DUALSPEC: EXPOSING DISGUISED FEDERATED BACKDOOR VIA CROSS-CLIENT STRUCTURE AND TRAJECTORY RESIDUALS

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated learning (FL) is vulnerable to backdoor attacks that manipulate client updates to implant backdoor features into the global model. Existing server-side defenses often rely on observable discrepancies in update magnitude, coordinate-wise statistics, or on comparisons against a fixed reference model in isolation, which may become unreliable against disguised malicious updates or adaptive trigger attacks. Other approaches rely on auxiliary or trusted datasets, introducing additional data requirements. We propose **DualSpec**, an FL backdoor defense that combines two complementary detection signals, without requiring auxiliary data. The *Model-wise Cross-client* branch applies singular value decomposition (SVD) to per-layer normalized client updates to obtain a shared spectral space across clients, where suspicious groups are identified based on their directional cohesion and isolation. The *Layer-wise Trajectory-residual* branch instead applies SVD to the global model's own recent update to construct a per-layer reference subspace, and compares each client's update against it using residual energy to identify updates that deviate from the recent trajectory. Both branches derive from spectral decomposition of model updates, which exposes their underlying structure beneath surface-level disguise. Finally, malicious updates detected from both branches are combined before aggregation to capture backdoor behavior at both the group and the individual level. Experiments across different datasets and IID/non-IID settings show that DualSpec consistently suppresses a diverse set of state-of-the-art backdoor attacks while preserving main task accuracy, and generalizes beyond convolutional architectures.

1 INTRODUCTION

Federated learning (FL) (Konečnỳ et al., 2016; Fu et al., 2026) enables multiple clients to collaboratively train a global model without sharing raw data. However, the server must rely on client updates whose provenance and reliability are difficult to verify, making FL vulnerable to backdoor attacks (Sun et al., 2019; Feng et al., 2025). Compromised clients can manipulate local training to implant malicious behavior while preserving the global model's performance on the main task. A fundamental challenge for backdoor attacks is persistence, as malicious information can be diluted by benign updates and subsequent optimization, motivating diverse attack mechanisms. Some coordinate multiple compromised clients around a common backdoor objective, allowing their updates to reinforce one another during aggregation (Dai & Li, 2023; Xie et al., 2019). Others embed malicious information in parameters or directions that are less affected by subsequent benign optimization, reducing their dependence on cross-client coordination (Zhang et al., 2022).

Many existing defenses (Huang et al., 2023; Ding et al., 2025) detect malicious clients through discrepancies in the raw update space, such as update magnitude, distance, or coordinate-wise statistics. Such signals can be manipulated by attacks that constrain update statistics (He et al., 2026; Yang et al., 2025), or dynamically adapt to the evolving global model and training process (Zhang et al., 2023) while preserving their malicious objective. Other defenses use server side auxiliary or trusted data to evaluate client behavior (Li & Dai, 2024; Xu et al., 2025c), but such data may be unavailable or poorly representative of client distributions.

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.

We therefore exploit directional and subspace structure in client updates via SVD of per-layer normalized updates. Coordinated attackers may form cohesive spectral groups, whereas non-coordinated attacks may introduce per-layer directions inconsistent with the recent global trajectory. These observations lead to two complementary detection mechanisms: *Model-wise Cross-client* and *Layer-wise Trajectory-residual* detection.

Based on this insight, we propose DualSpec, a server side backdoor defense that requires no auxiliary data. The Cross-client branch represents each client update as a per-layer normalized supervector and embeds clients into a shared spectral space, where HDBSCAN (Campello et al., 2013) identifies cohesive and isolated groups. The Trajectory-residual branch compares each client's per-layer update against the recent optimization trajectory of the global model and detects residual components outside the corresponding reference subspace. By combining these complementary signals before aggregation, DualSpec targets both coordinated group-level and individually persistent backdoor behaviors.

The main contributions of this work are threefold: **(I)** We propose DualSpec, a server-side backdoor defense for FL that combines spectral cross-client analysis with trajectory-based per-client detection, exploiting complementary signals in low dimensional subspaces derived from client updates and from the global model's own optimization trajectory, rather than relying on raw parameter similarity or auxiliary data. **(II)** We introduce a two-branch detection framework: **model-wise Cross-client branch** identifies cohesive, isolated groups of client updates via spectral clustering, while the **layer-wise Trajectory-residual branch** flags individual clients whose per-layer update directions deviate from the global model's recent optimization trajectory, together capturing both coordinated and individually persistent attack behaviors. **(III)** We conduct extensive experiments under both IID and non-IID settings against a diverse set of SOTA backdoor attacks, comparing DualSpec with existing SOTA defenses; the results demonstrate its effectiveness and robustness across attack and data-distribution settings.

2 Related Work

Existing backdoor defenses in FL can be categorized into three paradigms:

• **Robust Aggregation Rules.** These methods replace FedAvg (McMahan et al., 2017) with robust estimators to limit the influence of anomalous updates, including Krum, Multi-Krum (Blanchard et al., 2017), Trimmed Mean, Median (Yin et al., 2018), RFA (Pillutla et al., 2022), RLR (Ozdayi et al., 2021), and LASA (Xu et al., 2025a). However, robustness can degrade under heterogeneous clients and attacks that mimic benign updates.

• **Anomaly Detection and Filtering.** These approaches identify and filter suspicious clients using update similarity, clustering, or spectral information. Representative approaches include Yu & Fang (2026), FLAME (Nguyen et al., 2022), MultiMetrics (Huang et al., 2023), DnC (Shejwalkar & Houmansadr, 2021), FedDLAD (Ding et al., 2025), AlignIns (Xu et al., 2025b), and FoolsGold (Fung et al., 2020). However, high-dimensional spaces weaken distance-based discrimination, while disguised backdoor perturbations can remain entangled with main-task parameters.

• **Proxy/Trust-based Defenses.** These methods leverage server-side auxiliary data to assess client reliability. FLTrust (Cao et al., 2021) uses a trusted root dataset to compute client trust scores, while BackdoorIndicator (Li & Dai, 2024) and MASA (Xu et al., 2025c) exploit OOD or proxy data for backdoor detection. However, they depend on available, private, and representative reference data.

• **Position of Our Work.** From a technical perspective, DnC and FLAME are the two defenses most related to our work. DnC exploits only the dominant spectral component, potentially missing attacks whose structure lies in other directions. FLAME uses HDBSCAN to identify benign groups, making it less effective when malicious updates resemble benign ones. In contrast, **DualSpec** performs client grouping in a shared spectral space, exposing structural discrepancies that are obscured in the original parameter space and thereby improving the separability of malicious and benign clients under backdoor attacks.

From a conceptual perspective, both our method and AlignIns (Xu et al., 2025b) examine each client against a reference derived from the global model, but differ in what that reference is: AlignIns compares each client to the current global model via a scalar distance, whereas our Trajectory-

residual branch uses a per-layer reference subspace derived from the global trajectory. In addition, DualSpec introduces a Cross-client branch that AlignIns lacks, constructing a shared spectral space to detect coordinated behavior. More details of FL backdoor defense provided in Supplementary.

3 METHODOLOGY

3.1 THREAT MODEL

Following Li & Dai (2024) and Huang et al. (2023), we formalize the threat model in terms of the objectives and capabilities of both the attacker and the defender.

• **FL Setting.** We consider a standard FL system consisting of a central server $\mathcal{S}$ and N clients, where each client i possesses a private local dataset D_i. At each communication round, the server selects a subset of M clients to perform local training, receives their updates, filters potentially malicious updates, and aggregates the remainder with FedAvg.

• **Attacker's Goal and Capability.** The attacker implants a backdoor that maps triggered inputs to a target class while preserving the model's performance on benign inputs. The attacker may compromise one or more participating clients and obtain full control over their local training and model update processes. Compromised clients may collude using a shared trigger and target label. The attacker may also choose when to initiate the backdoor attack. Following common practice in Byzantine-robust FL defenses (Blanchard et al., 2017), we assume an honest majority, i.e., the fraction of compromised clients participating in any given round does not exceed 50%.

• **Defender's Goal and Capability.** The defender has white-box access to the models or updates uploaded by participating clients and can inspect and exclude suspicious updates before aggregation. However, the defender has no access to client datasets or their underlying distributions. The defender aims to identify and remove malicious updates from the aggregation process while preserving the global model's performance on the primary task.

3.2 MOTIVATION

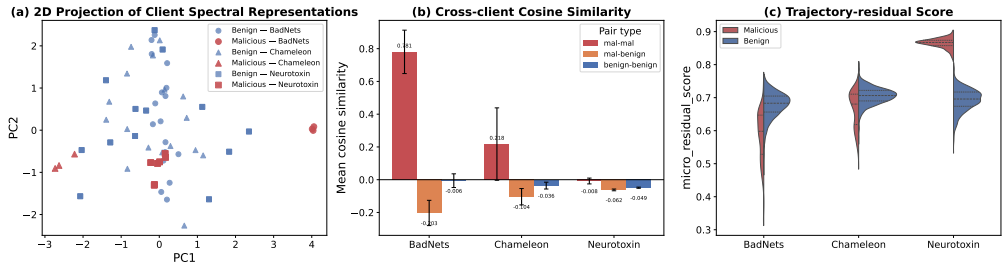


Figure 1: Complementary structure of benign and malicious client updates in spectral and trajectory-residual spaces. (a) 2D projection of client spectral representations at a representative training round. (b) Pairwise cosine similarity between malicious-malicious, malicious-benign, and benign-benign client pairs across evaluation rounds. (c) Distribution of the trajectory-residual score for malicious and benign clients.

Motivated by the possibility that collaborative malicious updates share directional structure despite differing magnitudes and parameters, we conduct a preliminary experiment inspired by the spectral analysis of Shejwalkar & Houmansadr (2021). We train FL models on CIFAR-10 under BadNets (Gu et al., 2017), Neurotoxin (Zhang et al., 2022) and Chameleon (Dai & Li, 2023), with 100 clients (20% malicious, 20 sampled per round). For each client, we per-layer normalize and concatenate its update, then apply SVD to obtain a low-dimensional spectral representation, examining the structure formed by multiple leading components rather than only the dominant one.

As illustrated by the PCA projection of client spectral representations at a representative round in Fig. 1(a), malicious clients form a distinct and cohesive group in the spectral space, while benign clients remain more dispersed. To examine whether this structure persists across rounds, Fig. 1(b) re-

ports pairwise cosine similarities between client spectral representations. Across evaluation rounds, BadNets and Chameleon exhibit higher similarity among malicious clients than between malicious and benign clients. These observations support the first design hypothesis of DualSpec: when compromised clients coordinate around a common poisoning objective, their updates can exhibit a population-level spectral structure distinguishable from benign clients and thus, exploitable for backdoor detection.

This structure across the client population, however, is attack dependent. For Neurotoxin (Zhang et al., 2022), the ordering mal-mal $>$ benign-benign $>$ mal-benign is consistent to BadNets and Chameleon but the gap narrows because its coordinate masking dilutes full-vector directional correlation. The cross-client signal is therefore insufficient for detection.

We therefore measure, per client and layer, the update energy outside the global model's recent trajectory subspace, without comparing client updates to one another. As shown in Fig. 1(c), this trajectory-residual score separates malicious from benign clients under Neurotoxin, where malicious clients concentrate at higher residual values than benign ones, in contrast to BadNets and Chameleon, where the two populations overlap. This indicates that the discriminative signal weakened in the cross-client comparison can reemerge when individual clients are examined relative to their own recent trajectory at the layer level.

These observations motivate the two branches: Cross-client detects coordinated structure across clients, while Trajectory-residual detects deviations from the global trajectory. We formalize the spectral embedding, the cohesion-isolation selection criterion, and the residual score underlying these two branches in the following sections.

3.3 Model-wise Cross-client Spectral Detection

3.3.1 Per-layer Normalized Client Supervectors

At round t, let θ_g^{t-1} denote the global model before aggregation and θ_i^t the model returned by client i, so the client update is $\Delta_i^t = \theta_i^t - \theta_g^{t-1}$. We operate on a selected set of parameter tensors $\mathbb{L}$; for layer $l \in \mathbb{L}$, $\Delta_i^{t,l}$ denotes the corresponding per-layer update. To capture directional structure shared across clients without letting layers with larger update norms dominate, we ℓ_2-normalize each selected layer update and concatenate the results into a fixed-length supervector $\boldsymbol{m}_i$:

$$\boldsymbol{v}_i^{(l)} = \mathrm{vec}(\Delta_i^{t,l}) / \| \mathrm{vec}(\Delta_i^{t,l}) \|_2, \qquad \boldsymbol{m}_i = [\boldsymbol{v}_i^{(l_1)}; \ldots; \boldsymbol{v}_i^{(l_{|\mathbb{L}|})}]. \tag{1}$$

Since each $\boldsymbol{v}_i^{(l)}$ is unit norm, $\|\boldsymbol{m}_i\|_2^2 = |\mathbb{L}|$ for every client. Stacking all n supervectors into $\boldsymbol{M} \in \mathbb{R}^{n \times d}$ and forming the Gram matrix $\boldsymbol{G} = \boldsymbol{M}\boldsymbol{M}^\top$ therefore gives $G_{ij} = |\mathbb{L}| \cos\theta_{ij}$: $\boldsymbol{G}$ is exactly the pairwise cosine similarity matrix between client supervectors, the same quantity used by the clustering step in Sec. 3.3.2.

Because $n \ll d$, we obtain the spectral embedding from $\boldsymbol{G}$, which is equivalent to decomposing $\boldsymbol{M}$ before truncation and is cheaper. Writing the eigendecomposition $\boldsymbol{G} = \boldsymbol{U}\Sigma^2\boldsymbol{U}^\top$, we retain the first $r = 8$ components as each client's spectral representation, $\boldsymbol{s}_i = U_{i,1:r}\Sigma_{1:r,1:r} \in \mathbb{R}^r$. Retained energy analysis and justification for the choice of r presented in Supplementary.

3.3.2 Spectral Clustering and Cluster Scoring Mechanism

We apply HDBSCAN to the spectral representations $\{\boldsymbol{s}_i\}$, to identify suspicious groups without specifying their number or size. We set the minimum cluster size to two, to allow detection of the smallest possible client collusion. HDBSCAN may return both malicious and non malicious candidate clusters. We score each candidate cluster $\mathbb{C}$ to determine which, if any, is most likely to contain colluding attackers.

We first characterize the directional structure of each candidate cluster using pairwise cosine similarity between spectral representations:

$$c_{ij} = \frac{\boldsymbol{s}_i^\top \boldsymbol{s}_j}{\max(\|\boldsymbol{s}_i\|_2, \varepsilon) \max(\|\boldsymbol{s}_j\|_2, \varepsilon)}. \tag{2}$$

The cluster internal direction consistency and its interaction with the remaining clients are measured by:

$$\text{cohesion}(\mathbb{C}) = \frac{1}{\binom{|\mathbb{C}|}{2}} \sum_{\substack{i,j \in \mathbb{C} \\ i<j}} c_{ij}, \qquad \text{isolation}(\mathbb{C}) = \frac{1}{|\mathbb{C}|(n-|\mathbb{C}|)} \sum_{i \in \mathbb{C}} \sum_{j \notin \mathbb{C}} c_{ij}. \tag{3}$$

We compute isolation as the average cosine similarity between members of $\mathbb{C}$ and the remaining clients; lower values indicate stronger isolation. We define the shape score as:

$$\text{shape}(\mathbb{C}) = \text{cohesion}(\mathbb{C}) - \text{isolation}(\mathbb{C}). \tag{4}$$

A high shape score favors candidate groups that are coherent in direction while remaining well separated from the rest of the client population. However, relying on this shape score alone is insufficient. As the global model approaches convergence, updates from many benign clients become aligned as they all shrink toward a common optimum, so a cohesive but benign group can obtain a high shape score. To distinguish true malicious collusions from such benign cohesion, we incorporate the average magnitude of the cluster along the dominant spectral direction as a secondary metric: $\mu_{\text{pc1}}(\mathbb{C}) = \frac{1}{|\mathbb{C}|} \sum_{i \in \mathbb{C}} |s_{i,1}|$, where $s_{i,1}$ denotes the first coordinate of the spectral representation $\boldsymbol{s}_i$. This complements shape by favoring groups with strong projection onto the dominant spectral direction:

$$s_{\text{comb}}(\mathbb{C}) = w_{\text{shape}}\, z(\text{shape}(\mathbb{C})) + (1 - w_{\text{shape}})\, z(\mu_{\text{pc1}}(\mathbb{C}))\,. \tag{5}$$

The weighting parameter w_{shape} is selected based on the characteristics (e.g. number of classes) of each dataset.

3.3.3 Malicious Cluster Selection

We apply the scoring mechanism in two ways. At every round t, we perform detection within the current round using the clients participating in round t. Starting from round $t = 2$, we additionally perform two-round detection using the union of clients from rounds $t-1$ and t.

This extension to two rounds is motivated by two considerations. First, to reflect temporal persistence of a malicious structure across consecutive rounds, the preceding round provides additional points that reinforce the density structure of the malicious group, allowing HDBSCAN to form a stable cluster even when round t alone contains too few malicious clients to satisfy the minimum cluster size; most notably rounds with single malicious client, where no cluster of size ≥ 2 can otherwise be formed. Second, we restrict the additional round to $t-1$ rather than further back in training, since updates from earlier rounds may reflect a different optimization state and therefore introduce update directions that are less representative of the current round, contaminating the joint spectral space. Because this extension requires a preceding round, it is applied starting from round $t = 2$.

Notably, the two rounds must be combined *before* the spectral embedding is computed, rather than by merging two independent computed embeddings: the Gram based decomposition produces a coordinate system specific to the set of supervectors it is computed from, so spectral coordinates obtained from separate decompositions at $t-1$ and t are not directly comparable. We therefore concatenate the undecomposed supervectors from both rounds, $\tilde{\boldsymbol{M}} \in \mathbb{R}^{(n_{t-1}+n_t)\times d}$, and apply the same Gram-trick construction to $\tilde{\boldsymbol{M}}$ to obtain a single joint embedding $\{\tilde{\boldsymbol{s}}_i\}$, over which cohesion, isolation, shape, and dominant spectral direction statistics are computed as before.

To account for the additional temporal information in the joint setting, we introduce a cross round bonus that rewards clusters populated by clients from both rounds, reflecting the persistence of the backdoor spectral representations across time:

$$\text{cross}(\mathbb{C}) = \frac{\min(|\mathbb{C} \cap t{-}1|,\, |\mathbb{C} \cap t|)}{|\mathbb{C}|} \in [0, 0.5], \tag{6}$$

where $|\mathbb{C} \cap t{-}1|$ and $|\mathbb{C} \cap t|$ denote the number of round $(t-1)$ and round t members of $\mathbb{C}$, respectively; the score is maximized when the cluster is balanced across the two rounds, and is zero when all cluster members come from only one round. The combined score used to select the winning cluster in the joint setting becomes:

$$s^{(2)}_{\text{comb}}(\mathbb{C}) = w_{\text{shape}}\, z(\text{shape}(\mathbb{C})) + w_{\text{cross}}\, z(\text{cross}(\mathbb{C})) + \max(0,\, 1 - w_{\text{shape}} - w_{\text{cross}})\, z(\mu_{\text{pc1}}(\mathbb{C})). \tag{7}$$

In both settings, the candidate cluster with the highest combined score: $s_{\text{comb}}(\mathbb{C})$ for the round t case and $s^{(2)}_{\text{comb}}(\mathbb{C})$ for the joint case, is selected as the suspicious group. In the joint case, only its round t members are retained as flagged clients, since round $(t-1)$ membership serves only to inform the current decision and is not itself reflagged.

3.3.4 CLUSTER REFINEMENT

Two refinement mechanisms are applied to the winning cluster $\mathbb{C}_{\text{win}}$ selected in Sec. 3.3.3, in order: Gating, then Expansion.

Gating. A small winning cluster can obtain an unstable or inflated shape score, since few pairs are enough to determine its cohesion, letting it outscore a larger, more representative runner-up $\mathbb{C}_{\text{run}}$. When $\mathbb{C}_{\text{win}}$ is small and only ahead of a larger candidate, we discount its shape score by a confidence factor that grows with the number of pairs in the cluster and recompute s_{comb} for all candidates; $\mathbb{C}_{\text{win}}$ is replaced by $\mathbb{C}_{\text{run}}$ if the discounted score now favors it. This discount is applied only in this narrow condition, since penalizing every cluster's shape score by size would also suppress truly small malicious groups.

Expansion. Density-based clustering can fragment a single coherent attacker group into several smaller subclusters. After gating, we search among the remaining candidates for the one most similar to the (possibly updated from gating) winner,

$$\mathbb{C}^\star = \arg\max_{\mathbb{C}' \neq \mathbb{C}_{\text{win}}} \text{affinity}(\mathbb{C}_{\text{win}}, \mathbb{C}'), \qquad \text{affinity}(\mathbb{C}, \mathbb{C}') = \frac{1}{|\mathbb{C}||\mathbb{C}'|} \sum_{i \in \mathbb{C}} \sum_{j \in \mathbb{C}'} c_{ij}, \tag{8}$$

and merge it into the winner if its affinity exceeds the winner's own isolation from the rest of the population by a margin τ_{expand}.

Rather than a fixed similarity threshold, this margin condition requires $\mathbb{C}^\star$ to be closer to the winner than the winner is, on average, to the rest of the population; consistent with $\mathbb{C}^\star$ being a fragment of the same attacker group rather than an unrelated benign cluster. At most one additional cluster is merged per round. The exact Gating condition, the $s_{\text{cap}}, \tau_{\text{size}}, \tau_{\text{gap}}, \tau_{\text{expand}}$ thresholds, and the scope of Expansion are given in Supplementary.

3.4 LAYER-WISE TRAJECTORY-RESIDUAL DETECTION

For attacks such as Neurotoxin, which lack consistent cross-client structure, we instead compare each client update with the recent global trajectory. When attackers cannot reinforce the backdoor through a shared update direction, maintaining the malicious behavior may require introducing update directions that differ from those recently followed by the global model. Such deviations can therefore manifest as residual energy outside the recent global trajectory, particularly at the layers affected by the attack.

Motivated by this observation, the Trajectory-residual branch evaluates each client update against a reference subspace derived from the server's own recent trajectory, rather than against other clients. The reference is constructed per layer to capture localized deviations. For each client and layer, we then measure the fraction of update energy lying outside this reference subspace. Unlike the Cross-client branch, this signal does not require cross-client coherence.

3.4.1 PER-LAYER REFERENCE SUBSPACE FROM THE GLOBAL TRAJECTORY

For each layer $l \in \mathbb{L}$, we reshape the corresponding update into a matrix by flattening all dimensions except the leading output-channel dimension, so that singular vectors can be interpreted per output channel. The most recent global trajectory at layer l, $\Delta_g^{t-1,l} = \theta_g^{t-1,l} - \theta_g^{t-2,l}$, is decomposed via SVD,

$$\Delta_g^{t-1,l} = \boldsymbol{U}^{(l)} \boldsymbol{\Sigma}^{(l)} \boldsymbol{V}^{(l)\top}. \tag{9}$$

Here, each per-layer trajectory matrix is decomposed directly via SVD.

We form the reference subspace from the leading r left-singular vectors, representing the directions along which the global model has recently moved at layer l:

$$\boldsymbol{U}^{(l)}_{\text{ref}} = \boldsymbol{U}^{(l)}_{[:,1:r]}. \tag{10}$$

3.4.2 RESIDUAL SCORE AND CLIENT FLAGGING

For each client i, we measure the fraction of its trajectory-residual update, reshaped in the same way as above, that lies outside this reference subspace:

$$\rho_i^{(t,l)} = \frac{\left\| \Delta_i^{t,l} - \boldsymbol{U}_{\text{ref}}^{(l)} \boldsymbol{U}_{\text{ref}}^{(l)\top} \Delta_i^{t,l} \right\|_F^2}{\left\| \Delta_i^{t,l} \right\|_F^2}. \tag{11}$$

As an energy ratio, $\rho_i^{(t,l)}$ is invariant to the overall magnitude of the client update and therefore captures its directional deviation from the server's recent trajectory.

The trajectory-residuals are averaged to obtain a client-level score,

$$R_i^t = \frac{1}{|\mathbb{L}|} \sum_{l \in \mathbb{L}} \rho_i^{(t,l)}. \tag{12}$$

We then compute robust z-scores across clients using the median and median absolute deviation (MAD), and flag client i as $\text{micro} = \{i : z_i^R > \lambda\}$, with $\lambda = 2.0$. The Trajectory-residual branch is skipped in round $t = 1$, since θ_g^{t-2} is not yet available; it is applied starting from round $t = 2$.

3.5 AGGREGATING DETECTION SIGNALS

The detection procedure combines signals from both branches. cross_client denotes the clients flagged by the Cross-client branch on the current round (Sec. 3.3); $\text{cross_client_2_round}$ denotes current-round clients flagged by the two-round extension, which uses rounds $t-1$ and t to construct the spectral space (starting from $t = 2$); and $\text{trajectory_residual}$ denotes the clients flagged by the Trajectory-residual branch (Sec. 3.4). The flagged indexes are:

$$\text{flagged_idx} = \text{cross_client} \cup \text{cross_client_2_round} \cup \text{trajectory_residual}. \tag{13}$$

DualSpec's full algorithmic procedure and its complexity analysis are provided in Supplementary.

4 EXPERIMENTS

4.1 SETUP

Following prior work of Xu et al. (2025b), we adopt the experimental setting described below.

Datasets and models. We conduct the main experiments on CIFAR10 with ResNet-18 and CIFAR100 (Krizhevsky et al., 2009) with ResNet-34 (He et al., 2016), using Dirichlet non-IID partitions (Hsu et al., 2019) with $\alpha = 0.5$.

Baselines. We evaluate six representative backdoor attacks and eight defense/aggregation baselines, covering diverse attack mechanisms and defense categories.

FL setting. We consider 100 clients, with 20 clients randomly selected in each round. Clients perform local training using SGD. Unless otherwise specified, 20% of the clients are malicious.

Attack setting. All attacks use a 5×5 trigger, except DBA (Xie et al., 2019), which uses a 4×4 trigger distributed across the four corners. Unless otherwise specified, the attacker initiates poisoning at round 0.

Metrics. We report three primary metrics. *Main task accuracy (MA)* is the classification accuracy of the aggregated global model on a clean test set, reflecting utility preservation. *Attack success rate (ASR)* is the fraction of triggered test samples misclassified into the attacker's target label, reflecting backdoor effectiveness under each defense. *Malicious elimination rate (MER)* is the fraction of malicious client participations that are correctly identified and excluded from aggregation over the entire FL process, reflecting detection quality independent of its downstream effect on MA and ASR.

Additional experimental configurations, detailed descriptions of the attack and defense baselines, and results including IID and more heterogeneous non-IID settings ($\alpha = 0.1$), blended triggers, additional datasets, varying poisoning ratios, and further ablations are provided in Supplementary.

Table 1: Performance comparison of DualSpec and baseline defenses on CIFAR10 and CIFAR100 under backdoor attacks. We report Main Task Accuracy (MA, ↑), Attack Success Rate (ASR, ↓), and Malicious Elimination Rate (MER, ↑). Avg. Rank (↓) is the average rank (1 = best) over all combinations within each dataset, computed among the eight defenses; FedAvg is excluded. For each dataset, attack, and metric (including Avg. Rank), the **best** and second best results are highlighted. DualSpec achieves the best average rank on both datasets, outperforming the second-best defense, AlignIns in CVPR 2025.

Defense	BadNets $_{2017}$			Neurotoxin$_{ICML2022}$			Chameleon$_{ICML2023}$			DBA$_{ICLR2020}$			BATMAN$_{CVPR2026}$			A3FL$_{NeurIPS2023}$			Avg.
	MA(↑)	ASR(↓)	MER(↑)	MA	ASR	MER	MA	ASR	MER	MA	ASR	MER	MA	ASR	MER	MA	ASR	MER	Rank(↓)
CIFAR10																			
FedAvg	86.14	78.10	0.00	86.41	72.86	0.00	87.11	96.96	0.00	84.08	81.87	0.00	86.31	85.53	0.00	87.58	100.00	0.00	–
MultiKrum	83.06	1.00	82.93	85.19	3.69	64.73	83.79	52.32	23.67	82.44	24.92	71.65	86.40	1.04	**99.03**	87.43	100.00	10.51	4.61
FLTrust	55.77	0.89	64.90	68.00	56.31	60.86	38.04	11.01	24.59	48.32	11.26	61.51	46.28	6.01	38.01	49.80	100.00	17.87	6.28
DnC	87.33	1.02	96.75	85.80	57.63	22.05	86.52	96.92	25.98	87.11	1.80	96.75	86.50	0.87	98.29	87.18	28.41	88.51	3.67
FLAME	84.92	66.83	38.80	84.54	44.56	3.24	85.14	88.67	0.00	55.23	16.54	38.12	56.88	3.58	51.28	86.02	100.00	13.68	6.28
Indicator	83.13	0.76	94.70	84.48	2.54	74.87	80.26	97.84	42.56	81.58	25.45	89.23	81.97	25.22	92.30	82.05	100.00	92.43	5.17
AlignIns	86.35	0.86	96.24	86.80	0.86	95.21	87.07	1.10	92.47	86.76	2.03	98.96	87.78	0.87	97.40	86.57	100.00	43.80	2.94
FedDLAD	69.17	7.46	48.95	80.59	1.28	64.25	81.82	80.80	40.77	59.46	67.54	46.53	84.23	4.83	93.07	84.84	100.00	17.71	6.00
DualSpec	**87.59**	**0.64**	**100.00**	**87.89**	**0.77**	**98.98**	**87.42**	**0.63**	**100.00**	**87.44**	**0.81**	**99.82**	**87.79**	**0.78**	98.40	**87.65**	**2.22**	**95.84**	**1.06**
CIFAR100																			
FedAvg	40.67	75.63	0.00	63.69	88.01	0.00	61.91	72.13	0.00	32.36	66.95	0.00	47.79	81.87	0.00	63.35	100.00	0.00	–
MultiKrum	62.64	0.22	95.86	57.08	34.64	4.24	56.72	77.34	15.61	62.70	0.50	97.01	60.96	0.43	98.62	61.11	99.99	50.61	4.44
FLTrust	2.02	45.32	17.43	10.28	2.52	4.76	3.80	7.18	38.35	2.12	15.96	14.23	3.36	0.20	36.74	9.84	18.84	27.13	6.39
DnC	61.74	2.08	89.43	62.33	89.57	9.07	61.14	0.50	21.35	61.59	0.51	89.55	61.58	0.61	89.55	63.40	100.00	44.66	4.56
FLAME	39.70	85.50	11.59	54.20	62.25	0.00	51.17	90.10	0.00	49.56	52.95	7.57	53.58	37.95	30.31	55.45	100.00	5.62	7.00
Indicator	35.73	**0.00**	**100.00**	37.20	0.19	95.63	30.83	97.72	70.14	39.44	**0.13**	**100.00**	19.55	74.69	**99.75**	37.30	6.45	97.31	4.11
AlignIns	62.77	0.58	95.40	**64.24**	0.40	97.70	64.19	0.56	96.56	63.52	0.30	97.81	**62.03**	0.23	81.97	63.79	16.29	94.94	2.78
FedDLAD	58.06	0.34	73.59	61.03	**0.12**	69.57	60.24	93.22	29.85	12.91	2.17	74.97	61.38	0.35	77.26	61.48	100.00	33.18	5.00
DualSpec	**62.88**	0.13	97.94	63.81	0.19	**99.23**	**64.37**	**0.24**	**99.38**	**64.13**	0.20	98.28	60.59	**0.10**	89.67	**64.01**	**5.91**	**97.93**	**1.72**

4.2 Experimental Results

4.2.1 DualSpec performance

We first evaluate DualSpec under the standard setting, comparing it with attack and defense baselines. Table 1 reports MA, ASR, and MER for each attack-defense pair on CIFAR10 and CIFAR100. The results demonstrate that DualSpec achieves the best or second best result in 34 of 36 combinations. The closest competitor is AlignIns, which also relies on the direction of client updates rather than their magnitude. The difference is most pronounced on A3FL, where DualSpec improves MER over AlignIns on CIFAR10, while achieving a smaller improvement on CIFAR100.

On BATMAN/CIFAR100, early-stage instability and its subspace-based construction may weaken detection, but DualSpec still achieves 0.10% ASR with competitive MA and MER. In contrast, FLTrust, FLAME, and FedDLAD show larger MA degradation without suppressing ASR.

4.2.2 Robustness to Delayed Attack

Delayed attacks can be more effective, as smaller and more stable update magnitudes near convergence make them stealthier (Zhang et al., 2023).

Table 2: Robustness to delayed attack initiation on CIFAR10. We compare DualSpec with baseline defenses when backdoor poisoning starts at communication rounds 0, 50, or 100.

Defense	Attack start round	BadNets			Neurotoxin			Chameleon		
		MA (↑)	ASR (↓)	MER (↑)	MA (↑)	ASR (↓)	MER (↑)	MA (↑)	ASR (↓)	MER (↑)
Multi-Krum	0	83.06	1.00	82.93	85.19	3.69	64.73	83.79	52.32	23.67
	50	85.44	1.10	87.22	85.48	1.18	76.86	85.66	69.62	34.93
	100	85.44	1.02	97.05	85.40	1.13	79.90	85.98	71.20	53.88
DnC	0	87.33	1.02	96.75	85.80	57.63	22.05	86.52	96.92	25.98
	50	87.06	1.01	98.07	86.30	73.92	34.21	87.23	**0.86**	92.77
	100	87.43	0.96	98.96	86.76	18.62	36.08	87.60	**0.80**	95.87
AlignIns	0	86.35	0.86	96.24	86.80	0.86	95.21	87.07	1.10	92.47
	50	85.98	1.06	93.73	86.22	0.95	92.77	85.82	1.20	94.69
	100	86.08	1.05	90.68	86.74	**0.67**	87.74	86.83	0.95	90.68
Indicator	0	83.13	0.76	94.70	84.48	2.54	74.87	80.26	97.84	42.56
	50	79.47	33.27	89.63	84.48	1.36	98.55	80.56	98.31	71.80
	100	84.27	3.72	87.25	84.26	1.45	99.50	83.38	97.85	65.19
DualSpec	0	**87.59**	**0.64**	**100.00**	**87.89**	**0.77**	**98.98**	**87.42**	**0.63**	**100.00**
	50	**87.58**	**0.63**	**100.00**	**87.16**	**0.92**	**100.00**	**87.63**	0.91	**100.00**
	100	**87.56**	**0.78**	**100.00**	**87.74**	0.71	**100.00**	**87.71**	0.82	**100.00**

We therefore evaluate DualSpec under a delayed attack by initiating poisoning at rounds 50 and 100, representing later stages of training. As shown in Table 2, DualSpec remains stable across attack start rounds, with 100% MER for BadNets and Chameleon and 98.98–100% for Neurotoxin. The slight drop at round 0 reflects the Trajectory-residual branch's one-round dependency.

4.2.3 Robustness to Increasing Malicious-Client Ratios

We evaluate malicious ratios of 10-40%, remaining below the 50% honest-majority assumption from Blanchard et al. (2017). As the malicious ratio increases (Fig. 2), BadNets and Chameleon remain unaffected: because our Cross-client branch relies on density-based clustering, larger fraction of

colluding malicious forms denser, more separable cluster rather than a harder one to isolate. Our cluster refinement step (Expand) further guards against this cluster being fragmented.

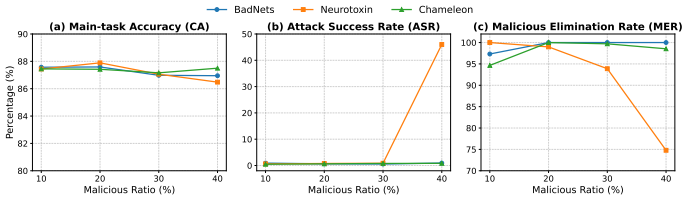


Figure 2: DualSpec robustness to Increasing Malicious-Client Ratios

Neurotoxin, however, degrades at high malicious ratios because malicious residual scores increasingly affect the median/MAD calibration. DualSpec remains effective up to 30%, while at 40% ASR rises and MER drops.

4.2.4 DualSpec components ablation

DualSpec combines two detection branches: the Cross-client branch and the Trajectory-residual branch. To isolate the contribution of each branch, we evaluate three configurations (Fig. 3): the Cross-client branch alone, the Trajectory-residual branch alone, and their union.

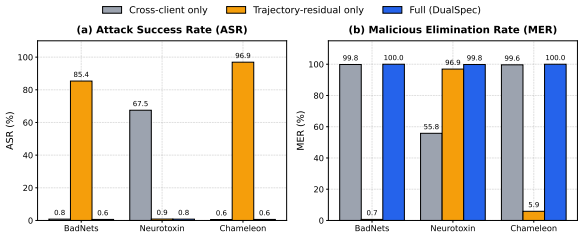


Figure 3: DualSpec branches components ablation

The ablation confirms the complementary roles of the two branches. The Cross-client branch suppresses collaborative attacks, such as BadNets and Chameleon; but is weaker on attacks that target less influential parameters, e.g., Neurotoxin, whereas the Trajectory-residual branch shows the opposite pattern. Their union therefore covers the complementary attack behaviors observed in Fig. 1.

4.2.5 Adaptation to Transformer-based FL

Prior FL security work targets CNN architectures; LoRA (Hu et al., 2021) has since made Transformer-based models practical in FL, therefore we examine whether DualSpec generalizes to them. We evaluate DualSpec with ViT-Small (Dosovitskiy et al., 2020) on CIFAR10 under BadNets and Neurotoxin, comparing it with representative baselines. We fine-tune ViT-Small via LoRA (rank $= 4$) for 100 communication rounds, injecting the backdoor from round 10 onward, while keeping the remaining factors identical to our standard setting. We apply DualSpec to the reconstructed update matrix BA, rather than to the individual LoRA factors A and B. As shown in Table 3, DualSpec maintains low ASR, high MER, and comparable MA under both attacks.

Table 3: Performance comparison on transformer-based federated learning. Results are obtained on CIFAR10 using ViT-Small with LoRA

Defense	BadNets			Neurotoxin		
	MA (↑)	ASR (↓)	MER (↑)	MA (↑)	ASR (↓)	MER (↑)
FedAvg	98.35	93.84	0.00	98.33	89.66	0.00
Multi-Krum	98.14	92.01	47.44	98.20	**0.01**	81.41
DnC	98.34	**0.01**	85.26	98.36	**0.01**	81.41
DualSpec	**98.38**	**0.01**	**97.44**	**98.44**	**0.01**	**91.67**

5 Conclusion

We presented DualSpec, a server side backdoor defense for FL that requires no auxiliary data. By combining cross-client spectral structure with trajectory-residual analysis, DualSpec detects both coordinated and individually persistent backdoor behaviors. Experiments across diverse attacks and heterogeneous settings demonstrate effective backdoor suppression while maintaining main-task performance, with further generalization to transformer-based FL. Future work includes extending DualSpec to detect and filter Byzantine updates beyond backdoor attacks, as well as improving its robustness under lower honest-majority margins and more heterogeneous federated settings.

AI USE STATEMENT

AI USE STATEMENT In this work, we used generative AI tools (Claude and ChatGPT) to sketch preliminary drafts of selected parts of the manuscript, improve clarity and readability, and check the logical consistency of mathematical derivations. All AI-assisted content was subsequently reviewed, revised, and independently verified by the authors.

Generative AI tools were not used to develop the core theoretical models, formulate the main mathematical claims or results, or generate synthetic datasets. The authors take full responsibility for the accuracy, originality, and content of the manuscript.

REPRODUCIBILITY STATEMENT

To ensure the reproducibility of our results, we have provided comprehensive details throughout the paper and Supplemetary Materials. The detailed pseudocode for our proposed method, DualSpec, is presented in Algorithm 1 (Sup. A.1). The formal time and space complexity analysis is detailed in Sup. A.5. Furthermore, complete implementation details, including hyperparameter configurations for DualSpec as well as all baseline models, are provided in Sup. A.7. The full source code to reproduce all experimental results will be made publicly available upon acceptance.

REFERENCES

Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. *Advances in neural information processing systems*, 30, 2017.

Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining*, pp. 160–172. Springer, 2013.

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*. The Internet Society, 2021.

Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017.

Yanbo Dai and Songze Li. Chameleon: Adapting to peer images for planting durable backdoors in federated learning. In *International Conference on Machine Learning*, pp. 6712–6725. PMLR, 2023.

Binbin Ding, Penghui Yang, and Sheng-Jun Huang. Feddlad: A federated learning dual-layer anomaly detection framework for enhancing resilience against backdoor attacks. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-25*, pp. 5021–5029, 2025.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Yunhao Feng, Yanming Guo, Yinjian Hou, Yulun Wu, Mingrui Lao, Tianyuan Yu, and Gang Liu. A survey of security threats in federated learning. *Complex & Intelligent Systems*, 11(2):165, 2025.

Jie Fu, Yuan Hong, Xinpeng Ling, Leixia Wang, Xun Ran, Zhiyu Sun, Wendy Hui Wang, Zhili Chen, and Yang Cao. Differentially private federated learning: A systematic review. *ACM Computing Surveys*, 58(11):1–38, 2026.

Clement Fung, Chris JM Yoon, and Ivan Beschastnikh. The limitations of federated learning in sybil settings. In *23rd International symposium on research in attacks, intrusions and defenses (RAID 2020)*, pp. 301–316, 2020.

Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017.

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

Wenwen He, Wenke Huang, Yiyang Fang, Wenjie Qu, Jiaheng Zhang, and Mang Ye. Batman: Benign knowledge alignment through malicious null space in federated backdoor attack. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13316–13325, 2026.

Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019.

Siquan Huang, Yijiang Li, Chong Chen, Leyu Shi, and Ying Gao. Multi-metrics adaptively identifies backdoors in federated learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4652–4662, October 2023.

Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

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

Songze Li and Yanbo Dai. {BackdoorIndicator}: Leveraging {OOD} data for proactive backdoor detection in federated learning. In *33rd USENIX Security Symposium (USENIX Security 24)*, pp. 4193–4210, 2024.

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

Thien Duc Nguyen, Phillip Rieger, Huili Chen, Hossein Yalame, Helen Möllering, Hossein Fereidooni, Samuel Marchal, Markus Miettinen, Azalia Mirhoseini, Shaza Zeitouni, et al. {FLAME}: Taming backdoors in federated learning. In *31st USENIX Security Symposium (USENIX Security 22)*, pp. 1415–1432, 2022.

Mustafa Safa Ozdayi, Murat Kantarcioglu, and Yulia R. Gel. Defending against backdoors in federated learning with robust learning rate. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 9268–9276, 2021. doi: 10.1609/aaai.v35i10.17118.

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

Virat Shejwalkar and Amir Houmansadr. Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning. In *Network and Distributed System Security Symposium*, 2021.

Ziteng Sun, Peter Kairouz, Ananda Theertha Suresh, and H Brendan McMahan. Can you really backdoor federated learning? *arXiv preprint arXiv:1911.07963*, 2019.

Chulin Xie, Keli Huang, Pin-Yu Chen, and Bo Li. Dba: Distributed backdoor attacks against federated learning. In *International conference on learning representations*, 2019.

Jiahao Xu, Zikai Zhang, and Rui Hu. Achieving byzantine-resilient federated learning via layer-adaptive sparsified model aggregation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2025a.

Jiahao Xu, Zikai Zhang, and Rui Hu. Detecting backdoor attacks in federated learning via direction alignment inspection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20654–20664, June 2025b.

Jiahao Xu, Zikai Zhang, and Rui Hu. Identify backdoored model in federated learning via individual unlearning. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 7960–7969. IEEE, 2025c.

Qingqian Yang, Peishen Yan, Xiaoyu Wu, Jiaru Zhang, Tao Song, Yang Hua, Hao Wang, Liangliang Wang, and Haibing Guan. Stealthy backdoor attack in federated learning via adaptive layer-wise gradient alignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 29163–29172, 2025.

Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International conference on machine learning*, pp. 5650–5659. Pmlr, 2018.

Jingfan Yu and Zhixuan Fang. Near optimal robust federated learning against data poisoning attack. In *International Conference on Learning Representations*, volume 2026, pp. 56361–56384, 2026.

Hangfan Zhang, Jinyuan Jia, Jinghui Chen, Lu Lin, and Dinghao Wu. A3fl: Adversarially adaptive backdoor attacks to federated learning. *Advances in neural information processing systems*, 36: 61213–61233, 2023.

Zhengming Zhang, Ashwinee Panda, Linyue Song, Yaoqing Yang, Michael Mahoney, Prateek Mittal, Ramchandran Kannan, and Joseph Gonzalez. Neurotoxin: Durable backdoors in federated learning. In *International Conference on Machine Learning*, pp. 26429–26446. PMLR, 2022.

DUALSPEC SUPPLEMENTARY MATERIALS

This Supplementary Materials section provides additional theoretical analyses and experimental results that complement the main paper.

A SUPPLEMENTARY TECHNICAL DETAILS

A.1 DUALSPEC ALGORITHM

Algorithm 1 DualSpec: Server-side Backdoor Detection at Round t

Require: Global model θ_g^{t-1} (and θ_g^{t-2} if $t \geq 2$); client updates $\{\Delta_i^t\}_{i=1}^n$; previous-round supervectors $\{\boldsymbol{m}_i^{t-1}\}$ if $t \geq 2$
Ensure: Flagged client set $\mathcal{F}$; updated global model θ_g^t

// Model-wise Cross-client branch (Sec. 3.3)
for $i = 1, \ldots, n$ **do**
  Compute per-layer normalized supervector $\boldsymbol{m}_i$ ▷ Eq. 1
end for
$\{\boldsymbol{s}_i\} \leftarrow$ spectral embedding of $\{\boldsymbol{m}_i\}$ via Gram-trick ▷ Sec. 3.3.1
$\mathcal{C}_{\text{cur}} \leftarrow \text{HDBSCAN}(\{\boldsymbol{s}_i\})$
$C_{\text{win}} \leftarrow \arg\max_{C \in \mathcal{C}_{\text{cur}}} s_{\text{comb}}(C)$ ▷ Eqs. 2–5
$C_{\text{win}} \leftarrow \text{Gating}(C_{\text{win}}, \mathcal{C}_{\text{cur}})$, then $\text{Expansion}(C_{\text{win}}, \mathcal{C}_{\text{cur}})$ ▷ Sec. 3.3.4
$\text{cross_client} \leftarrow C_{\text{win}}$
if $t \geq 2$ **then**
  $\{\tilde{\boldsymbol{s}}_i\} \leftarrow$ joint spectral embedding from $\{\boldsymbol{m}_i^{t-1}\} \cup \{\boldsymbol{m}_i^t\}$
  $\mathcal{C}_{\text{joint}} \leftarrow \text{HDBSCAN}(\{\tilde{\boldsymbol{s}}_i\})$
  $C_{\text{win}}^{(2)} \leftarrow \arg\max_{C \in \mathcal{C}_{\text{joint}}} s_{\text{comb}}^{(2)}(C)$ ▷ Eqs. 6–7
  $C_{\text{win}}^{(2)} \leftarrow \text{Gating}(C_{\text{win}}^{(2)}, \mathcal{C}_{\text{joint}})$, then $\text{Expansion}(C_{\text{win}}^{(2)}, \mathcal{C}_{\text{joint}})$
  $\text{cross_client_2_round} \leftarrow$ round-t members of $C_{\text{win}}^{(2)}$
else
  $\text{cross_client_2_round} \leftarrow \emptyset$
end if
// Layer-wise Trajectory-residual branch (Sec. 3.4)
if $t \geq 2$ **then**
  for $l \in \mathcal{L}$ **do**
    $\Delta_g^{t-1,l} \leftarrow \theta_g^{t-1,l} - \theta_g^{t-2,l}$
    $U_{\text{ref}}^{(l)} \leftarrow$ leading r left-singular vectors of $\Delta_g^{t-1,l}$ ▷ Eqs. 9–10
  end for
  for $i = 1, \ldots, n$ **do**
    for $l \in \mathcal{L}$ **do**
      Compute residual energy ratio $\rho_i^{(t,l)}$ ▷ Eq. 11
    end for
    $R_i^t \leftarrow \frac{1}{|\mathcal{L}|} \sum_{l \in \mathcal{L}} \rho_i^{(t,l)}$ ▷ Eq. 12
  end for
  $z^R \leftarrow$ robust z-scores of $\{R_i^t\}_{i=1}^n$ via median/MAD
  $\text{trajectory_residual} \leftarrow \{i : z_i^R > \lambda\}, \quad \lambda = 2.0$
else
  $\text{trajectory_residual} \leftarrow \emptyset$
end if
// Aggregation (Sec. 3.5)
$\mathcal{F} \leftarrow \text{cross_client} \cup \text{cross_client_2_round} \cup \text{trajectory_residual}$ ▷ Eq. 13
$\theta_g^t \leftarrow \theta_g^{t-1} + \text{FedAvg}\left(\{\Delta_i^t : i \notin \mathcal{F}\}\right)$
return $\mathcal{F}$, θ_g^t

A.2 Extended Discussion of Existing Backdoor Defenses

Existing backdoor defenses in Federated Learning (FL) primarily seek to reduce the influence of malicious client updates during or prior to the aggregation process. These approaches can be categorized into three main paradigms:

• **Robust Aggregation Rules**: Robust aggregation methods replace the conventional FedAvg (McMahan et al., 2017) with statistical estimators designed to limit the influence of anomalous client updates. Representative approaches include Krum, Multi-Krum (Blanchard et al., 2017), Trimmed Mean, and Median (Yin et al., 2018), as well as RFA (Pillutla et al., 2022), which employs the geometric median for robust aggregation. More recent approaches include RLR (Ozdayi et al., 2021), which determines server-side learning rates based on the signs of client updates, and LASA (Xu et al., 2025a), which combines model sparsification with adaptive layer-wise filtering.

Despite their robustness against certain Byzantine behaviors, these methods can experience substantial utility degradation under heterogeneous and non-IID client distributions, where benign updates may exhibit large statistical deviations. Moreover, robust aggregation rules can be circumvented by adaptive attacks that constrain or optimize malicious updates to remain statistically similar to benign updates, including model-scaling and optimization-based attacks.

• **Anomaly Detection and Filtering**: This paradigm identifies suspicious client updates and excludes or downweights them before aggregation. FLAME (Nguyen et al., 2022) employs HDBSCAN based clustering together with model clipping and noise injection to mitigate malicious updates. MultiMetrics (Huang et al., 2023) combines multiple distance-based metrics for client screening, while DnC (Shejwalkar & Houmansadr, 2021) employs spectral analysis based on singular value decomposition to identify anomalous update directions. More recent approaches include FedDLAD (Ding et al., 2025), which adopts a dual-layer detection framework based on COF and IQR, and AlignIns (Xu et al., 2025b), which evaluates the alignment of client update directions. FoolsGold (Fung et al., 2020), in contrast, exploits the historical similarity of client updates to identify and penalize colluding clients.

However, distance and representation based detection methods face fundamental challenges in high dimensional model spaces, where distance concentration can reduce the discriminative power of conventional metrics. Moreover, backdoor perturbations may be entangled with parameters that are also important for the main task. Such entanglement makes malicious behavior difficult to isolate without affecting benign updates, potentially increasing false positives and reducing model utility. These limitations become pronounced when attackers optimize their updates to remain close to the benign update distribution.

• **Proxy/Trust based Defenses**: These approaches leverage a small amount of auxiliary data available at the server to evaluate the reliability of client updates. FLTrust (Cao et al., 2021) computes client trust scores according to the similarity between client updates and a server update obtained from a trusted root dataset. BackdoorIndicator (Li & Dai, 2024) introduces indicator tasks based on out-of-distribution (OOD) data to monitor potentially malicious client behavior, while MASA (Xu et al., 2025c) utilizes proxy data to perform localized machine unlearning and identify models exhibiting anomalous unlearning behavior.

Although auxiliary data can provide a useful reference for distinguishing malicious updates, these approaches introduce additional assumptions regarding the availability, privacy, and representativeness of the server side reference data. In particular, a reference dataset that poorly reflects the underlying global distribution may lead to unreliable trust estimation or detection decisions.

A.3 Full Derivation: Gram-trick Spectral Embedding and Retained Energy

Equivalence to direct SVD. Let $\boldsymbol{M} = \boldsymbol{U}_M \Sigma_M \boldsymbol{V}_M^\top$ denote the thin SVD of the supervector matrix $\boldsymbol{M} \in \mathbb{R}^{n \times d}$ ($n \ll d$), so that $\boldsymbol{M}\boldsymbol{M}^\top = \boldsymbol{U}_M \Sigma_M^2 \boldsymbol{U}_M^\top$. Forming the Gram matrix $\boldsymbol{G} = \boldsymbol{M}\boldsymbol{M}^\top \in \mathbb{R}^{n \times n}$ and taking its eigendecomposition $\boldsymbol{G} = \boldsymbol{U}\Sigma^2\boldsymbol{U}^\top$ therefore recovers exactly $\boldsymbol{U} = \boldsymbol{U}_M$ and $\Sigma = \Sigma_M$ (up to sign and ordering within degenerate eigenspaces), matching the left singular vectors and singular values of the direct decomposition of $\boldsymbol{M}$. This formulation never forms or stores the d-dimensional right singular vectors $\boldsymbol{V}_M$, which the client space embedding does not use. In practice we symmetrize $\boldsymbol{G}$ to remove floating point asymmetry and clip the

small negative eigenvalues introduced by numerical error to zero before taking the square root. The spectral representation of client i is the i-th row of $\boldsymbol{U}\Sigma$ restricted to the top r components, $\boldsymbol{s}_i = U_{i,1:r}\Sigma_{1:r,1:r} \in \mathbb{R}^r$; stacking over all clients gives $S = U_{:,1:r}\Sigma_{1:r,1:r}$, matching the spectral representation defined in Sec. 3.3.1.

Complexity. Decomposing $\boldsymbol{M}$ directly costs $O(n^2 d)$ for the thin SVD, but also requires computing and holding $\boldsymbol{V}_M \in \mathbb{R}^{d\times n}$. The Gram-trick instead costs $O(n^2 d)$ to form $\boldsymbol{G}$ plus $O(n^3)$ to eigendecompose the $n \times n$ matrix, i.e. $O(n^2 d + n^3)$ overall, while never constructing $\boldsymbol{V}_M$; since $n \ll d$ in our setting (tens of participating clients versus millions of parameters), this removes the dominant memory cost of the direct decomposition.

Retained spectral energy and choice of r. In a preliminary analysis on CIFAR10 under the BadNets and Chameleon attacks (Sec. 3.2), retaining $r = 4$, 8, and 16 spectral components preserves approximately 55%, 92%, and 98% of the total spectral energy $\sum_i \Sigma_{ii}^2$, respectively. We choose $r = 8$ as a practical trade-off, retaining most of the dominant spectral energy while keeping the spectral space compact enough for HDBSCAN to remain effective in low sample-size regimes.

A.4 Full Derivation: Cluster Refinement (Gating and Expansion)

Gating. We discount the shape score of a candidate cluster $\mathbb{C}$ by a confidence factor that grows with the number of internal pairs, saturating at a cap s_{cap}:

$$\text{conf}(\mathbb{C}) = \sqrt{\frac{\min\left(\binom{|\mathbb{C}|}{2}, \binom{s_{\text{cap}}}{2}\right)}{\binom{s_{\text{cap}}}{2}}}, \qquad \text{shape}_{\text{scaled}}(\mathbb{C}) = \text{conf}(\mathbb{C}) \cdot \text{shape}(\mathbb{C}). \tag{14}$$

This discount is triggered only when the winning cluster is small and narrowly ahead of a larger runner-up:

$$|\mathbb{C}_{\text{win}}| \le \tau_{\text{size}} \ \wedge\ |\mathbb{C}_{\text{run}}| > |\mathbb{C}_{\text{win}}| \ \wedge\ s_{\text{comb}}(\mathbb{C}_{\text{win}}) - s_{\text{comb}}(\mathbb{C}_{\text{run}}) < \tau_{\text{gap}}, \tag{15}$$

rather than applied across all rounds; discounting every cluster's shape score by size would also penalize truly small malicious groups, which the Cross-client branch is designed to detect (Sec. 3.3.2). When triggered, s_{comb} is recomputed for *every* candidate cluster using $\text{shape}_{\text{scaled}}$ in place of shape, and $\mathbb{C}_{\text{win}}$ is replaced by $\mathbb{C}_{\text{run}}$ if its rescaled score now exceeds that of the original winner. We use $s_{\text{cap}} = 5$, $\tau_{\text{size}} = 2$, and $\tau_{\text{gap}} = 0.5$.

Expansion. After gating, we search among all remaining candidate clusters, which are not restricted to a local neighborhood, for the one most similar to the (possibly updated clusters from gating) winner:

$$\mathbb{C}^\star = \arg\max_{\mathbb{C}' \neq \mathbb{C}_{\text{win}}} \text{affinity}(\mathbb{C}_{\text{win}}, \mathbb{C}'), \qquad \text{affinity}(\mathbb{C}, \mathbb{C}') = \frac{1}{|\mathbb{C}||\mathbb{C}'|} \sum_{i\in\mathbb{C}} \sum_{j\in\mathbb{C}'} c_{ij}. \tag{16}$$

We merge $\mathbb{C}^\star$ into the winner if its affinity exceeds the winner's own isolation from the rest of the client population by a margin τ_{expand}:

$$\text{affinity}(\mathbb{C}_{\text{win}}, \mathbb{C}^\star) - \text{isolation}(\mathbb{C}_{\text{win}}) > \tau_{\text{expand}} \implies \mathbb{C}_{\text{win}} \leftarrow \mathbb{C}_{\text{win}} \cup \mathbb{C}^\star. \tag{17}$$

We use $\tau_{\text{expand}} = 0.4$. This margin condition is defined relative to the winner's own isolation rather than a fixed absolute threshold, so a candidate is merged only when it is closer to the winner than the winner is, on average, to the rest of the population, consistent with $\mathbb{C}^\star$ being a fragment of the same attacker group split off by HDBSCAN, rather than an unrelated benign cluster. At most one additional cluster is merged per round, and Expansion is applied only after Gating, using whichever cluster (original or gated) is the current winner.

Scope. Gating guards against a small but unstable cluster winning by chance; Expansion guards against a genuine attacker group being fragmented into multiple subclusters by HDBSCAN's density-based partitioning. Expansion is not a mechanism for detecting a second, independent malicious group with a different update direction: if two malicious groups are truly well separated in the spectral space, their affinity will be low and Expansion will not merge them.

A.5 DUALSPEC COMPLEXITY ANALYSIS

We analyze the per-round computational cost of DualSpec's two branches and compare it against DnC (Shejwalkar & Houmansadr, 2021) and AlignIns (Xu et al., 2025b), the two baselines that similarly rely on spectral or anchor-based analysis of client updates. Let n denote the number of clients participating in a round, $d = \sum_{l \in \mathbb{L}} d_l$ the total dimension of the selected parameters, and $r = 8$ the number of retained spectral components (shared by both branches, Sec. 3.3.1 and 3.4.1). In our setting, n is on the order of tens of clients per round ($n = 20$ in our experiments) while d reaches into the millions, so $n \ll d$ throughout.

Model-wise Cross-client branch. Per-layer normalization and concatenation (Eq. 1) cost $O(d)$ per client, $O(nd)$ overall. Forming the Gram matrix $\boldsymbol{G} = \boldsymbol{M}\boldsymbol{M}^\top$ costs $O(n^2 d)$, and its eigendecomposition costs $O(n^3)$ (Sup A.3), so the spectral embedding costs $O(n^2 d + n^3)$. HDBSCAN clustering over the n resulting r-dimensional points, and computing cohesion/isolation (Eqs. 2–3) over all candidate clusters, are both bounded by $O(n^2)$ since r is a fixed constant and disjoint candidate clusters together span at most n points. Cluster refinement (Sup A.4) adds only $O(k^2)$ for k candidate clusters, with $k \leq n$. The two-round extension (Sec. 3.3.3) applies the same procedure to $2n$ clients, changing only the constant factor. The branch is therefore dominated by

$$O(n^2 d + n^3). \tag{18}$$

Layer-wise Trajectory-residual branch. For each layer l, we only need the top r singular vectors of the global trajectory $\Delta_g^{t-1,l}$, computed via truncated SVD at cost $O(d_l r)$ rather than the full $O(d_l^2 \max(c_l, f_l))$ decomposition; summed over layers this is $O(dr)$, and is computed once per round independently of n. Projecting each client's per-layer update onto the reference subspace and computing the residual (Eq. 11) costs $O(d_l r)$ per client per layer, giving $O(ndr)$ over all clients and layers. The robust z-score over client scores (Sec. 3.4.2) adds $O(n \log n)$. The branch is therefore dominated by

$$O(ndr) = O(nd), \tag{19}$$

since r is a fixed constant.

Overall per-round cost. Combining both branches gives $O(n^2 d + n^3 + ndr)$. Since $r = O(1)$ and $n \geq 2$, the $n^2 d$ term dominates ndr, and since $n \ll d$ in our setting, $n^2 d$ also dominates n^3 by several orders of magnitude ($n^3 \approx 8{,}000$ versus $n^2 d \gtrsim 4 \times 10^8$ at $n = 20$ and $d \sim 10^6$). The overall cost is therefore effectively $O(n^2 d)$, attributable to the Cross-client branch's Gram matrix construction. Since n is a fixed hyperparameter set by the cross-device sampling budget rather than a quantity that scales with model size, this overall per-round cost is effectively *linear in the model dimension* d for fixed n: the n^2 and n^3 factors act only as constants with respect to d, matching the linear-in-d scaling of DnC ($O(bnd)$) and AlignIns ($O(nd)$).

Comparison with DnC and AlignIns. DnC estimates the projection onto the leading singular direction via power iteration over b bootstrap subsamples, at cost $O(bnd)$, linear in d, and cheaper than the Cross-client branch's $O(n^2 d + n^3)$, since it avoids forming the full $n \times n$ Gram matrix, retains only a single component rather than $r = 8$, and performs no clustering. AlignIns computes an anchor-based directional distance for each client and, with its Top-k selection procedure, has an overall complexity of $O(nd \log d)$. In the typical cross-device FL regime where $n \ll d$, the $O(n^2 d)$ term can remain much smaller than the $O(nd \log d)$ cost of AlignIns, while the Trajectory-residual branch itself requires only $O(nd)$ computation. Thus, DualSpec can achieve favorable computational scaling relative to AlignIns in high-dimensional FL settings.

A.6 ATTACKS AND DEFENSES BASELINES

Attack baselines. We evaluate DualSpec against six representative backdoor attacks covering diverse attack mechanisms and levels of adversarial sophistication. BadNets (Gu et al., 2017) serves as a canonical fixed-trigger attack, while DBA (Xie et al., 2019) distributes different trigger components across malicious clients to form a global backdoor. Chameleon (Dai & Li, 2023) represents a semantic backdoor setting, whereas Neurotoxin (Zhang et al., 2022) selectively manipulates less influential parameters to preserve the backdoor while reducing interference with benign training.

We further include A3FL (Zhang et al., 2023), which employs an adaptive trigger mechanism; and BATMAN (He et al., 2026) (CVPR 2026), an attack that exploits spectral information to construct stealthy malicious updates.

For ablation studies, we use a reduced attack set consisting of BadNets, Chameleon, DBA, and Neurotoxin, chosen to retain representative and widely studied attack behaviors while keeping the ablation matrix tractable.

Defense baselines. We compare DualSpec with representative defenses spanning the three defense categories discussed in Sec. 2. The main comparison includes DnC (Shejwalkar & Houmansadr, 2021), Multi-Krum (Blanchard et al., 2017), FLAME (Nguyen et al., 2022), FLTrust (Cao et al., 2021), Indicator (Li & Dai, 2024), AlignIns (Xu et al., 2025b), and FedDLAD (Ding et al., 2025), together with FedAvg as the undefended aggregation baseline. This selection includes several recent baselines: AlignIns (CVPR 2025), FedDLAD (IJCAI 2025), and Indicator (USENIX 2024). DnC and FLAME are technically related to DualSpec, as discussed in Sec. 2.

For ablation studies, we use a reduced defense set selected for strong overall performance in the main evaluation while maintaining coverage across the three defense categories: FedAvg, AlignIns, DnC, Indicator, and Multi-Krum.

A.7 Implementation Details

Federated learning setup. All experiments use $N = 100$ total clients with $M = 20$ clients sampled uniformly at random each round. Table 4 reports the remaining training hyperparameters, which differ between CIFAR10 and CIFAR100.

Table 4: Federated learning and local training hyperparameters.

Hyperparameter	CIFAR10	CIFAR100
Communication rounds	150	200
Local training epochs	5	3
Batch size	64	128
Local learning rate	0.05	0.1

Baseline defense hyperparameters. Table 5 reports the hyperparameters used for each baseline defense. We reference hyperparameters from each baseline's original work and adjust them to fit our standard setting, selecting values that give the best balance between MA and ASR. Multi-Krum's K and M denote the number of closest neighbors used in the Krum score and the number of selected updates, respectively.

Table 5: Baseline defense hyperparameters on CIFAR10 and CIFAR100.

Defense	Hyperparameter	CIFAR10	CIFAR100
Multi-Krum	K	10	12
	M	12	12
DnC	niters	7	8
	c	1.5	1.3
AlignIns	λ_c	1.5	1.5
	λ_s	1.5	1.5
	sparsity	0.3	0.3
FedDLAD	b_g	12	10
	p_g	4	3
	IQR scale	1.6	1.2
FLAME	noise σ	0.001	0.001
FLTrust	root samples	1000	1000
	server epochs	1	1

DualSpec hyperparameters. Unless otherwise noted, DualSpec uses the same hyperparameters on both datasets: HDBSCAN minimum cluster size 2, spectral rank $r = 8$ for both branches, Trajectory-residual threshold $\lambda = 2.0$, cluster refinement $s_{\text{cap}} = 5$, $\tau_{\text{size}} = 2$, $\tau_{\text{gap}} = 0.5$, $\tau_{\text{expand}} = 0.4$, cross round bonus weight $w_{\text{cross}} = 0.15$, and the two-round extension (Sec. 3.3.3) enabled. As discussed in Sec. 3.3.2, the shape score weight w_{shape} is dataset-dependent and set to 0.7 on CIFAR10 and 0.9 on CIFAR100 for the main results in Table 1, consistent with the ablation in Table 10.

B Additional Experimental Results

B.1 Evaluation under IID and heavy non-IID setting

We evaluate DualSpec and baseline defenses under both IID and non-IID client data distributions. Table 6 reports results under the IID partition, and Table 7 reports results under a more heterogeneous non-IID setting with Dirichlet $\alpha = 0.1$. DualSpec achieves the best or second-best MA, ASR, and MER in most attack-defense pairs across both settings, though all defenses, including DualSpec, show some degradation under the more heterogeneous $\alpha = 0.1$ partition.

Table 6: Detailed performance under the IID setting on CIFAR10.

Defense	BadNets			Neurotoxin			Chameleon		
	MA (↑)	ASR (↓)	MER (↑)	MA (↑)	ASR (↓)	MER (↑)	MA (↑)	ASR (↓)	MER (↑)
Multi-Krum	88.85	0.71	95.79	89.28	0.67	92.34	88.44	1.26	25.67
DnC	87.62	0.71	96.75	89.46	2.88	84.90	89.51	91.27	31.81
AlignIns	87.71	0.71	96.69	89.55	0.71	97.29	89.48	0.61	97.74
Indicator	86.41	1.91	98.04	88.11	34.17	85.73	86.22	97.37	70.57
DualSpec	**90.31**	**0.48**	**100**	**90.08**	**0.60**	**99.08**	**89.63**	**0.50**	**100.00**

Table 7: Detailed performance under the more heterogeneous non-IID setting with Dirichlet $\alpha = 0.1$ on CIFAR10

Defense	BadNets			Neurotoxin			Chameleon		
	MA (↑)	ASR (↓)	MER (↑)	MA (↑)	ASR (↓)	MER (↑)	MA (↑)	ASR (↓)	MER (↑)
Multi-Krum	46.25	50.90	6.12	61.01	49.20	12.07	59.46	39.88	15.64
DnC	56.52	2.98	75.00	57.42	37.21	9.50	64.30	5.40	21.08
AlignIns	58.64	76.55	45.41	63.61	4.34	70.06	67.51	21.84	39.62
Indicator	38.40	98.64	59.01	53.93	0.96	66.83	39.58	12.05	**84.52**
DualSpec	**71.05**	**2.64**	**86.57**	**63.89**	**0.46**	**95.23**	**72.23**	**5.27**	82.68

B.2 Evaluation with Blended Triggers

We evaluate DualSpec and baseline defenses using the blended trigger (Chen et al., 2017), defined as $X_{\text{poison}} = (1 - \alpha_{blended}) \cdot X_{\text{clean}} + \alpha_{blended} \cdot T$, where T is a fixed 32×32 random noise image and $\alpha_{blended} = 0.1$. Table 8 reports results on CIFAR10. DualSpec maintains low ASR and high MER across all three attacks, indicating that its detection remains effective when the trigger departs from the fixed-patch triggers used in the main experiments.

B.3 Evaluation on Sentiment140 and FMNIST

To evaluate the generalization of DualSpec beyond CIFAR-based image datasets, we further conduct experiments on Sentiment-140 and Fashion-MNIST (FMNIST). Sentiment-140 is used to assess the effectiveness of DualSpec on a text classification task, while FMNIST evaluates its performance on a non-CIFAR grayscale image dataset.

For Sentiment-140, we use an Embedding → BiLSTM → Dropout → Classifier architecture, where the classifier consists of Linear-ReLU-Linear layers. The federated learning system contains

Table 8: Performance under blended-trigger attacks on CIFAR10

Defense	BadNets			Neurotoxin			Chameleon		
	MA (↑)	ASR (↓)	MER (↑)	MA (↑)	ASR (↓)	MER (↑)	MA (↑)	ASR (↓)	MER (↑)
FedAvg	85.08	78.92	0.00	**86.97**	40.10	0.00	**87.84**	81.30	0.00
Multi-Krum	86.88	0.71	85.59	84.49	1.70	69.90	82.19	4.41	16.51
DnC	86.95	0.08	97.08	86.19	26.17	37.70	87.13	52.33	40.12
AlignIns	82.73	0.08	91.30	85.49	0.62	89.69	86.04	4.01	87.76
Indicator	80.04	2.28	98.22	82.73	1.67	35.42	80.04	100.00	14.01
DualSpec	**86.99**	**0.03**	**100.00**	86.76	**0.06**	**98.97**	87.23	**1.03**	**92.33**

Table 9: Performance of DualSpec on FMNIST and Sentiment-140 under BadNets and Neurotoxin.

Dataset	Defense	BadNets			Neurotoxin		
		MA (↑)	ASR (↓)	MER (↑)	MA (↑)	ASR (↓)	MER (↑)
FMNIST	FedAvg	88.25	89.86	0.00	**89.37**	91.50	0.00
	DualSpec	**88.65**	**0.08**	**98.75**	89.36	**0.14**	**96.21**
Sentiment-140	FedAvg	76.60	98.30	0.00	77.44	42.37	0.00
	DualSpec	**79.11**	**0.09**	**100.00**	**78.83**	**0.08**	**98.88**

300 clients, with 60 clients participating in each round, including 20% malicious clients. We train the model for 150 rounds with a batch size of 64, two local epochs, and a learning rate of 0.001.

For FMNIST, we adopt a model consisting of two convolutional layers followed by two fully connected layers. The setting includes 100 clients, with 20 clients participating in each round and 20% malicious clients. We use 150 communication rounds, a batch size of 32, five local epochs, and a learning rate of 0.03. The results are reported in Table 9.

As shown in the table, DualSpec suppresses the ASR while maintaining high mitigation effectiveness across both datasets. These results demonstrate that DualSpec generalizes beyond CIFAR-based vision tasks to both text classification and grayscale image classification.

B.4 Ablation Study of the Shape Score Weight w_{shape}

We conduct an ablation study on the shape-score weight w_{shape} (Eq. 5), which balances the shape score against the dominant-spectral-direction magnitude in the combined cluster score. We evaluate $w_{\text{shape}} \in \{0.5, 0.7, 0.9\}$ on BadNets, DBA, and Chameleon, attacks that exhibit collaborative structure and are therefore detected by the Cross-client branch, which lets us isolate the effect of w_{shape} on this branch's detection quality. Table 10 reports results on CIFAR10 and CIFAR100.

Table 10: DualSpec ablation study of the shape score weight w_{shape} across datasets and attacks. Results are reported for BadNets, DBA, and Chameleon on CIFAR10 and CIFAR100.

Dataset	w_{shape}	BadNets			DBA			Chameleon		
		MA (↑)	ASR (↓)	MER (↑)	MA (↑)	ASR (↓)	MER (↑)	MA (↑)	ASR (↓)	MER (↑)
CIFAR10	0.5	87.42	0.86	99.68	87.02	**0.11**	99.66	87.01	0.95	97.95
	0.7	**87.59**	**0.64**	**100.00**	**87.44**	0.81	**99.82**	**87.42**	**0.63**	**100.00**
	0.9	87.13	0.88	98.71	86.72	4.05	98.46	87.39	0.78	99.03
CIFAR100	0.5	51.50	8.29	56.51	63.13	0.52	97.67	64.14	0.57	**99.38**
	0.7	**63.89**	0.16	97.82	63.61	**0.19**	97.92	61.92	0.32	96.67
	0.9	62.88	**0.13**	**97.94**	**64.13**	0.20	**98.28**	**64.37**	**0.24**	**99.38**

On CIFAR10, $w_{\text{shape}} = 0.7$ gives the best trade-off. On CIFAR100, by contrast, $w_{\text{shape}} = 0.9$ shows strong detection across all three attacks. We attribute this contrast to the larger number of classes and greater intra-class diversity in CIFAR100: when clients collude, their updates form a more separable spectral cluster, so the shape score, which captures cohesion and isolation, becomes a stronger and

more reliable signal, favoring a higher w_{shape}. On CIFAR10, clusters are less distinct, so a lower w_{shape}, which leans more on the dominant-spectral-direction magnitude, performs better. Across both datasets, however, w_{shape}'s contribution dominates, confirming that clustering quality remains driven by the cohesion-isolation mechanism rather than the dominant-direction magnitude alone.

B.5 Ablation Study of Cross-client Cluster Refinement

Table 11: Ablation study of DualSpec components under different malicious-client ratios. We evaluate the full model and variants without the two-round Cross-client detection (w/o 2-round), cluster gating (w/o Gating), and cluster expansion (w/o Expand) at malicious-client ratios of 20% and 30%.

Malicious Ratio (%)	Scenario	BadNets			DBA			Chameleon		
		CA (↑)	ASR (↓)	MER (↑)	CA (↑)	ASR (↓)	MER (↑)	CA (↑)	ASR (↓)	MER (↑)
20	Full	**87.59**	**0.64**	**100.00**	**87.44**	**0.81**	**99.82**	**87.42**	**0.63**	**100.00**
	w/o 2-round	87.05	0.97	98.61	87.30	1.31	99.49	87.30	2.65	97.93
	w/o Gating	87.23	0.93	99.31	86.75	2.07	**99.82**	87.36	1.84	99.13
	w/o Expansion	87.48	1.00	99.48	86.64	1.90	98.79	87.23	0.84	99.48
	w/o all	87.03	0.93	97.94	86.57	3.59	97.82	87.26	14.47	97.06
30	Full	**86.98**	**0.55**	**100.00**	**86.95**	**0.23**	**100.00**	**87.16**	0.78	**99.70**
	w/o 2-round	86.97	1.01	99.88	86.82	1.95	99.30	87.13	**0.76**	98.47
	w/o Gating	86.91	0.65	99.07	86.65	2.32	99.76	87.10	0.82	99.68
	w/o Expansion	86.72	0.81	99.30	86.34	1.57	99.65	86.78	1.45	99.06
	w/o all	86.95	1.40	97.09	86.54	2.64	99.00	86.85	12.07	97.07

We further evaluate the contribution of Cross-client branch cluster refinement by removing or modifying its components: the two-round extension (w/o 2-round), Gating (w/o Gating), and Expansion (w/o Expansion), and all three removed simultaneously (w/o all). We use BadNets, DBA, and Chameleon, which are all detected primarily by the Cross-client branch, providing a consistent setting for isolating the effect of cluster refinement. Table 11 reports results at malicious-client ratios of 20% and 30%.

Removing any single component causes a drop in MER, with the two-round extension contributing the most: w/o 2-round yields the largest MER drop among the three individual ablations across settings. Removing all three components (w/o All) causes an even larger drop than any individual removal, confirming that each mechanism contributes independently to detection quality.

B.6 Robustness to Poisoning Ratio

Table 12: DualSpec performance under varying poisoning ratios (0.3, 0.5, and 0.7) on CIFAR10.

Poisoning Ratio	BadNets			Neurotoxin			Chameleon		
	MA (↑)	ASR (↓)	MER (↑)	MA (↑)	ASR (↓)	MER (↑)	MA (↑)	ASR (↓)	MER (↑)
0.3	87.53	0.74	**100.00**	87.26	0.81	**98.98**	87.26	**0.62**	99.47
0.5	87.59	**0.64**	**100.00**	**87.89**	0.77	**98.98**	**87.42**	0.63	**100.00**
0.7	**87.73**	0.81	**100.00**	87.26	**0.74**	98.00	87.27	0.88	99.50
Range (Δ)	0.20	0.17	0.00	0.63	0.07	0.98	0.16	0.26	0.53

We evaluate DualSpec under poisoning ratios of 0.3, 0.5, and 0.7 on CIFAR10. As shown in Table 12, MA and ASR remain stable across poisoning ratios, while MER stays near 100% for most settings, indicating consistent detection performance as the poisoning ratio varies.