and honest submissions fixed. Let w^{t+1}_{anchor} denote the counterfactual FedAvg output in which clients in $\mathcal{M}_t$ submit their clean anchors, and let w^{t+1}_{OTPA} denote the output in which they submit OTPA. Then*

$$\begin{aligned} w^{t+1}_{\text{anchor}} &= w^t + \sum_{k \in \mathcal{S}_t \setminus \mathcal{M}_t} p^t_k \Delta^t_k + \sum_{k \in \mathcal{M}_t} p^t_k (\theta^t_{B,k} - w^t), \\ w^{t+1}_{\text{OTPA}} &= w^t + \sum_{k \in \mathcal{S}_t \setminus \mathcal{M}_t} p^t_k \Delta^t_k + \sum_{k \in \mathcal{M}_t} p^t_k (\theta^t_{F,k} - w^t). \end{aligned} \tag{17}$$

Consequently,

$$\|w^{t+1}_{\text{OTPA}} - w^{t+1}_{\text{anchor}}\|_2 \leq \alpha \sum_{k \in \mathcal{M}_t} p^t_k \|\theta^t_{A,k} - \theta^t_{B,k}\|_2. \tag{18}$$

Proof of Proposition D.2. The honest terms cancel between the two counterfactual FedAvg outputs. For each malicious client, Eq. (3) gives

$$w^{t+1}_{\text{OTPA}} - w^{t+1}_{\text{anchor}} = \alpha \sum_{k \in \mathcal{M}_t} p^t_k (\theta^t_{A,k} - \theta^t_{B,k}).$$

The triangle inequality proves Eq. (18). □

Coupling the two outcomes at the same w^t isolates OTPA's contribution to one aggregation step while holding the broadcast, participants, weights, and honest submissions constant. The 100-round experiments measure the resulting behavior over complete training trajectories.

D.2 Sensitivity of exact entropic OT to cost perturbations

For a matrix, $\|\cdot\|_1$ below denotes the entrywise sum of absolute values and $\|\cdot\|_\infty$ denotes the maximum absolute entry; these are not induced operator norms.

Proposition D.3 (Sensitivity of the entropic transport plan)**.** *For $\lambda > 0$, let $\Pi_{\widetilde{C}}$ and $\Pi_{\widetilde{C}'}$ be exact minimizers of Eq. (7) when its cost is replaced by finite costs $\widetilde{C}, \widetilde{C}'$ with the same positive marginals, normalized to total mass one. Then*

$$\|\Pi_{\widetilde{C}} - \Pi_{\widetilde{C}'}\|_1 \leq \frac{\|\widetilde{C} - \widetilde{C}'\|_\infty}{\lambda}. \tag{19}$$

In the square, uniform setting used by OTPA, where $a = b = \mathbf{1}/n$ and $T = n\Pi$, this further implies

$$\|T_{\widetilde{C}} - T_{\widetilde{C}'}\|_1 \leq \frac{n\|\widetilde{C} - \widetilde{C}'\|_\infty}{\lambda}. \tag{20}$$

Proof. Write $P = \Pi_{\widetilde{C}}$ and $Q = \Pi_{\widetilde{C}'}$. Finite costs and positive marginals give positive optimal plans. The first-order optimality conditions on the common transport polytope are

$$\langle \widetilde{C} + \lambda \log P, Q - P\rangle \geq 0,$$
$$\langle \widetilde{C}' + \lambda \log Q, P - Q\rangle \geq 0.$$

Adding them and using the Bregman divergence of negative entropy, written as the Kullback–Leibler (KL) divergence, gives

$$\lambda \mathrm{KL}(Q\|P) + \lambda \mathrm{KL}(P\|Q) \leq \langle \widetilde{C} - \widetilde{C}', Q - P\rangle.$$

Viewing the matrices as unit-mass probability vectors and using natural logarithms, Pinsker's inequality in both directions bounds the left side below by $\lambda\|P - Q\|_1^2$. Hölder's inequality bounds the right side above by $\|\widetilde{C} - \widetilde{C}'\|_\infty \|P - Q\|_1$. Canceling the nonzero distance proves Eq. (19); the zero case is immediate. In the square, uniform case, $T = n\Pi$ gives Eq. (20). □

Proposition D.3 characterizes the entropic optimum under the same rescaled cost used by OTPA. Its explicit $1/\lambda$ dependence shows how entropic regularization controls the stability of the learned correspondence.

D.3 Barycentric descriptor error

Lemma D.4 (Jensen bound for normalized descriptors)**.** *Let $X_M, X_B \in \mathbb{R}^{n\times d}$ contain the normalized descriptors used in Eq. (6), with rows x_i and y_j. Let $C_{ij} = \|x_i - y_j\|_2^2$, define $s = \max_{ij} C_{ij} + \varepsilon$ and $\widetilde{C} = C/s$, and let Π be any feasible coupling with uniform marginals. For $T = n\Pi$,*

$$\|T^\top X_M - X_B\|_F^2 \leq n\langle \Pi, C\rangle = ns\langle \Pi, \widetilde{C}\rangle. \tag{21}$$

Proof. Each column of T sums to one. Therefore, for target row j, Jensen's inequality gives

$$\left\| \sum_i T_{ij} x_i - y_j \right\|_2^2 \leq \sum_i T_{ij} \|x_i - y_j\|_2^2.$$

Summing over j and substituting $T = n\Pi$ proves the result. □

Lemma D.4 bounds the normalized descriptors used to construct C; the factor s converts between the raw and rescaled costs. The empirical study separately evaluates raw-parameter geometry, functional impact, and defense behavior.

E REPRODUCIBILITY DETAILS

Table 11: Core experimental configuration.

Item	Value
Primary model	ViT, patch 4, width 256, 6 blocks, 8 heads, MLP 512
CIFAR-10 primary parameter count	3,195,146
Clients / malicious	100 / 25 (CIFAR); 50 / 10 (Tiny-ImageNet); 20 / 5 (SST-2)
Experiment horizon	100 rounds (CIFAR); 30 rounds (Tiny-ImageNet); 10 rounds (SST-2)
CIFAR primary local/source epochs	10 / 100
Tiny-ImageNet model	85.4M-parameter ViT-Base initialized from ImageNet-21k; patch 8, width 768, 12 blocks, 12 heads, MLP 3072; new 200-class head
Tiny-ImageNet local/source epochs	1 / 50
CNN model	Three 3×3 convolutions with 32/64/128 channels, ReLU, two pooling stages, global average pooling, and a linear head
CNN alignment	Separate transport map for each convolutional channel
ResNet model	ResNet-50 with a CIFAR stem and group normalization
ResNet alignment	Residual-aware transport through all bottlenecks and projection branches
Primary optimization	batch 128; AdamW, lr 10^{-3}, weight decay 5×10^{-5}
Primary schedule / clipping	cosine schedule restarted for each local or source training call; gradient clipping at norm 1
Primary vision partitions	IID, full participation
Extended partitions	IID and Dirichlet concentrations 0.1, 0.5, and 1.0
Extended participation	10%, 20%, 50%, and 100%
Extended optimization	batch 128; AdamW, lr 10^{-3}, weight decay 5×10^{-5}
Extended local/source epochs	10 / 100; cosine schedule; gradient clip 1
SST-2 model	6-layer BERT-style encoder trained end to end, width 768, 12 heads, MLP 3072; bert-base-uncased tokenizer
SST-2 protocol	20 / 5 clients, contiguous equal-size partitions, full participation; 10 rounds; local/source epochs 10 / 100; sequence length 128
SST-2 optimization	batch 32; AdamW, lr 2×10^{-5}, weight decay 10^{-2}
Source initialization	Each compromised client's source is trained from its own independent random initialization, drawn separately from the initialization of the federated global model
Source objective	fresh random label guaranteed to be incorrect on each access with probability p
Default attack	$\alpha = 0.9$, $\lambda = 0.06$; $p = 1.0$ for CIFAR-10, $p = 0.8$ for CIFAR-100/Tiny-ImageNet, and $p \in \{0.5, 1.0\}$ for SST-2
OT descriptors	Output-feature rows after applying the incoming map, normalized to unit ℓ_2 norm
OT cost	squared Euclidean, rescaled to $[0, 1]$
OT solver	POT Sinkhorn, at most 1000 vision iterations and 5000 language iterations; cost-normalization constant $\varepsilon = 10^{-8}$

Participation and filter metrics. Partial-participation studies sample 10, 20, or 50 clients uniformly without replacement and never force a malicious client to participate. The realized malicious counts average 2.44, 4.93, and 12.63 per round, with ranges 0–6, 1–11, and 7–17. The explicit-filter study reports final accuracy and client-level area under the receiver operating characteristic curve (AUC), true-positive rate (TPR), and false-positive rate (FPR).

Source lifecycle. For Direct, Linear, and OTPA, each compromised client trains and stores one source model before FL begins. In each participating round, Linear and OTPA train a clean anchor from the current broadcast; Linear interpolates the source directly, whereas OTPA first aligns it with the anchor. Direct submits the stored source without an anchor.

Comparator implementations. Direct submits the unscaled source, while Linear interpolates it with the clean anchor without alignment. Norm-matched noise, Min-Max, and Min-Sum use current-

round honest updates as oracle information. Min-Max and Min-Sum construct a shared malicious update along the negative unit-mean direction and choose its scale under their published squared-distance constraints. Noisy-label training, sign flip, BadSampler, and PoisonedFL do not use this oracle; PoisonedFL follows its published multi-round adaptive scaling.

Label corruption. For every untargeted source, a Bernoulli(p) corruption event replaces the label with a newly sampled class that is guaranteed to be incorrect. Labels are redrawn when an example is accessed rather than fixed once in advance. The source-based Direct, Linear, and OTPA conditions use the same objective and matched cached source states within each run.

Software and entry points. Experiments use Python 3.9.25, PyTorch 2.2.1 with CUDA 12.1, torchvision 0.17.1, NumPy 1.26.4, SciPy 1.12.0, POT 0.8.2, and Transformers 4.27.4. The supplementary artifact provides the core method implementation and experiment drivers.

Defense configurations. For aggregation rules that require an adversary count, $f = 25$. Krum scores each submission by its squared distances to the $n - f - 2$ nearest neighbors and selects the lowest-scoring one; Multi-Krum averages the $n - f$ lowest-scoring submissions. Coordinatewise trimmed mean removes f values from each tail, coordinatewise median takes the lower middle value, and RFA uses at most ten Weiszfeld iterations with tolerance 10^{-6}. FLTrust reserves 100 clean training examples for its server root update. Every rule is evaluated with a common initialization, partition, and schedule across attack conditions. Direct submits each compromised client's unscaled stored source. PoisonedFL starts from $c_0 = 8$, applies the released sign consistency test every 50 rounds, and multiplies c_t by 0.7 after a failed test subject to a minimum of 0.5. These settings are held fixed across aggregators. DnC independently samples at most 20,000 coordinates in each of five iterations, removes the 25 updates with the largest projected scores, and retains the intersection of the five kept sets. FoolsGold accumulates classifier-head updates by stable client identity, applies its cosine-similarity pardoning and logit weighting, and aggregates complete client deltas without renormalizing the retained weight. FLAME clusters pairwise cosine distances between complete submitted models with HDBSCAN (minimum cluster size 51 and minimum samples 1), keeps the strict-majority cluster, clips its updates at the median norm over all submissions, and adds Gaussian noise with standard deviation 0.001 times that median. We define anomaly score as one minus the FoolsGold weight, or one minus FLAME membership inside the retained cluster and one outside it, and use a fixed threshold of 0.5 for TPR and FPR. The filter matrix contains one matched 100-round run per condition using a common initialization, partition, participation schedule, and compromised-client set.

Geometry. Let $\mathrm{vec}(\Delta_i)$ concatenate all floating parameter tensors of client update i. For the n_t updates submitted in a round, define $\bar{\Delta} = n_t^{-1} \sum_i \Delta_i$ and $\widetilde{\Delta}_j = \text{lower-median}_i \, \Delta_{ij}$, matching PyTorch's convention when n_t is even. We report $D_{\mathrm{mean},i} = \|\Delta_i - \bar{\Delta}\|_2$ and $D_{\mathrm{med},i} = \|\Delta_i - \widetilde{\Delta}\|_2$. Both centers include honest and malicious submissions. A run-level value first averages client distances within the indicated role at that round; where configurations are replicated, tables average these run-level values across runs.

Alignment-ablation protocol. All ablation conditions fuse every floating parameter; they differ only in how feature maps are estimated and routed. Attention-restricted and MLP-restricted OT estimate new Sinkhorn maps only in the named component family and propagate the existing map through the other family. Full OTPA estimates patch, position, attention, and MLP maps using the architecture-wide routing in Appendix C. The hard-assignment condition replaces the soft plans with linear-assignment maps and uses a shared query/key map. Linear interpolation estimates no maps. The runtime profile uses same-architecture synthetic states, two warmups, five measured repetitions, and synchronized timing boundaries; it measures only in-memory post-training construction.

Sensitivity settings. The α sweep uses $\{0.1, 0.5, 0.9\}$ with $\lambda = 0.06$, whereas the λ sweep uses $\{0.06, 0.10, 0.30\}$ with $\alpha = 0.9$. All remaining settings follow the primary CIFAR-10 protocol.

E.1 CONTRACTION-MATCHED CONTROL

A doubly stochastic map contracts every source component not aligned with the uniform direction, so a lower distance could in principle reflect contraction rather than the estimated correspondence. We separate the two by replacing the solved map T with TP for a uniformly random permutation P. Permutations are orthogonal, so TP retains exactly the marginals, row entropies, and singular values of T, and only the source-to-anchor correspondence is destroyed.

Table 12: Contraction-matched control at the round-100 construction, averaged over eight compromised clients. $d(\theta_A, \theta_B)$ is the distance from the aligned source to its anchor; the untransported source sits at 62.85. Entropy is normalized so that 0 is a permutation and 1 the uniform map.

Transport map	$d(\theta_A, \theta_B) \downarrow$	Entropy	σ_2	$\|\theta_A\|/\|\theta_M\|$
Solved OT map (OTPA)	**8.23**	0.811	0.819	0.683
Correspondence-free (TP)	11.05	0.892	0.828	0.682
Row-shuffled descriptors	10.33	0.819	0.811	0.681
Uniform map	10.10	1.000	0.000	0.680

All four variants contract the source by the same factor, so contraction accounts for most of the reduction from 62.85. The solved map nevertheless places the aligned source closer than the correspondence-free control at matched entropy and matched spectrum, indicating that the estimated correspondence contributes beyond contraction alone. This is a construction-level measurement rather than a full training run.

F COMPLETE DEFENSE RESULTS

Table 4 reports absolute defended accuracy in the main text. Figure 5 shows the same comparison as accuracy recovery relative to each attack's matched FedAvg run, which makes the relative effect of each mechanism easier to read across attacks.

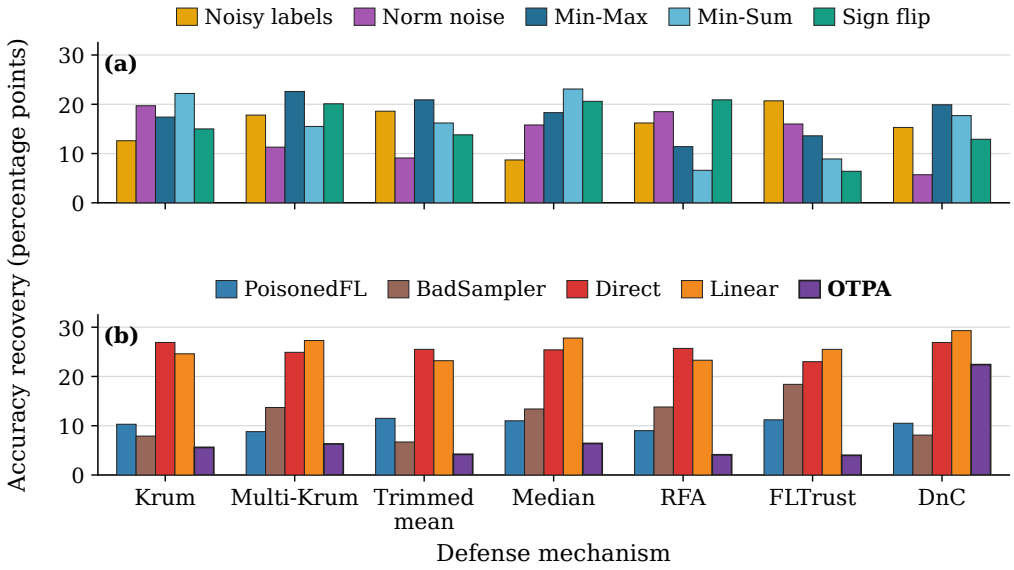

Figure 5: CIFAR-10 accuracy recovery for all ten attack conditions under seven defenses after 100 rounds, relative to each attack's matched FedAvg run. Lower values indicate fewer restored points.

F.1 EXPLICIT CLIENT FILTERING

Table 13: Round-100 CIFAR-10 client-filter results from one matched full-horizon run per condition. Acc. is defended test accuracy. AUC uses the final client anomaly scores; TPR and FPR use threshold 0.5. Higher AUC and TPR and lower FPR indicate better detection. Dashes denote metrics undefined for the clean control because it contains no malicious clients.

(a) FoolsGold

Attack	Acc.	AUC	TPR	FPR
Clean	65.88	–	–	0.00
Noisy-label training	64.17	1.00	1.00	0.00
Norm-matched noise	24.77	0.00	0.00	1.00
Min-Max	64.01	1.00	1.00	0.00
Min-Sum	64.01	1.00	1.00	0.00
Sign flip	57.10	0.56	0.00	0.00
PoisonedFL	64.35	1.00	1.00	0.00
BadSampler	62.15	0.50	0.00	0.00
Direct	8.01	0.00	0.00	1.00
Linear	23.20	0.00	0.00	1.00
OTPA	64.01	1.00	1.00	0.00

(b) FLAME

Attack	Acc.	AUC	TPR	FPR
Clean	58.66	–	–	0.49
Noisy-label training	58.78	0.84	1.00	0.32
Norm-matched noise	58.89	0.84	1.00	0.32
Min-Max	34.20	0.17	0.00	0.65
Min-Sum	38.15	0.21	0.00	0.57
Sign flip	58.87	0.84	1.00	0.32
PoisonedFL	59.52	0.84	1.00	0.32
BadSampler	48.16	0.17	0.00	0.65
Direct	58.67	0.84	1.00	0.32
Linear	58.67	0.84	1.00	0.32
OTPA	58.67	0.84	1.00	0.32

FoolsGold neutralizes noisy-label training, Min-Max, Min-Sum, PoisonedFL, and OTPA, but does not identify norm-matched noise, Direct, or Linear; sign flip and BadSampler receive no positive classifications at the fixed threshold. For OTPA, every malicious client has zero aggregation weight and every benign client has unit weight at the endpoint. The mean pairwise cosine between cumulative malicious classifier-head histories is 0.997 for OTPA, compared with -0.039 for Direct and -0.038 for Linear. This is consistent with alignment removing source-coordinate diversity and creating a shared directional signal even while it reduces full-model Euclidean distance under FedAvg. In the Direct and Linear runs, mutually similar benign correction histories are consistent with FoolsGold's reversed ordering.

FLAME neutralizes noisy-label training, norm-matched noise, sign flip, PoisonedFL, Direct, Linear, and OTPA, but retains Min-Max, Min-Sum, and BadSampler. It rejects every OTPA malicious client and retains 51 benign clients, producing 58.67% final accuracy. FLAME also rejects 49% of clients in the clean control. Its OTPA AUC of 0.84 therefore corresponds to rejecting all 25 malicious clients together with 24 of 75 benign clients, rather than cost-free separation.