Trust-Aware Topology Learning for Dynamic Decentralized Federated Learning under Adversaries

Shubham Vaishnav*, Murtaza Rangwala*, Ali Beikmohammadi, Sindri Magnússon, and Rajkumar Buyya

***Abstract*—In dynamic mobile decentralized federated learning (DFL), where device connectivity varies over time with movement, failures, and adversarial behavior, an adversary can poison both the model updates and the *topology information* that devices use to decide whom to learn from. We present DMTT (Dynamic `MURMURA` with Trusted Topology), a decentralized personalized federated learning protocol built on `MURMURA`, a prior framework that uses evidential deep learning to assess peer model compatibility and down-weight distribution-mismatched collaborators. DMTT extends this foundation to time-varying graphs under topology-manipulation attacks. Each device maintains a confidence-weighted local topology view from direct link-reliability estimates, signed topology claims, witness corroboration, and a Beta-distributed source-trust model, and then aggregates only over a trust-screened collaborator set using a composite score that fuses model compatibility, topology trust, and link reliability. We prove that the induced screened mixing matrices confine Byzantine influence to a bounded residual $\delta_{\max}$ that vanishes under perfect screening, and we implement DMTT as a coordinator-free distributed framework in which each client runs as an independent ZeroMQ process synchronised by a shared wall-clock epoch. On UCI HAR and PAMAP2, each partitioned across 100 mobile clients with Dirichlet heterogeneity, DMTT sustains honest-node accuracy above 0.862 at every tested fraction from 10% to 80% on UCI HAR and above 0.829 on PAMAP2, essentially matching no-attack accuracy at low fractions and degrading gracefully to near-local-only performance at extreme fractions. Static and dynamic FedAvg collapse to chance at every fraction, and dedicated robust aggregators (Krum, BALANCE, UBAR) all fail to consistently beat a safe local-only baseline that forgoes collaboration entirely. DMTT is the only method that clears the local-only bar across both datasets at all tested fractions, demonstrating that trust-aware screening both resists poisoning and profits from honest peers. Empirically the surviving Byzantine aggregation weight is exactly zero across all runs, consistent with the $\delta_{\max} = 0$ condition of the theoretical analysis. The same protocol runs end-to-end on a coordinator-free distributed ZeroMQ backend that reproduces these results on real nodes from the Melbourne Research Cloud.**

***Index Terms*—Decentralized federated learning, mobile computing, Byzantine fault tolerance, dynamic topologies, trust management, peer-to-peer computing.**

I. Introduction

Federated learning (FL) lets many devices train a shared model together without ever sending their raw data to a central server. It is now used in mobile keyboards, healthcare, and industrial IoT. A core difficulty in any realistic FL deployment is that no two devices see the same kind of data: a smartwatch on one wearer records different activity patterns from one on another. A hospital sees a different patient mix from its neighbour, and so on. With this kind of *non-IID* (non-identically-distributed) data, a single global model rarely fits every device well. Each device benefits from a *personalized* model that reflects its own data, while still learning from peers whose data is related but not identical. Standard FL still relies on one central server to coordinate this, which becomes both a privacy bottleneck and a single point of failure. Decentralized federated learning (DFL) removes that server entirely: devices exchange and aggregate model updates directly with peers [1], [2], [3]. The system becomes more private and more resilient, but every device is now exposed to its peers. Without a central authority, each device must decide on its own which peers to learn from, even though any participant can in principle behave adversarially.

`MURMURA` [4] addresses this per-device decision problem under non-IID data. Its central observation is that not every peer is a useful collaborator. A peer trained on a very different distribution can actively hurt local performance, even if the peer is honest and well-trained. `MURMURA` calls this property *peer compatibility* and quantifies it from each device's local point of view. The device runs an incoming peer model on its own data and looks at *how* the peer model is uncertain about that data. Using evidential deep learning, it separates two kinds of uncertainty in the peer's predictions: uncertainty because the peer has never really seen data like this (a sign of incompatibility), and uncertainty because the data is just inherently tricky (which is not the peer's fault). Peers showing the first kind are down-weighted during aggregation. Peers showing the second kind are kept. Each round produces a fresh per-device compatibility score, so the set of useful collaborators adapts as training progresses.

However, `MURMURA` does not solve the systems-level question of how peers discover and maintain knowledge of a changing network topology in the first place. This gap becomes particularly sharp in dynamic DFL, where connectivity varies over time due to mobility, failures, or adversarial behavior. Existing `MURMURA` evaluations assume fixed graph structures and synchronous rounds. In contrast, dynamic topology works such as Soft-DSGD and ToLRDUL either rely on link reliability models that do not consider malicious topology reports or assume globally coordinated topology updates that are too strong for fully decentralized environments [4], [1], [5]. Therefore, there is a need for dynamic `MURMURA` with

*These authors contributed equally to this work.

M. Rangwala and R.Buyya are with the Quantum Cloud Computing and Distributed Systems (qCLOUDS) Laboratory at the University of Melbourne, Australia (e-mail: murtaza.rangwala@student.unimelb.edu.au; rbuyya@unimelb.edu.au).

S. Vaishnav, A. Beikmohammadi, and S. Magnússon are with the Department of Computer and System Science, Stockholm University, Sweden (e-mail: shubham.vaishnav@dsv.su.se; beikmohammadi@dsv.su.se; sindri.magnusson@dsv.su.se).

TABLE I
THREE-LAYER SEPARATION BETWEEN REACHABILITY, TOPOLOGY TRUST, AND COLLABORATION

Layer	Description
Direct reachability	Tracks which peers can be contacted in the current epoch based on local observations such as authenticated beacons and packet acknowledgments [1].
Topology trust	Maintains confidence in topology claims using signed reports, corroboration across multiple witnesses, historical consistency, and path diversity [6], [7].
Collaboration graph	Determines which reachable and trustworthy peers should influence local training, combining MURMURA model trust, topology trust, and communication cost [4], [2], [3].

adversary-aware topology trust.

Beyond practicality, there are also strong theoretical reasons why this gap has persisted. Nesterenko and Tixeuil show that under Byzantine faults, weak topology discovery requires at least $k+1$-connectivity, while strong topology discovery requires more than $2k+1$-connectivity to tolerate k Byzantine nodes [6].

These requirements make exact global topology agreement impractical for sparse wearable, IoT, or mobile FL deployments. As a result, the appropriate target is not exact topology discovery, but rather a confidence-weighted topology view: exact for direct neighbors, reasonably reliable for two-hop structure, and probabilistic beyond that.

This work proposes a decentralized personalized FL framework that extends MURMURA beyond a fixed-neighborhood setting. It maintains a confidence-weighted topology state built from direct observation, signed claims, corroboration, and uncertainty-aware source trust. It then uses this trusted graph to choose collaborators using evidential trust. The proposed system can be viewed as MURMURA augmented with a trustworthy topology memory. The design separates the problem into three distinct layers as shown in Table I.

a) Contributions.:

1) **DMTT protocol with convergence guarantees.** A four-algorithm protocol integrating link-reliability tracking, witness-weighted signed topology claims, Beta-distributed source trust, and trust-aware collaborator selection. We prove that the induced mixing matrices confine Byzantine influence to a bounded residual that vanishes under perfect screening.
2) **Fully distributed MURMURA framework.** MURMURA is redesigned from single-process simulation into independent ZeroMQ processes synchronised by a shared wall-clock epoch, with a passive PULL-only monitor collecting metrics without influencing training.
3) **Mobility-driven dynamic topology and topology-liar attack.** A deterministic random-walk mobility model that generates $\mathcal{G}^t$ identically across all nodes from a shared seed, plus an oracle-free self-inclusion check: if j delivered a message to i this round, j's topology claim must include i. We introduce an *omission* topology-liar attack in which Byzantine nodes claim only the Byzantine coalition, so the self-inclusion contradiction fires every round.
4) **Empirical validation.** On UCI HAR and PAMAP2 under combined topology-liar and Gaussian poisoning at Byzantine fractions of 10%–80%, DMTT attains the highest honest-node accuracy in every setting while non-robust baselines collapse to chance; a trust diagnostic confirms zero aggregation weight to Byzantine peers, instantiating the perfect-screening regime of the theory.

II. RELATED WORK

a) Decentralized and personalized federated learning.: Decentralized federated learning removes the central server and allows clients to exchange model updates over a peer-to-peer graph. Decentralized FedAvg and related local-update methods show that convergence and communication cost depend on the number of local updates, the frequency of communication, and the mixing matrix [8], [9]. This design is useful for edge and mobile systems because it avoids a server bottleneck and a single point of failure.

However, clients usually have non-IID data, so a fixed neighborhood is not always useful. Some decentralized personalized-learning methods therefore learn collaboration graphs or adaptive mixing weights from the relationships between clients and their local tasks [10], [11], [12]. Directed collaboration also allows clients with different data, computing, and communication conditions to use different peer sets [3]. MURMURA [4] follows a related model-level approach. Each client tests an incoming model on its own data and uses evidential uncertainty to identify peers whose data is too different from its own [13]. DMTT keeps this idea of client-specific model compatibility, but also considers whether the topology information used to select peers can be trusted.

b) Dynamic topologies and mobile decentralized learning.: Several studies examine decentralized learning when links are unreliable or the graph changes over time. Soft-DSGD uses link-dependent mixing weights to handle packet loss and other communication problems [1]. ToLRDUL selects links by considering data differences, representation differences, and unreliable D2D connections [5]. Koloskova et al. analyze local updates and changing mixing matrices in a common framework [14]. Other studies examine optimization over directed, time-varying graphs and show the importance of connectivity over time [15], [16]. For non-IID data, the choice of neighbors matters as well as the number of neighbors. Neighborhood heterogeneity can affect the convergence bound, and a carefully learned sparse graph can reduce this problem by connecting clients with complementary data [17].

Recent work has applied these ideas to mobile and asynchronous DFL. Lu et al. use the Metropolis–Hastings rule to update mixing weights when the communication graph changes. They also use secret sharing to protect model states. However, their threat model assumes semi-honest participants and does not cover active attacks [18]. DySTop controls update staleness and builds a changing topology for asynchronous DFL. Its topology construction considers non-IID data, delay, and communication cost [19]. Li et al. study DFL over time-varying and heterogeneous mobile computing networks. Their

analysis links neighborhood heterogeneity with the properties of the communication graph [20]. These studies are close to DMTT because they consider changing mobile graphs. However, they assume that topology and link information are honest. They do not consider a Byzantine client that hides honest neighbors, invents links, or sends false topology reports.

c) Byzantine resilience and trusted topology.: Another line of work studies Byzantine-resilient decentralized learning. Most of these methods assume that the communication graph is already known and focus on filtering model or gradient messages. BRIDGE studies Byzantine failures in directed peer-to-peer networks and gives convergence guarantees under graph connectivity conditions [21]. UBAR provides a Byzantine-resilient aggregation rule for decentralized learning. Iterative outlier scissor also shows that local robust aggregation can create disagreement and mixing errors if the resulting virtual mixing process is not properly controlled [22], [23]. Self-centered clipping shows that topology bottlenecks can make Byzantine attacks more harmful. It gives a convergence bound that depends on the remaining Byzantine influence and the graph structure [24]. BALANCE uses each client's local model as a reference for filtering poisoned peer models and provides convergence results for decentralized FL under poisoning attacks [25]. CMFL uses rotating committees to score local gradients and select updates in a serverless FL system [26]. Other work studies Byzantine resilience under heavy-tailed data and compressed communication, but it still uses a central server for aggregation [27].

These model-level defenses usually start after a peer has already entered the communication or aggregation neighborhood. A Byzantine client may therefore lie about reachability, adjacency, or witness information before its model update is filtered. Work on Byzantine topology discovery shows that exact global topology recovery requires strong connectivity assumptions. Weak discovery needs connectivity greater than the number of Byzantine faults, while strong discovery needs connectivity greater than twice that number [6]. This makes a confidence-weighted local topology view more realistic than a claim of perfect global knowledge. In low-power IoT routing, TRAIL uses reachability tests to detect topological attackers without relying on heavy cryptography [28]. In mobile ad hoc networks, reputation systems combine direct observations with recommendations from other nodes while reducing the effect of stale or inconsistent information [29]. Blockchain-based FL can provide a stronger audit trail by combining malicious-gradient detection, homomorphic protection, and blockchain records, but it also adds infrastructure and coordination costs [30].

DMTT combines these ideas through three separate decisions. Direct reachability determines which peers can exchange messages at the current time. Topology trust determines whether a peer's claims about the wider graph should affect peer selection. MURMURA model compatibility determines whether that peer is useful for local personalization. Dynamic-topology methods usually assume that the graph information is honest, while Byzantine aggregators usually filter only model updates. DMTT addresses both problems by screening topology claims before building the collaborator set and then screening the models of the remaining peers. This produces a screened mixing matrix whose Byzantine influence can be bounded while keeping the implementation fully distributed.

III. Problem Formulation

This section states the system, data, and adversary model, identifies the local state each client must maintain, and writes down the optimization problem that the protocol of Section IV solves. Concrete update rules and equations for each state variable are deferred to Section IV.

A. System and Communication Model

We consider a decentralized personalized FL system with a fixed set of clients $\mathcal{V} = \{1, 2, \ldots, N\}$. Client identities are persistent, but client positions vary over time. Let $\mathbf{r}_i^t$ denote the position of client i at epoch t. Mobility induces a time-varying communication topology $\mathcal{G}^t = (\mathcal{V}, \mathcal{E}^t)$, where $(i, j) \in \mathcal{E}^t$ if clients i and j are physically able to communicate at epoch t. The node set is fixed; the edge set changes due to motion, channel variation, or obstruction. This is the dynamic-topology extension beyond the fixed-neighborhood setting of `MURMURA` [4], and it follows the standard time-varying graph model used in decentralized optimization [1], [5].

B. Local Data and Personalized Objective

Each client $i \in \mathcal{V}$ holds a private dataset $\mathcal{D}_i = \mathcal{D}_i^{tr} \cup \mathcal{D}_i^{val}$ drawn from a client-specific distribution P_i. The distributions $\{P_i\}$ are generally non-IID, so the system seeks not a single global model but a collection of personalized models $\Theta = \{\theta_1, \ldots, \theta_N\}$ with $\theta_i \in \mathbb{R}^d$, following the personalization principle of `MURMURA` [4]. The local population risk at client i is

$$F_i(\theta_i) = \mathbb{E}_{(x,y)\sim P_i}\big[\ell(f(x;\theta_i), y)\big],$$

and the population objective is

$$\min_{\Theta} \sum_{i=1}^{N} F_i(\theta_i).$$

Unlike conventional DFL, client i cannot safely collaborate with every other client at every epoch: communication feasibility and topology trust are both time-varying.

C. Adversary Model

Let $\mathcal{H} \subseteq \mathcal{V}$ denote the honest clients and $\mathcal{B} = \mathcal{V} \setminus \mathcal{H}$ the Byzantine clients. The Byzantine set is fixed over the training horizon, although Byzantine clients move like honest ones. Honest clients follow the prescribed protocol. Byzantine clients may behave arbitrarily, including issuing false edge-addition or edge-removal claims, replaying stale topology states, flooding the network with weak claims, or sending misleading model updates. The formulation therefore separates topology trust from model usefulness so that topology misinformation is filtered before collaboration decisions are made.

D. Local State to Be Maintained

At every epoch t, each client i must maintain four categories of local state, whose precise update rules are given in Section IV:

- *Direct reachability and link reliability.* A one-hop neighbor set $N_i^{dir,t}$ from authenticated hello beacons and acknowledgments, and a scalar link-reliability estimate $\hat{p}_{ij}^t \in [0,1]$ tracking communication quality (*not* topology trust) [1].
- *Signed topology claims and witness evidence.* Each locally observed topology event is emitted as a signed, versioned claim; received claims are collected in per-edge support and opposition buffers $\mathcal{W}_i^{+,t}(e), \mathcal{W}_i^{-,t}(e)$, which the protocol summarizes into an edge-confidence score $\chi_i^t(e) \in [0,1]$.
- *Source-trust state.* For each peer j, a Beta-style evidence state $(\alpha_{ij}^t, \beta_{ij}^t)$ tracks the long-run reliability of j as a topology reporter, from which the protocol derives a reputation R_{ij}^t, an epistemic uncertainty U_{ij}^t, and a topology-trust score $T_{ij}^{topo,t} \in [0,1]$ (the topology-side analogue of the model-side evidential trust of MURMURA [4]).
- *Trusted local topology view.* By accepting an edge e only when $\chi_i^t(e)$ and corroboration are sufficient, client i maintains a confidence-weighted local view $\hat{\mathcal{G}}_i^t = (\mathcal{V}, \hat{\mathcal{E}}_i^t)$. The target is not exact global topology discovery (which would require $(2k+1)$-connectivity under k Byzantine faults [6]) but a locally sufficient view for safe collaboration.

E. Trusted Feasible Set and Effective Learning Graph

Given the local state above, the protocol must induce, at every epoch, a *trusted feasible neighbor set* $\mathcal{F}_i^t \subseteq N_i^{dir,t}$ restricted to peers that are simultaneously reachable (sufficient $\hat{p}_{ij}^t$) and trustworthy as topology reporters (sufficient $T_{ij}^{\mathrm{topo},t}$), and an *active collaborator set* $C_i^t \subseteq \mathcal{F}_i^t$ satisfying $|C_i^t| \le B$ within an edge budget B, as in budgeted decentralized personalized graph learning [2]. The corresponding aggregation weights $\{w_{ij}^t\}$ define a row-stochastic, time-varying *screened mixing matrix* $W^t = [w_{ij}^t]_{i,j=1}^N$ with

$$w_{ij}^t = 0 \text{ if } j \notin C_i^t \cup \{i\}, \quad \sum_j w_{ij}^t = 1, \quad w_{ij}^t \ge 0.$$

Hence the effective learning graph is not the raw physical graph $\mathcal{G}^t$ but a client-specific, time-varying, trust-screened graph. This connects the formulation to standard analyses of decentralized learning over time-varying graphs, where topology enters through the support and spectral properties of the mixing matrices.

F. Optimization Problem

The problem addressed in this work is to jointly design four protocol rules:

1) the topology-claim processing rule, producing $\hat{\mathcal{G}}_i^t$;
2) the source-trust update rule, producing $T_{ij}^{topo,t}$;
3) the trusted-feasible-neighbor selection rule, producing $\mathcal{F}_i^t$ and C_i^t;
4) the personalized aggregation rule, producing W^t and θ_i^{t+1}.

Over a horizon of T epochs, the goal is to minimize the time-averaged aggregate personalized risk

$$\min_{\{\theta_i^t\},\, \Pi_{topo},\, \Pi_{collab}} \ \frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{N} \mathbb{E}\big[F_i(\theta_i^t)\big],$$

subject, for every epoch t and client i, to

$$C_i^t \subseteq \mathcal{F}_i^t \subseteq N_i^{dir,t}, \qquad |C_i^t| \le B,$$

the row-stochastic mixing-matrix constraints above, and the requirement that topology information may influence collaboration only through the screened local topology state $(\hat{\mathcal{G}}_i^t, T_{ij}^{topo,t})$.

In words, the proposed method, Dynamic MURMURA with Trusted Topology (DMTT), must maintain a trustworthy, confidence-weighted topology view and perform personalized decentralized learning only over the resulting trusted feasible graph. Section IV specifies the four rules above and gives the equations defining $\hat{p}_{ij}^t$, $\chi_i^t(e)$, $T_{ij}^{topo,t}$, and W^t.

IV. The DMTT Protocol

The proposed method is a decentralized training protocol with an explicit topology-trust layer. Instead of trusting a topology claim on arrival, each node first grounds its network view in direct neighbor observations, then verifies signed claims for freshness and authenticity, accumulates corroborating and opposing witness evidence, updates long-term trust in each reporting source, and limits forwarding through trust thresholds, hop limits, and rate budgets. Only after this topology-filtering stage does the node invoke MURMURA-style model trust to decide which reachable peers should influence training.

The protocol contains six logical steps, but for clarity they are grouped into four algorithmic modules. Algorithm 1 integrates direct reachability estimation, signed topology-claim generation, local model training, model exchange, and collaborator selection. Topology-claim processing and edge-confidence computation are handled by Algorithm 2. Controlled forwarding is handled by Algorithm 3. Source-trust updating is handled by Algorithm 4.

Local State at Node i: Each node i maintains local state comprising direct link quality, topology trust, witness evidence, and model-level collaboration scores. Algorithm 1 gives the overall decentralized protocol, executed locally and in parallel at every epoch.

A. Direct Reachability and Signed Topology Claims

The protocol starts from direct observations rather than an assumed global topology. As shown in Algorithm 1, each node discovers direct neighbors through authenticated hello beacons and transport-level acknowledgments. For a direct link (i,j), node i updates a local reliability score in the spirit of unreliable link estimation in Soft-DSGD, but without assuming a global reliability matrix [1]:

$$\hat{p}_{ij}^{t+1} = (1-\rho)\hat{p}_{ij}^t + \rho \cdot \mathrm{ack}_{ij}^t. \tag{1}$$

Here, $\rho \in (0,1)$ is a smoothing parameter and $\mathrm{ack}_{ij}^t \in \{0,1\}$ records recent communication success. This quantity

Algorithm 1 DMTT: Dynamic MURMURA with Trusted Topology

Initialize local model θ_i^0
Initialize direct-neighbor set N_i^{dir}
Initialize trust state $\{\alpha_{ij}, \beta_{ij}, T_{ij}^{topo}\}_j$
Initialize witness buffers $\mathcal{W}_i(e)$ and edge confidences $\chi_i(e)$
Initialize collaborator set C_i
for each epoch t executed in parallel at all nodes **do**
  for each authenticated neighbor j heard in epoch t **do**
    Add j to $N_i^{dir,t}$
    $\hat{p}_{ij}^{t+1} \leftarrow (1-\rho)\hat{p}_{ij}^t + \rho \cdot \mathrm{ack}_{ij}^t$
  end for
  Remove stale neighbors from $N_i^{dir,t}$ if timeout expires
  if node i observes addition or removal of edge $e = (u, v)$ **then**
    Construct signed claim $c \leftarrow (u, v, \sigma, \nu, \tau, \pi_u)$
    Broadcast c to current direct neighbors
  end if
  for each received claim c about edge e **do**
    ProcessTopologyClaim(i, c, e, t)
  end for
  for each source j with new evidence **do**
    UpdateSourceTrust(i, j, t)
  end for
  for each received claim c **do**
    ControlledForward(i, c, t)
  end for
  Perform E local SGD steps to obtain $\theta_i^{t+\frac{1}{2}}$
  Exchange candidate models or updates with currently reachable, topology-screened neighbors
  Evaluate received peer models on local validation data
  Compute validation accuracy a_{ij}^t and mean epistemic uncertainty $\bar{u}_{ij}^t$
  Compute MURMURA-style compatibility scores $s_{ij}^{model,t}$
  for each currently reachable and topology-screened neighbor j **do**
    $q_{ij}^t \leftarrow \lambda_1 s_{ij}^{model,t} + \lambda_2 T_{ij}^{topo,t} + \lambda_3 \hat{p}_{ij}^t - \lambda_4 c_{ij}^{comm,t}$
  end for
  $C_i^t \leftarrow \mathrm{TopB}_j \; q_{ij}^t$
  Aggregate over C_i^t and update θ_i^{t+1}
end for

captures link quality, not trust [1]. It provides the local connectivity view on which the later filtering stages build.

When a node observes a local topology event, it emits the signed claim c defined in Section III-D. For direct-neighbor claims, the preferred form is bilateral attestation, meaning both endpoints sign the same event when possible. One-sided claims are allowed but start with lower confidence. Each node may also periodically send a compact signed digest of its current neighbor list (similar to sketching [31]).

B. Topology Claim Processing and Edge Confidence

Received topology claims are not committed immediately. Instead, as shown in Algorithm 2, node i stores each claim in a witness buffer and treats it as weighted evidence about the reported edge. This is the primary defense against malicious topology gossip.

Algorithm 2 Topology claim processing and edge-confidence update

Extract source j and hop count $h_{ij}^t(e)$ from c
if signature invalid or version stale **then**
  Reject c and return
end if
Add source j to witness buffer $\mathcal{W}_i(e)$
$\psi_{ij}^t(e) \leftarrow T_{ij}^{topo,t} \exp(-\xi h_{ij}^t(e))$
if c supports existence of edge e **then**
  Add $\psi_{ij}^t(e)$ to support pool
else
  Add $\psi_{ij}^t(e)$ to opposition pool
end if
$S_i^{+,t}(e) \leftarrow \sum_{j \in \mathcal{W}_i^{+,t}(e)} \psi_{ij}^t(e)$
$S_i^{-,t}(e) \leftarrow \sum_{j \in \mathcal{W}_i^{-,t}(e)} \psi_{ij}^t(e)$
$\chi_i^t(e) \leftarrow \dfrac{S_i^{+,t}(e)}{S_i^{+,t}(e) + S_i^{-,t}(e) + \varepsilon}$
if c is an edge addition **then**
  if $\chi_i^t(e) \geq \tau_{add}$ and witness diversity conditions hold **then**
    Accept edge addition
  end if
else
  if $1 - \chi_i^t(e) \geq \tau_{add}$ and witness diversity conditions hold **then**
    Accept edge removal
  end if
end if

When node i receives a topology claim from source j, it computes the witness weight

$$\psi_{ij}^t(e) = T_{ij}^{\mathrm{topo},t} \exp\big(-\xi h_{ij}^t(e)\big) \cdot \mathbf{1}\{\text{valid signature}\} \cdot \mathbf{1}\{\text{fresh version}\}. \tag{2}$$

Here, $T_{ij}^{\mathrm{topo},t}$ is trust in source j, $h_{ij}^t(e)$ is the hop count traveled by the claim before reaching i, and ξ penalizes distant claims. Thus, a witness counts more if the source is trusted, the claim traveled fewer hops, the signature is valid, and the claim is fresh. Signed stale claims are ignored.

For each edge e, node i maintains separate support and opposition totals:

$$S_i^{+,t}(e) = \sum_{j \in \mathcal{W}_i^{+,t}(e)} \psi_{ij}^t(e), \tag{3}$$

$$S_i^{-,t}(e) = \sum_{j \in \mathcal{W}_i^{-,t}(e)} \psi_{ij}^t(e). \tag{4}$$

The opposition total is therefore the weighted sum of all witnesses that argue against the edge claim. From these totals, node i defines edge confidence as

$$\chi_i^t(e) = \frac{S_i^{+,t}(e)}{S_i^{+,t}(e) + S_i^{-,t}(e) + \varepsilon}. \tag{5}$$

Algorithm 3 Controlled forwarding

Extract source j from c
if signature invalid or version stale **then**
    Drop c
else if $T_{ij}^{topo,t} < \tau_{fwd}$ **then**
    Drop c
else if forwarding budget for source j exhausted **then**
    Drop c
else if hop count of $c \geq H$ **then**
    Drop c
else
    Forward c to eligible direct neighbors
    Decrement forwarding budget for source j
end if

Algorithm 4 Source-trust update

Observe evidence events $d_{ij}^t, c_{ij}^t, f_{ij}^t, x_{ij}^t, q_{ij}^t, z_{ij}^t$
$\alpha_{ij}^{t+1} \leftarrow \lambda\alpha_{ij}^t + w_d d_{ij}^t + w_c c_{ij}^t + w_f f_{ij}^t$
$\beta_{ij}^{t+1} \leftarrow \lambda\beta_{ij}^t + w_x x_{ij}^t + w_q q_{ij}^t + w_s z_{ij}^t$
$R_{ij}^{t+1} \leftarrow \dfrac{\alpha_{ij}^{t+1}}{\alpha_{ij}^{t+1} + \beta_{ij}^{t+1}}$
$U_{ij}^{t+1} \leftarrow \sqrt{\dfrac{\alpha_{ij}^{t+1}\beta_{ij}^{t+1}}{(\alpha_{ij}^{t+1} + \beta_{ij}^{t+1})^2(\alpha_{ij}^{t+1} + \beta_{ij}^{t+1} + 1)}}$
$T_{ij}^{topo,t+1} \leftarrow R_{ij}^{t+1}\exp\big(-\eta\max\{0, U_{ij}^{t+1} - \tau_U\}\big)$

An edge addition is accepted only if three conditions hold:

- $\chi_i^t(e) \geq \tau_{add}$, where $\tau_{add} \in (0, 1]$ is the *edge-acceptance threshold*,
- support comes from at least m distinct first-hop witnesses,
- in high-assurance mode, support arrives through at least $k + 1$ internally node-disjoint paths when the graph is connected enough to make that meaningful [6].

An edge removal uses the symmetric rule with $1-\chi_i^t(e)$. Thus, a single liar may inject a claim, but that claim is not committed unless enough independent, trusted evidence accumulates.

C. Controlled Forwarding

Even a locally observed claim should not be forwarded without restriction. As shown in Algorithm 3, a node forwards a topology claim only if the signature is valid, the version is newer than the local record, the trust score satisfies $T_{ij}^{\text{topo},t} \geq \tau_{\text{fwd}}$ (the *forwarding trust threshold*), the forwarding budget for that source has not been exceeded [32], and the claim has not crossed a hop limit H.

The forwarding budget is important. Gupta et al. [32] show that rate limiting and verification matter even before one considers learning quality. In the present design, repeated flooding of unconfirmed or stale claims contributes negative evidence in the trust update, so forwarding control not only limits immediate spread but also reduces the long-run influence of abusive sources.

D. Source Trust Update

The claim-processing stage in Algorithm 2 relies on the current trust score $T_{ij}^{\text{topo},t}$, so this quantity must be updated explicitly over time. As shown in Algorithm 4, node i maintains trust in each source j with a Beta-style evidence tracker.

Positive evidence for j comes from three events: directly observed claims from j about incident edges later match local handshakes, claims from j are corroborated by independent trusted witnesses, and claims from j remain fresh and internally consistent over time. Negative evidence comes from three events: a claim from j is contradicted by later direct observation, a claim from j is contradicted by a trusted quorum, or j floods stale or conflicting topology updates.

A concrete update is

$$\alpha_{ij}^{t+1} = \lambda\alpha_{ij}^t + w_d d_{ij}^t + w_c c_{ij}^t + w_f f_{ij}^t, \tag{6}$$

$$\beta_{ij}^{t+1} = \lambda\beta_{ij}^t + w_x x_{ij}^t + w_q q_{ij}^t + w_s z_{ij}^t, \tag{7}$$

where $\lambda \in (0, 1]$ is a forgetting factor, d is direct confirmation, c is corroboration, f is freshness consistency, x is contradiction, q is spam or rate violation, and z is severe conflict with the eventually committed local view.

The posterior mean reputation is

$$R_{ij}^{t+1} = \frac{\alpha_{ij}^{t+1}}{\alpha_{ij}^{t+1} + \beta_{ij}^{t+1}}. \tag{8}$$

The epistemic uncertainty of this reputation is taken from the Beta variance:

$$U_{ij}^{t+1} = \sqrt{\frac{\alpha_{ij}^{t+1}\beta_{ij}^{t+1}}{(\alpha_{ij}^{t+1} + \beta_{ij}^{t+1})^2(\alpha_{ij}^{t+1} + \beta_{ij}^{t+1} + 1)}}. \tag{9}$$

The final topology trust score mirrors `MURMURA`'s uncertainty penalization principle [4]: the reputation mean is penalized when epistemic uncertainty is high,

$$T_{ij}^{\text{topo},t+1} = R_{ij}^{t+1} \cdot \exp\big(-\eta\max\{0, U_{ij}^{t+1} - \tau_U\}\big). \tag{10}$$

A node is trusted not only when its claims are often correct, but when that correctness is established with enough evidence. A malicious node that alternates between truthful and false claims may retain a moderate mean reputation, but its uncertainty remains high, keeping its topology influence small.

E. Distributed Local Training and Trust-Aware Collaborator Selection

The previous modules determine which topology information is credible and which neighbors remain eligible for collaboration. The learning procedure itself is decentralized and runs in parallel across nodes. As shown in Algorithm 1, each node performs local training at every epoch, exchanges models or updates only with currently reachable neighbors that survive topology filtering, and then selects a sparse collaborator set for personalized aggregation.

At epoch t, node i first performs E local optimization steps on its own data to obtain an intermediate model

$$\theta_i^{t+\frac{1}{2}} = \text{LocalUpdate}(\theta_i^t, \mathcal{D}_i, E), \tag{11}$$

where $\mathcal{D}_i$ is the local dataset. After topology filtering, node i exchanges candidate models or updates only with currently reachable and topology-screened neighbors. For those

neighbors, it computes the MURMURA model compatibility score $s_{ij}^{\text{model},t}$ from local validation accuracy and epistemic uncertainty [4]. Let a_{ij}^t denote the validation accuracy of peer j's model on node i's local validation data, and let $\bar{u}_{ij}^t$ denote the corresponding mean epistemic uncertainty. Following MURMURA, the epistemic uncertainty for a K-class evidential model is

$$u = \frac{K}{S}, \tag{12}$$

where $S = \sum_{k=1}^{K} \alpha_k$ is the Dirichlet strength. Averaging this quantity over node i's validation samples for peer j gives $\bar{u}_{ij}^t$.

The base compatibility score is then defined as

$$s_{ij,\text{base}}^t = \left(1 - \bar{u}_{ij}^t\right)\left(w_a a_{ij}^t + (1 - w_a)\right), \tag{13}$$

where $w_a \in [0,1]$ controls the contribution of validation accuracy.

The final MURMURA-style compatibility score applies an uncertainty penalty when the mean uncertainty exceeds a threshold τ_u:

$$s_{ij}^{\text{model},t} = \begin{cases} s_{ij,\text{base}}^t \exp\left(-(\bar{u}_{ij}^t - \tau_u)\right), & \bar{u}_{ij}^t > \tau_u, \\ s_{ij,\text{base}}^t, & \bar{u}_{ij}^t \le \tau_u. \end{cases} \tag{14}$$

The final collaboration score is

$$q_{ij}^t = \lambda_1 s_{ij}^{\text{model},t} + \lambda_2 T_{ij}^{\text{topo},t} + \lambda_3 \hat{p}_{ij}^t - \lambda_4 c_{ij}^{\text{comm},t}. \tag{15}$$

The collaborator set is then

$$C_i^t = \text{TopB}_j \; q_{ij}^t, \tag{16}$$

subject to an edge budget B, as in budgeted decentralized personalized graph learning [2]. Node i aggregates only the intermediate models or updates received from neighbors in C_i^t, together with its own model, using MURMURA-style trust-aware weighting:

$$\theta_i^{t+1} = w_{ii}^t \theta_i^{t+\frac{1}{2}} + \sum_{j \in C_i^t} w_{ij}^t \theta_j^{t+\frac{1}{2}}, \qquad \sum_{j \in C_i^t \cup \{i\}} w_{ij}^t = 1, \tag{17}$$

where the aggregation weights depend on the selected collaboration scores and the MURMURA trust-normalization rule. The collaborator set C_i^t in this final step is the trust-screened set: a candidate j enters C_i^t only if it passes the model-compatibility gate $s_{ij}^{\text{model},t} \ge \gamma\, s_{ii}^{\text{model},t}$, the topology-trust gate $T_{ij}^{\text{topo},t} \ge \tau_{\text{trust}}$, and a lightweight norm-sanity check $\|\theta_j^{t+\frac{1}{2}}\| \le \kappa_{\text{norm}} \|\theta_i^{t+\frac{1}{2}}\|$ that rejects large-norm model-poisoning outliers; combined with the bounded-domain Assumption 9, this gate directly enforces the bounded-Byzantine-norm condition of Assumption 8 with $\Theta_B = \kappa_{\text{norm}} R$. Crucially, this screening enters the *aggregation weights*, not merely the choice of which peers to contact: a peer that survives contact but fails screening receives weight zero, which is what drives the surviving Byzantine weight $\delta_{\max}$ to zero in practice.

This final stage keeps the topology and model layers distinct: $T_{ij}^{\text{topo},t}$ answers whether topology information from j should be believed, $\hat{p}_{ij}^t$ whether the link is currently usable, and $s_{ij}^{\text{model},t}$ whether j is useful for personalization. All nodes train in parallel, but only credible, reachable, and useful neighbors influence the local model.

V. Distributed System Implementation

The theoretical protocol described in Section IV requires a concrete execution substrate: independent client processes that exchange messages, maintain local state, and synchronise rounds without any central coordinator. This section describes the system architecture constructed to run DMTT experiments on real hardware, and to serve as a reusable open-source extension of the MURMURA framework.

A. Architecture Design Principles

Three principles guided our design.

*a) **No coordinator in the learning path.**:* A central coordinator introduces a single point of failure and contradicts the DFL threat model, since a compromised coordinator could suppress or fabricate round signals, undermining the decentralised guarantee. To eliminate this dependency, round k at node i begins at the absolute wall-clock time

$$t_k = t_{\text{start}} + k \cdot \Delta, \tag{18}$$

where t_{start} is a shared epoch announced once at experiment start and Δ is the round budget. Each node independently sleeps until t_k and then proceeds without waiting for any signal. A passive monitor, implemented as a PULL-only process, collects per-round accuracy and loss metrics without sending any messages. Because it has no send socket, it cannot influence the learning protocol, and its failure leaves training unaffected.

*b) **ZeroMQ P2P transport.**:* Model weights are exchanged peer-to-peer using ZeroMQ PUSH/PULL socket pairs. Each node i binds one PULL socket (its receiving endpoint) and maintains one PUSH socket per current collaborator. This design avoids the fixed process-group requirement of `torch.distributed` and permits the dynamic collaborator set C_i^t to change every round without re-initialisation. For single-machine experiments, all sockets use IPC transport; for multi-machine deployments, the same code switches to TCP by changing a single configuration field.

*c) **Deterministic shared mobility model.**:* The time-varying graph $\mathcal{G}^t$ must be consistent across all node processes without inter-process communication. This is achieved by defining $\mathcal{G}^t$ through a random-walk mobility model parameterised by a shared seed: given the same seed, every process computes identical node positions and therefore identical edge sets at every round.

B. ZeroMQ Socket Layout

Each node process uses three logical channels, as shown in Fig. 1.

- **PULL (bind):** Receives `MODEL_STATE` and `TOPO_CLAIM` messages from neighbours. A single socket handles both message types, distinguished by a 1-byte type field in the header.
- **PUSH $\times|C_i^t|$ (connect):** Sends `MODEL_STATE` and `TOPO_CLAIM` to each current collaborator. Sockets are created on first use so that the collaborator set can change without re-binding.

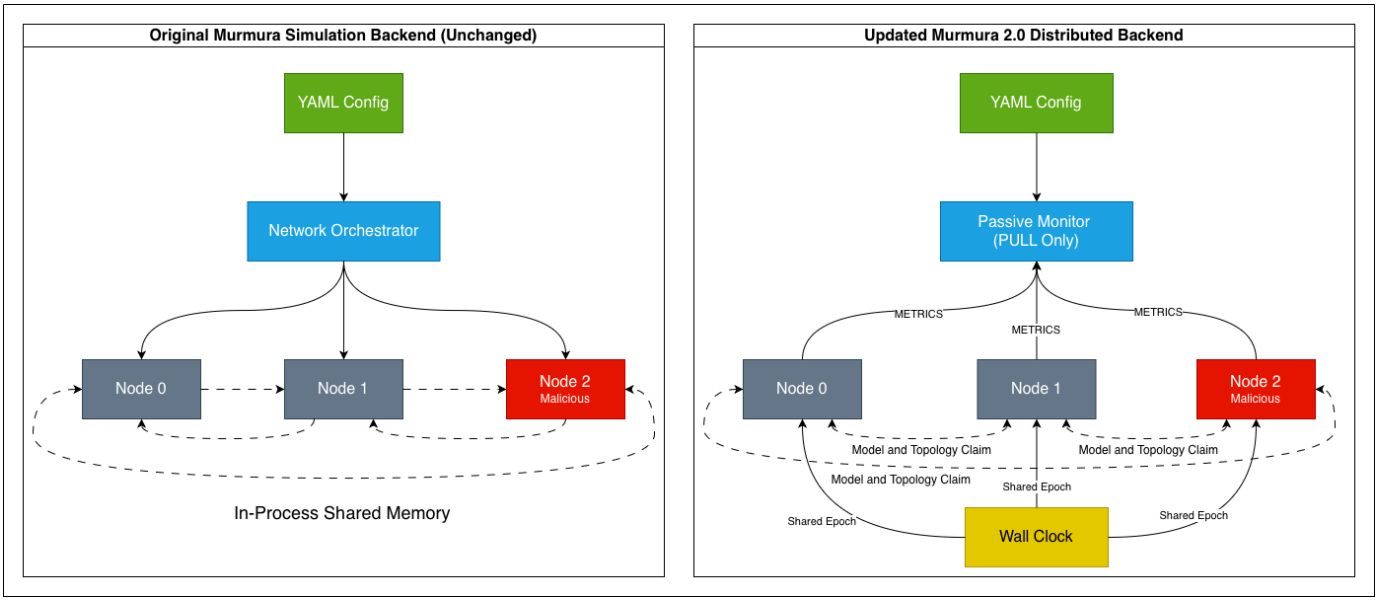

Fig. 1. Updated MURMURA architecture. The unchanged simulation backend (single OS process, shared memory) and the new distributed backend, in which each client and the passive monitor run as independent OS processes. Model weights travel via MODEL_STATE PUSH/PULL pairs; DMTT nodes additionally exchange signed neighbourhood observations as TOPO_CLAIM messages. Round boundaries are set by a shared wall-clock epoch; no coordinator is involved. Byzantine nodes send both poisoned model weights and falsified topology claims.

- **PUSH (monitor, connect):** Sends METRICS to the passive monitor after each round evaluation. The monitor never replies.

All messages use a two-frame ZMQ multipart format:

$$\underbrace{[\,\text{msg\_type}\,(1\ \text{B}) \parallel \text{sender\_id}\,(4\ \text{B})\,]}_{\text{frame 0 (header)}} \;\Big|\; \underbrace{[\,\text{payload}\,]}_{\text{frame 1}}$$

Three message types are defined: MODEL_STATE (serialised PyTorch state dict), METRICS (pickled evaluation dict containing a round_idx key for the monitor's ordering buffer), and TOPO_CLAIM (pickled neighbourhood assertion used exclusively by DMTT nodes).

C. Mobility Model

Client mobility is modelled as a bounded random walk on a 2-D torus of side length A. At each round, the position of client i is updated as

$$\mathbf{r}_i^{t+1} = \left(\mathbf{r}_i^t + \boldsymbol{\delta}_i^t\right) \bmod A, \tag{19}$$

where $\boldsymbol{\delta}_i^t \sim \text{Uniform}([-v_{\max}, v_{\max}]^2)$ is an independent per-round displacement drawn from a shared random generator initialised with the shared seed. An edge $(i,j) \in \mathcal{E}^t$ exists whenever the torus distance between $\mathbf{r}_i^t$ and $\mathbf{r}_j^t$ falls below the communication range r_c:

$$(i,j) \in \mathcal{E}^t \iff d_{\text{torus}}(\mathbf{r}_i^t, \mathbf{r}_j^t) < r_c. \tag{20}$$

An optional connectivity guarantee connects any isolated node to its nearest peer, ensuring that Assumption 4 (uniform connectivity) is not violated by unlucky mobility realisations.

Because both the initial positions and the displacement sequence are derived deterministically from the shared seed, every node process reconstructs $\mathcal{G}^t$ locally without any out-of-band communication, at the cost of maintaining one random-generator state per round.

D. DMTTNodeProcess Round Protocol

The DMTTNodeProcess class instantiates Algorithm 1 as a concrete ZeroMQ process. Each round it: sleeps until the wall-clock deadline t_k; trains locally; constructs and pushes MODEL_STATE + TOPO_CLAIM to the previous collaborator set; collects responses until the round boundary; updates link-reliability EMAs; evaluates received models to compute $s_{ij}^{\text{model},t}$; applies the oracle-free self-inclusion check—if j communicated with i this round then j's claim must include i—to accumulate d_{ij}^t (confirmations) and x_{ij}^t (contradictions) for the Beta trust update; applies the trust-screened aggregation rule of Section IV-E; updates C_i^t; and pushes evaluation metrics to the passive monitor. The dominant per-round overhead relative to vanilla NodeProcess is $|C_i^{t-1}|$ forward passes for foreign-model evaluation, scaling as $\mathcal{O}(|C_i^{t-1}| \cdot |\mathcal{D}_i^{\text{test}}|/b)$.

E. Topology-Liar Attack Implementation

The TopologyLiarAttack concretises the adversary model of Section III-C. A Byzantine node b mounts an *omission attack*: it sends a falsified TOPO_CLAIM containing only other Byzantine coalition members and no honest neighbours,

$$\widehat{\mathcal{N}}_b^t = \mathcal{B} \setminus \{b\}, \tag{21}$$

where $\mathcal{B}$ is the set of all Byzantine nodes. The omission is detectable without any shared topology knowledge. Because node i sent a MODEL_STATE to b this round, the directed link (i,b) is established at the transport layer; consequently i

is entitled to expect its own identity in b's claim. Since $i \notin \widehat{\mathcal{N}}_b^t$, the self-inclusion check fires a contradiction $x_{ib}^t \mathrel{+}= 1$ at every round, gradually raising β_{ib} and suppressing $T_{ib}^{\mathrm{topo},t}$ below the gate threshold τ_{trust}.

In addition to topology falsification, the liar optionally applies a Gaussian model-state attack (configurable via `model_attack_type: gaussian`), matching the combined threat model described in Section III-C.

VI. Convergence Analysis and Byzantine Bounds

This section presents the theoretical analysis of DMTT. The goal is not to prove exact Byzantine topology discovery. Instead, the analysis focuses on the learning dynamics induced by the trusted feasible graph maintained by each client. The presented analysis is consistent with existing works on decentralized optimization over time-varying graphs [14], [15], [16], topology–heterogeneity coupling [17], and Byzantine-resilient decentralized learning [24], [33]. All proofs are deferred to the supplementary material.

A. Effective collaboration matrix

Continuing the notation of Sections III–IV: $W^t = [w_{ij}^t]_{i,j=1}^N$ is the screened mixing matrix induced by the active collaborator sets $\{C_i^t\}$, with $w_{ij}^t = 0$ whenever $j \notin C_i^t \cup \{i\}$. DMTT is thus decentralized learning over a time-varying screened graph.

In personalized FL, perfect screening severs edges between clients with highly dissimilar data, partitioning the network into disjoint components of compatible clients. To analyze convergence under perfect screening (Theorems 1 and 2), we focus on an arbitrary isolated component $\mathcal{C}$; letting $N = |\mathcal{C}|$ we define the component-average iterate $\bar{\theta}^t = \frac{1}{N}\sum_{i\in\mathcal{C}} \theta_i^t$ and the component objective $f(x) = \frac{1}{N}\sum_{i\in\mathcal{C}} F_i(x)$. This correctly reflects the personalized DFL setting: the protocol filters the graph so that clients optimize their compatible component objective f rather than the true global average [14].

B. Standing assumptions and theory-local notation

Beyond the symbols of Sections III–IV, the analysis uses $\widetilde{W}^t$ (ideal honest mixing matrix), $p \in (0,1]$ (consensus-rate parameter of the screened graph sequence [14]), τ_{scr}^2 (screened neighborhood heterogeneity), Δ_{scr}^2 (screening-error level), $\delta_{\max}$ (maximum surviving Byzantine weight after screening), G (uniform stochastic-gradient second-moment bound, introduced with Assumption 6), and Θ_B (uniform Byzantine-message norm bound, introduced with Assumption 8), with $M := G + L\Theta_B$.

We collect below the four assumptions required by the consensus and descent lemmas (deferred to the supplementary material); Theorems and propositions later in the section introduce additional assumptions only where they are first needed. The link-reliability update, witness-weight rule, Beta trust update, edge-confidence rule, and collaborator score are treated as part of the DMTT mechanism, not as standalone assumptions.

a) Assumption 1: Smooth local objectives.: For each client i, the function F_i is L-smooth:

$$\|\nabla F_i(x) - \nabla F_i(y)\| \le L\|x - y\|, \qquad \forall x, y \in \mathbb{R}^d.$$

b) Assumption 2: Unbiased stochastic gradients with bounded variance.: Each client has access to a stochastic gradient oracle $g_i(x, \xi_i)$ such that

$$\mathbb{E}[g_i(x,\xi_i)] = \nabla F_i(x), \quad \mathbb{E}\|g_i(x,\xi_i) - \nabla F_i(x)\|^2 \le \sigma^2.$$

c) Assumption 3: Doubly stochastic screened mixing.: At each epoch t, the DMTT aggregation step produces a doubly stochastic matrix $W^t = [w_{ij}^t]$ satisfying

$$W^t\mathbf{1} = \mathbf{1}, \quad \mathbf{1}^\top W^t = \mathbf{1}^\top, \quad w_{ij}^t \ge 0,$$

with the screened-support property $w_{ij}^t = 0$ if $j \notin C_i^t \cup \{i\}$. In practice, this can be enforced by symmetrizing the trust-weighted scores, or by adopting a Metropolis–Hastings reweighting over the screened graph. A fully asymmetric collaboration matrix is also supported, but would require a push-sum or gradient-tracking treatment in place of the analysis below [15], [16].

d) Assumption 4: Consensus contraction of the screened honest graph.: Let $J = \frac{1}{N}\mathbf{1}\mathbf{1}^\top$. There exists $p \in (0,1]$ such that for every $t \ge 0$ and every $X \in \mathbb{R}^{N\times d}$ with $\mathbf{1}^\top X = 0$,

$$\|(W^t - J)X\|_F^2 \le (1-p)\|X\|_F^2.$$

The constant p is the consensus-rate parameter of the screened graph sequence. This per-step contraction is stronger than the B_{conn}-window connectivity formulation of [14]; the extension to the window case follows the same template with constants scaling as $1/p \to B_{\mathrm{conn}}/p$.

The convergence proof relies on standard structural invariants, a consensus contraction bound, and a descent lemma over the screened graph. These intermediate lemmas are detailed in the supplementary material.

C. Main theorem under perfect screening

Perfect screening means that the topology layer correctly excludes all Byzantine clients ($\delta_{\max} = 0$) and produces the ideal honest mixing matrix ($\Delta_{\mathrm{scr}} = 0$). Under this condition, DMTT reduces to time-varying decentralized SGD on the screened honest graph, with one local SGD step per round ($E = 1$). One additional assumption is needed beyond the standing ones: heterogeneity must be uniformly bounded across screened collaborators.

a) Assumption 5: Bounded screened heterogeneity.: There exists $\tau_{\mathrm{scr}}^2 \ge 0$ such that, for all $x \in \mathbb{R}^d$,

$$\frac{1}{N}\sum_{i\in\mathcal{C}} \big\|\nabla F_i(x) - \nabla f(x)\big\|^2 \le \tau_{\mathrm{scr}}^2.$$

The subscript "scr" indicates that this constant is measured over the population that participates in screened aggregation under DMTT. Corollary 1 discusses how `MURMURA`-style screening can reduce this constant relative to the heterogeneity over the raw feasible graph [17].

Theorem 1 (Convergence under perfect screening, $E = 1$)**.** *Suppose Assumptions 1–5 hold, screening is perfect ($\delta_{\max} = 0$ and $\Delta_{\mathrm{scr}}^2 = 0$), and the iterates are initialized at consensus ($\theta_i^0 = \bar{\theta}^0$ for all i). If the stepsize satisfies*

$$\eta \le \frac{p}{8L},$$

then for any horizon $T \ge 1$,

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\big\|\nabla f(\bar{\theta}^t)\big\|^2 \le \frac{4\big(f(\bar{\theta}^0) - f^\star\big)}{\eta T} + \frac{2L\eta\,\sigma^2}{N} + \frac{64\,L^2\,\eta^2\big(\tau_{\mathrm{scr}}^2 + \sigma^2\big)}{p^2}, \tag{22}$$

where $f^\star = \inf_x f(x)$. The proof is given in the supplementary material.

Remark 1 (Interpretation of the rate)*.* The three terms in (22) are the optimization error from initialization ($\mathcal{O}(1/T)$), the stochastic-noise floor ($\propto \sigma^2/N$), and the heterogeneity-and-topology term ($\propto (\tau_{\mathrm{scr}}^2 + \sigma^2)/p^2$). Choosing $\eta = \min\big\{\frac{p}{8L}, \sqrt{\frac{N(f^0 - f^\star)}{L\sigma^2 T}}\big\}$ yields an asymptotic convergence rate of $\mathcal{O}\big(\frac{1}{\sqrt{NT}} + \frac{N(\tau_{\mathrm{scr}}^2 + \sigma^2)}{p^2 T}\big)$. While state-of-the-art analyses can tighten the second term to $\mathcal{O}(1/(pT))$ using complex potential functions [14], this standard analysis suffices to explicitly show how the DMTT screening filter accelerates convergence by reducing the topological heterogeneity τ_{scr}^2.

Remark 2. In the supplementary material, we extend this result to $E \ge 1$ local steps (Theorem 2) under an additional bounded-gradient assumption ($\mathbb{E}\|g_i\|^2 \le G^2$); local updates introduce an $\mathcal{O}(\eta^2 L^2 G^2/p^2)$ drift penalty but reduce the stochastic noise floor to $\sigma^2/(NE)$.

D. Why screening helps

The bound (22) shows that convergence depends heavily on the data heterogeneity τ_{scr}^2. DMTT explicitly shrinks this constant by severing edges to model-incompatible peers (large $\bar{u}_{ij}^t$). While dropping edges may reduce the graph's consensus rate p, the reduction in heterogeneity typically dominates.

Corollary 1 (Screening improves the bound)**.** *Let $\mathcal{V}_{\mathrm{raw}}$ be the full, unfiltered network with objective f_{raw}, resulting in global heterogeneity τ_{raw}^2 and consensus rate p_{raw}. Suppose the DMTT screening mechanism isolates a highly compatible component $\mathcal{C}$, shifting the target objective to f as defined above. If the screening yields*

$$\tau_{\mathrm{scr}}^2 \le \tau_{\mathrm{raw}}^2 \quad \textit{and} \quad p_{\mathrm{scr}} \ge p_{\mathrm{raw}} - \epsilon$$

for some $\epsilon \ge 0$, then the heterogeneity-and-topology term in (22) *satisfies*

$$\frac{64L^2\eta^2\big(\tau_{\mathrm{scr}}^2 + \sigma^2\big)}{p_{\mathrm{scr}}^2} \le \frac{64L^2\eta^2\big(\tau_{\mathrm{raw}}^2 + \sigma^2\big)}{(p_{\mathrm{raw}} - \epsilon)^2}.$$

By filtering out incompatible peers, DMTT deliberately abandons the global stationary point in favor of a personalized component stationary point, achieving a tighter convergence bound as long as the reduction in data heterogeneity outpaces the loss in subgraph connectivity.

E. Perturbation under imperfect screening

We close the analysis with the bound that quantifies the cost of imperfect screening. The proposition makes precise the two terms that appear when (i) the screened matrix differs from the ideal honest mixing ($\Delta_{\mathrm{scr}} > 0$), and (ii) Byzantine peers retain residual aggregation weight ($\delta_{\max} > 0$). Three additional assumptions are needed: a bounded second moment of the stochastic gradient (which strengthens Assumption 2 and controls the local drift used in the perturbation term), a bound on the screening error and surviving Byzantine influence, and a bound on the norm of Byzantine messages. Without the last, Byzantine influence cannot be bounded in general because adversarial peers can broadcast arbitrarily large updates.

a) Assumption 6: Bounded gradient second moments.: There exists $G \ge 0$ such that for all $i \in \mathcal{V}$, $t \ge 0$, and points x visited by the iterates,

$$\mathbb{E}\|g_i(x, \xi_i)\|^2 \le G^2.$$

This is stronger than Assumption 2 (bounded *variance*); note $G^2 \ge \sigma^2 + \sup_x \|\nabla f(x)\|^2$. In practice, uniform Lipschitz regularization or gradient clipping enforces it. Assumption 6 is invoked in Proposition 1 below (through ηG in the drift magnitude ρ) and in the extension to $E > 1$ local steps (Theorem 2, supplementary material).

b) Assumption 7: Bounded screening error and surviving Byzantine influence.: Let $\widetilde{W}^t$ denote the ideal honest mixing matrix induced by the true feasible graph, and let W^t denote the actual DMTT screened matrix. Assume

$$\mathbb{E}\|W^t - \widetilde{W}^t\|_F^2 \le \Delta_{\mathrm{scr}}^2.$$

Also define

$$\delta_{\max} = \sup_t \max_{i \in \mathcal{H}} \sum_{j \in \mathcal{B}} w_{ij}^t,$$

and assume $\delta_{\max}$ is uniformly bounded and sufficiently small relative to the connectivity of the screened honest graph.

c) Assumption 8: Bounded Byzantine message norms.: There exists $\Theta_B \ge 0$ such that, for every Byzantine client $j \in \mathcal{B}$ and every epoch t,

$$\big\|\theta_j^{B,t+\frac{1}{2}}\big\| \le \Theta_B$$

almost surely, where $\theta_j^{B,t+\frac{1}{2}}$ denotes the (possibly adversarial) value broadcast by Byzantine peer j in round t. This assumption is invoked only in Proposition 1. It is *not* a consequence of Assumptions 1–7. The norm-sanity gate of Section IV-E provides a concrete enforcement: any peer j that survives to aggregation satisfies $\|\theta_j\| \le \kappa_{\mathrm{norm}}\|\theta_i\|$. Under Assumption 9 ($\|\theta_i\| \le R$ for all i, t), every surviving Byzantine peer satisfies $\|\theta_j\| \le \kappa_{\mathrm{norm}} R$, giving $\Theta_B = \kappa_{\mathrm{norm}} R$ as a hard bound for peers not dropped by the gate.

d) Assumption 9: Bounded domain.: The local iterates are confined to a bounded domain such that for all $i \in \mathcal{C}$ and $t \ge 0$, $\|\theta_i^t\| \le R$. In practice, this is enforced via projected SGD or ℓ_2 weight decay.

Proposition 1 (Imperfect screening)**.** *Suppose Assumptions 1–9 hold over the screened component $\mathcal{C}$ of size N. Let $\widetilde{W}^t$ be the*

ideal doubly stochastic matrix over $\mathcal{C}$, and let W^t be the actual DMTT submatrix. Assume $\mathbb{E}\|W^t - \widetilde{W}^t\|_F^2 \le \Delta_{\mathrm{scr}}^2$ and let $\delta_{\max}$ bound the total weight any honest node assigns to Byzantine peers. Define the worst-case per-step drift magnitude:

$$\rho := \Delta_{\mathrm{scr}}(R + \eta G) + \delta_{\max}\Theta_B.$$

If $\eta \le \frac{p}{24L}$ and $E = 1$, then for any horizon $T \ge 1$:

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\big\|\nabla f(\bar{\theta}^t)\big\|^2 \le \frac{4(f^0 - f^\star)}{\eta T} + \frac{4L\eta\sigma^2}{N} + \frac{256L^2\eta^2(\tau_{\mathrm{scr}}^2 + \sigma^2/4)}{p^2} + \frac{48L^2}{p^2}\rho^2 + \Big(\frac{8}{\eta^2} + \frac{4L}{\eta}\Big)\rho^2. \tag{23}$$

Remark 3 (Interpretation). The first three terms of (23) match the Byzantine-free rate of Theorem 1 up to constants; the final term, driven by ρ^2, isolates the topology-screening error Δ_{scr} and the surviving Byzantine influence $\delta_{\max}$. Because the Byzantine message induces an additive shift in *parameter* space rather than *gradient* space, the penalty scales as $\mathcal{O}(\rho^2/\eta^2)$: unlike standard SGD, shrinking η does not drive the error to zero — an arbitrarily small stepsize would let the unscaled Byzantine parameter drift dominate the gradient signal, so a fixed stepsize floor is essential in robust decentralized learning.

VII. Experimental Evaluation

A. Conditions and Baselines

The controlled comparison isolates the contributions of dynamic topology and of the DMTT trust protocol through three primary conditions, all subjected to the same combined topology-liar + Gaussian attack:

- **C1. FedAvg (static).** Fixed fully-connected topology; plain FedAvg over all neighbours; no trust. This is the threat faced by unmodified decentralized averaging when the topology is wrongly assumed static.
- **C2. FedAvg (dynamic).** Dynamic topology $\mathcal{G}^t$ from the mobility model; each node averages over all direct neighbours $\mathcal{N}_i^{\mathrm{dir},t}$ with no trust filtering. Comparing C1 and C2 isolates the effect of mobility alone.
- **C3. DMTT.** Same mobility model as C2 with the full DMTT protocol active: link-reliability EMA, Beta source-trust with `TOPO_CLAIM` verification, and the trust-screened, q_{ij}-weighted aggregation of Section IV-E. Comparing C2 and C3 isolates the benefit of trust.

To position DMTT against the state of the art in Byzantine-resilient decentralized aggregation, we additionally evaluate three robust aggregators over the same dynamic graph and attack: Krum (multi-Krum distance selection) [34], BALANCE (adaptive distance filtering) [25], and UBAR (two-stage distance-plus-loss filtering) [22]. We first validated the distributed ZeroMQ backend against the simulation backend at 10 nodes across three Byzantine fractions and two seeds each, running DMTT and FedAvg (dynamic) on a real Melbourne Research Cloud (MRC) testbed with wall-clock synchronisation and IPC transport. The two backends produce qualitatively identical convergence trajectories across all tested conditions: DMTT converges to high accuracy in both cases, and FedAvg collapses to chance under attack in both cases. The mean final honest-node accuracy (last 5 of 20 rounds) agrees within 0.07 across all condition and Byzantine fraction pairs (convergence curves in the supplementary material). Having established that the simulation faithfully reproduces real distributed behaviour, we ran all main experiments at 100 nodes on the simulation backend, which executes the protocol logic in a single process, making the full sweep across 8 Byzantine fractions and 3 seeds per condition deterministic and tractable.

B. Datasets and Models

a) UCI HAR.: We first evaluate on **UCI HAR** [35], the smartphone human-activity-recognition benchmark used in the original `MURMURA` study: 561 hand-crafted inertial features per window and six activity classes (walking, walking upstairs, walking downstairs, sitting, standing, lying). The 7,352 training instances are partitioned across $N = 100$ mobile clients with a Dirichlet distribution of concentration $\alpha = 0.5$, producing realistic non-IID local distributions; each client holds out 20% of its partition as a local test set on which all reported accuracies are computed. Every client trains a two-hidden-layer perceptron (256–128 units, dropout 0.3) with a Dirichlet (evidential) output head, optimised with the evidential loss of [4], [13]. Chance accuracy on the six balanced classes is $\frac{1}{6} \approx 0.167$.

b) PAMAP2.: To confirm generality we additionally evaluate on **PAMAP2** [36], a body-worn sensor benchmark with richer temporal structure and finer-grained activity classes: nine subjects performed twelve physical activities (lying, sitting, standing, walking, running, cycling, Nordic walking, ascending/descending stairs, vacuuming, ironing, rope jumping) while wearing inertial measurement units on hand, chest, and ankle (100 Hz, 13 features per IMU after excluding invalid orientation columns) together with a wrist heart-rate monitor. Raw sensor streams are segmented into 50-sample (0.5-second) windows with 25-sample stride; each window is flattened to a 2,000-dimensional feature vector (50×40). This representation captures within-activity dynamics while remaining computationally comparable to UCI HAR's pre-engineered 561-feature windows. The roughly 38,800 windows are partitioned across the same $N = 100$ clients using a Dirichlet $\alpha = 0.5$ split; each client trains an evidential perceptron (256–128 units, dropout 0.3) of the same depth as the UCI HAR model. Because PAMAP2 has 12 balanced classes, chance accuracy is $\frac{1}{12} \approx 0.083$, and any method above 0.50 is meaningfully learning. The model-compatibility score s_{ij}^{model} is computed identically on both datasets: local validation accuracy combined with epistemic vacuity uncertainty.

C. Results

1) UCI HAR:

a) Trust-aware screening is decisive.: Figure 2 (top two rows) shows final honest-node accuracy for all methods across Byzantine fractions from 10% to 80% (mean $\pm$ std over three

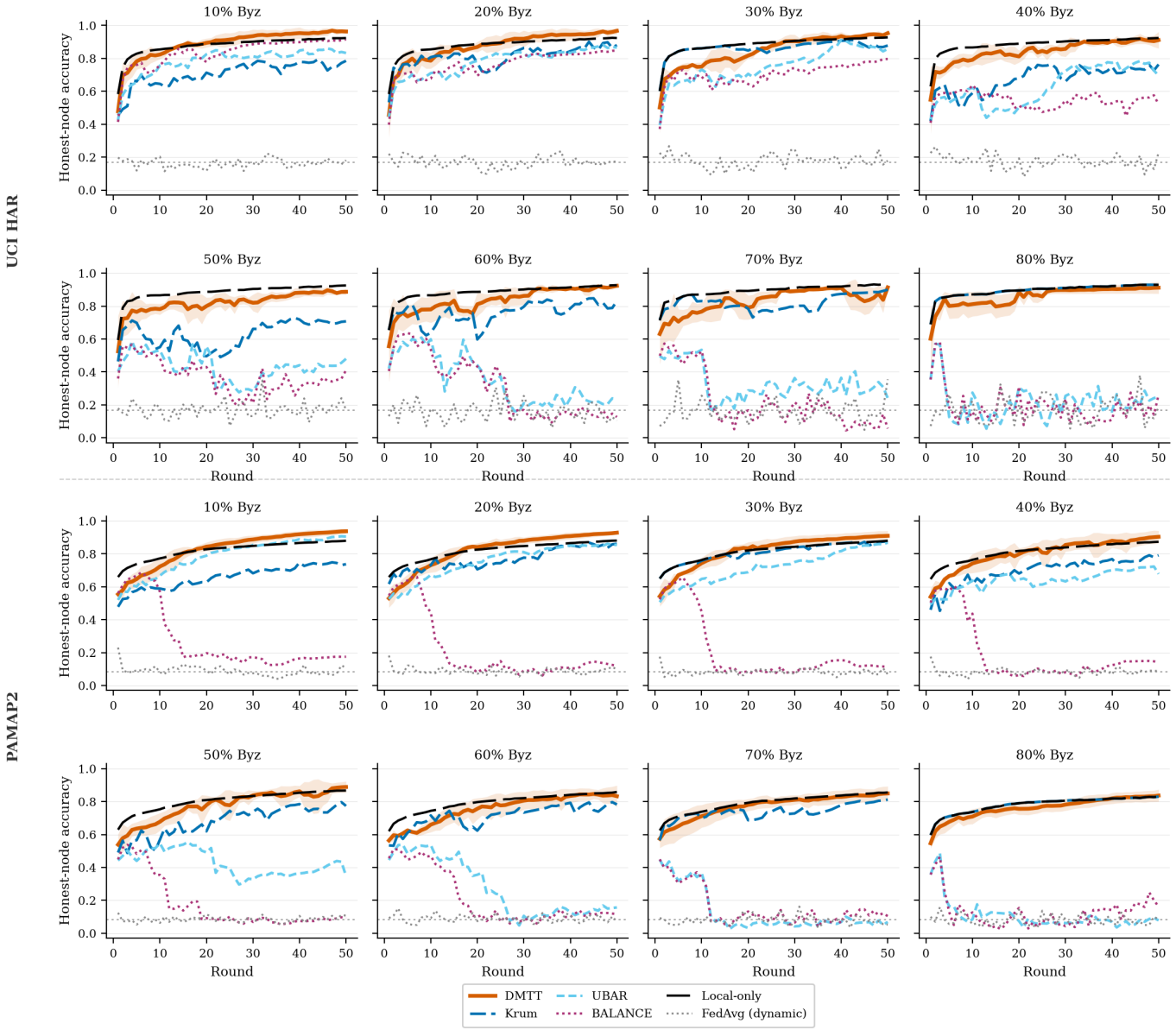

Fig. 2. Honest-node accuracy versus round under the combined topology-liar + Gaussian (σ=10) attack (mean $\pm$ std over three seeds). **Top two rows**: UCI HAR (6 classes, chance = 0.167). **Bottom two rows**: PAMAP2 (12 classes, chance = 0.083). FedAvg collapses to chance at all fractions on both datasets. Krum and BALANCE each fail to consistently clear the local-only bar. UBAR is competitive at low Byzantine fractions but collapses above 50%. DMTT sustains high honest-node accuracy across all 10–80% fractions on both datasets, remaining at or above local-only throughout.

seeds, averaged over the last 10 of 50 rounds). Both non-robust baselines collapse to chance at all fractions: **FedAvg (static)** and **FedAvg (dynamic)** never exceed 0.22, confirming that even a single $\sigma = 10$ Gaussian-poisoned model averaged into the aggregate destroys learning, regardless of graph type. **DMTT** instead sustains honest accuracy above 0.862 at 10%–80% Byzantine, approximately a 5$\times$ improvement over non-robust baselines at all fractions.

b) Graceful degradation at extreme Byzantine fractions.: At low-to-moderate fractions (10–30% Byzantine), DMTT significantly outperforms **Local-only** (0.920–0.923), demonstrating that DMTT extracts genuine collaborative benefit from honest neighbours while screening adversaries. As the Byzantine fraction rises above 40%, the number of reachable honest neighbours per node in the dynamic graph diminishes, so the screened aggregate converges toward the local model and the collaboration surplus over local-only narrows. Accuracy is non-monotone in the Byzantine fraction: DMTT dips to 0.862 ± 0.061 at 70% before recovering to 0.908 ± 0.031 at 80%. This reflects the fallback-to-local-only mechanism: at 80% Byzantine, trust gates exclude nearly all adversarial neighbours and the effective aggregate is almost entirely the local model; since local-only accuracy (0.927) is high, performance recovers relative to 70%. At 70% the honest-neighbour pool is still large enough that the system attempts more collaboration, but the high adversarial density makes

TABLE II
EXPERIMENTAL HYPERPARAMETERS. PRIMARY METRIC: MEAN HONEST-NODE ACCURACY ON EACH CLIENT'S HELD-OUT LOCAL TEST PARTITION, AVERAGED OVER THE FINAL 10 OF 50 ROUNDS.

Parameter	**Value**
Number of clients N	100
Datasets	UCI HAR, PAMAP2
Heterogeneity	Dirichlet $\alpha = 0.5$
Byzantine fraction	10–80% (10–80 nodes)
Attack type	Topology liar + Gaussian ($\sigma = 10$)
Training rounds	50
Local epochs E	2
Batch size / learning rate	32 / 0.01
Seeds	$\{42, 43, 44\}$
Mobility model (dynamic conditions)	
Arena A	100×100
Communication range r_c	40
Max speed $v_{\max}$	8 per round
DMTT protocol (C3)	
Budget B	5
EMA rate ρ	0.1
Forgetting factor λ	0.9
Trust weights (w_d, w_x)	$(1.0,\ 1.0)$
Uncertainty threshold / penalty (τ_U, η)	$(0.3,\ 5.0)$
Score weights $(\lambda_1, \lambda_2, \lambda_3, \lambda_4)$	$(0.4,\ 0.3,\ 0.2,\ 0.1)$
Model-compat. gate γ	0.6
Topology-trust gate τ_{trust}	0.25
Norm-sanity ratio	5×
Self-weight ω	0.5

clean peer selection harder, yielding a lower mean with higher variance (σ=0.061 vs. 0.031 at 80%). At 80% Byzantine (80 of 100 nodes adversarial), DMTT (0.908 ± 0.031) is within one standard deviation of Local-only (0.927 ± 0.035) and clearly superior to BALANCE (0.191) and UBAR (0.225), confirming the *near-safety* property: even when honest nodes are overwhelmed by adversaries, trust-screened DMTT degrades to approximately local-only performance rather than collapsing to chance.

c) Dedicated robust aggregators fail to clear the local-only bar.: By the local-only bar, all three dedicated robust baselines fail. **Krum** is competitive at 10–30% Byzantine (0.768–0.887) but degrades at 40–60% (0.734–0.819), below both DMTT (0.880–0.907) and local-only (0.919–0.922), because topology-liar poisoning causes it to select compromised models that pass the distance filter. At 70% Byzantine, Krum's performance is variable across seeds (ranging from 0.805 to 0.942, sometimes exceeding local-only and sometimes falling below it). At 80% Byzantine, Krum's output matches local-only *exactly on every seed* (mean accuracy difference: 0.000). With 80 of 100 nodes adversarial, Multi-Krum's nearest-neighbour count collapses for any realistic neighbourhood size, so Krum selects only the single closest model. Under $\sigma = 10$ Gaussian noise the honest models cluster tightly and the selected peer is invariably an honest one, but aggregating with a single peer under Dirichlet $\alpha = 0.5$ non-IID data provides negligible collaborative gain, yielding accuracy statistically indistinguishable from local-only. This is the same graceful degradation toward local-only that DMTT exhibits at extreme fractions, reached here through a different mechanism. **BALANCE** degrades catastrophically (0.537 at 40%, 0.147 at 60%, 0.078 at 70%), driven by the brittleness of adaptive distance thresholds under simultaneous topology manipulation and non-IID data. **UBAR** degrades more gradually but still falls below local-only above 50% Byzantine. DMTT is the only method that consistently clears the local-only bar at all fractions from 10% to 80%.

d) Byzantine influence is screened to zero.: All poisoned peers are excluded by the three aggregation gates: $\delta_{\max} = 0$ in every round, even at Byzantine fractions up to 80%. Under 30% Byzantine participation, honest-peer trust rises to $T_{ij}^{\text{topo}} \approx 0.986$ by rounds 40–50, whereas topology liars fall to 0.012, producing a 0.974 separation via the oracle-free self-inclusion check.

2) PAMAP2: We next evaluate on PAMAP2 to test whether DMTT's trust mechanisms generalize to richer sensor data with finer-grained activity classes and raw temporal windows, a setting that stresses distance-based aggregators more severely than UCI HAR's pre-engineered features.

a) Generalisation to richer sensor data.: Figure 2 (bottom two rows) shows the same comparison on PAMAP2 (12 classes, chance = 0.083). Both FedAvg variants collapse to near-chance at all Byzantine fractions (0.064–0.095, indistinguishable from chance). **DMTT** sustains honest accuracy above 0.829 across all Byzantine concentrations, a 9×–13× lift over the non-robust baselines. Crucially, DMTT clears the local-only bar (0.873–0.823) at *every* Byzantine fraction from 10% to 80%; because PAMAP2's twelve-class task is harder, the collaboration benefit persists even at extreme fractions. **BALANCE** collapses entirely on PAMAP2 (reaching only 0.124–0.170 across all fractions, near-chance on a 12-class task), confirming that its distance thresholds cannot tolerate simultaneous topology manipulation and raw-window feature heterogeneity. **UBAR** starts competitively (0.894 at 10%) but collapses at 50% Byzantine (mean 0.395 ± 0.350, highly variable) and reaches chance at 70%–80%. The Beta trust separation replicates exactly: honest peers reach $T_{ij}^{\text{topo}} \approx 0.997$ while liars are suppressed to $T_{ij}^{\text{topo}} \approx 0.022$ (identical to UCI HAR), confirming that the trust layer is insensitive to the choice of feature representation.

b) Convergence.: Figure 2 (bottom two rows) shows PAMAP2 learning trajectories across all eight Byzantine fractions. **BALANCE** fails near-chance throughout at every fraction, confirming that its adaptive distance thresholds cannot tolerate simultaneous topology manipulation and raw-window feature heterogeneity. **UBAR** converges well at low Byzantine fractions but collapses above 50%, exhibiting high seed-to-seed variance at that threshold. **DMTT** maintains clear separation from all baselines at every fraction, remaining at or above the local-only reference throughout. The same qualitative pattern holds as on UCI HAR, but with a sustained collaboration surplus even at extreme fractions due to PAMAP2's twelve-class task difficulty.

TABLE III
DMTT HYPERPARAMETER SENSITIVITY (MEAN ± STD OVER 3 SEEDS, LAST 10 OF 50 ROUNDS, α=0.5, COMBINED TOPOLOGY-LIAR + GAUSSIAN ATTACK). ONE PARAMETER IS VARIED AT A TIME; ALL OTHERS REMAIN AT THE DEFAULT.

Configuration	10% Byz	30% Byz	50% Byz	80% Byz
Default (λ=0.90, ρ=0.10, τ=0.25, B=5)	0.960 ± 0.014	0.936 ± 0.015	0.880 ± 0.018	0.907 ± 0.025
$\lambda = 0.70$ (faster trust decay)	0.960 ± 0.015	0.940 ± 0.012	0.887 ± 0.012	0.908 ± 0.025
$\lambda = 0.95$ (slower trust decay)	0.960 ± 0.015	0.940 ± 0.012	0.886 ± 0.012	0.908 ± 0.025
$\rho = 0.05$ (slower link EMA)	0.959 ± 0.015	0.940 ± 0.012	0.885 ± 0.010	0.908 ± 0.025
$\rho = 0.30$ (faster link EMA)	0.960 ± 0.015	0.940 ± 0.012	0.885 ± 0.011	0.908 ± 0.025
$\tau_{\text{trust}} = 0.10$ (loose gate)	0.960 ± 0.015	0.940 ± 0.012	0.886 ± 0.012	0.908 ± 0.025
$\tau_{\text{trust}} = 0.50$ (strict gate)	0.960 ± 0.015	0.940 ± 0.012	0.886 ± 0.012	0.908 ± 0.025
$B = 3$ (fewer collaborators)	0.956 ± 0.018	0.936 ± 0.011	0.886 ± 0.012	0.908 ± 0.025
$B = 7$ (more collaborators)	0.959 ± 0.016	0.939 ± 0.011	0.886 ± 0.012	0.908 ± 0.025

D. Hyperparameter Sensitivity

A practical concern for any trust-based protocol is whether performance depends on careful parameter tuning. Table III addresses this by varying one DMTT hyperparameter at a time while holding all others at the default, across four Byzantine fractions on both datasets.

The four parameters tested cover the trust dynamics (λ, the Beta forgetting factor), the link-reliability estimator (ρ, the EMA smoothing rate), the topology-trust gate threshold (τ_{trust}), and the collaborator budget (B). Across all eight variants, honest-node accuracy stays within ± 0.005 of the default at every Byzantine fraction. The forgetting factor λ can be raised from 0.70 to 0.95 with no measurable effect, because topology-liar contradictions accumulate rapidly at every decay rate: even with $\lambda = 0.70$, a liar that falsifies two neighbours per round reaches a suppressed trust score within a handful of rounds. The EMA rate ρ is similarly irrelevant, as direct link quality is perfectly observable under the shared mobility model and does not require smoothing on any particular timescale. The trust gate threshold τ_{trust} can be relaxed to 0.10 or tightened to 0.50 without loss: Byzantine nodes are driven to $T_{ij}^{\text{topo}} \lesssim 0.02$ after trust convergence, well below even the loose threshold. Reducing the budget to $B = 3$ produces a small drop at 10% Byzantine (0.956 vs. 0.960) because the node has fewer honest collaborators available, but the difference closes entirely at higher Byzantine fractions where the feasible collaborator pool is already constrained.

VIII. CONCLUSIONS AND FUTURE WORK

We presented DMTT, a protocol that extends MURMURA to dynamic network topologies under adversarial topology-manipulation attacks. The theoretical analysis shows that DMTT confines Byzantine influence to a bounded residual $\delta_{\max}$ in the mixing matrices, which shrinks as the Beta-evidence trust mechanism accumulates observations. MURMURA was also redesigned into a production-ready distributed framework with coordinator-free wall-clock synchronisation (Eq. (18)), ZeroMQ peer-to-peer transport, a deterministic mobility model (Eqs. (19)–(20)), and an omission topology-liar attack (Eq. (21)) detectable via an oracle-free self-inclusion check.

Empirically, on UCI HAR and PAMAP2 across 100 mobile clients under combined topology-liar and Gaussian attacks, DMTT sustained honest-node accuracy above 0.862 on UCI HAR and 0.829 on PAMAP2 at every Byzantine fraction from 10% to 80%, while static and dynamic FedAvg collapsed to chance. Dedicated robust aggregators (Krum, BALANCE, UBAR) all failed to consistently beat the local-only baseline: Krum degraded at 40–60% Byzantine (0.734–0.819 vs. DMTT's 0.880–0.910) as topology-liar poisoning let compromised models pass its distance filter; BALANCE collapsed near-chance on PAMAP2 even at 10% Byzantine (0.170); and UBAR collapsed above 50% Byzantine (mean 0.395 ± 0.350 at 50%, chance by 70–80%). DMTT was the only method to clear the local-only bar across both datasets at all tested fractions, degrading gracefully toward near-local-only performance at extreme fractions rather than collapsing, a safety guarantee no other aggregator provides. A trust diagnostic confirmed *exactly zero* Byzantine aggregation weight at every fraction, consistent with the $\delta_{\max} = 0$ condition of Theorem 1.

Future work includes relaxing the doubly-stochastic assumption on W^t to accommodate asymmetric MURMURA-style weights, extending the analysis to bounded-staleness execution, and validating DMTT on real mobile deployments where neighbours are discovered via physical beacons rather than a shared mobility model, a change that would affect only the routing/discovery step, since the self-inclusion check already relies solely on transport-level delivery acknowledgements.

REFERENCES

[1] H. Ye, L. Liang, and G. Y. Li, "Decentralized federated learning with unreliable communications," *IEEE journal of selected topics in signal processing*, vol. 16, no. 3, pp. 487–500, 2022.

[2] S. Kharrat, M. Canini, and S. Horvath, "Decentralized personalized federated learning," *arXiv preprint arXiv:2406.06520*, 2024.

[3] Y. Liu, Y. Shi, Q. Li, B. Wu, X. Wang, and L. Shen, "Decentralized directed collaboration for personalized federated learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 23 168–23 178.

[4] M. Rangwala, R. O. Sinnott, and R. Buyya, "Evidential trust-aware model personalization in decentralized federated learning for wearable iot," *arXiv preprint arXiv:2512.19131*, 2025.

[5] Z. Wu, Z. Xu, D. Zeng, J. Li, and J. Liu, "Topology learning for heterogeneous decentralized federated learning over unreliable d2d networks," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 8, pp. 12 201–12 206, 2024.

[6] M. Nesterenko and S. Tixeuil, "Discovering network topology in the presence of byzantine faults," *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, no. 12, pp. 1777–1789, 2009.

[7] R. Gaucher, A. Dieuleveut, and H. Hendrikx, "Unified breakdown analysis for byzantine robust gossip," *arXiv preprint arXiv:2410.10418*, 2024.

[8] T. Sun, D. Li, and B. Wang, "Decentralized federated averaging," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 4, pp. 4289–4301, 2023.

[9] W. Liu, L. Chen, and W. Zhang, "Decentralized federated learning: Balancing communication and computing costs," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 8, pp. 131–143, 2022.

[10] V. Zantedeschi, A. Bellet, and M. Tommasi, "Fully decentralized joint learning of personalized models and collaboration graphs," in *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, S. Chiappa and R. Calandra, Eds., vol. 108. PMLR, 26–28 Aug 2020, pp. 864–874. [Online]. Available: https://proceedings.mlr.press/v108/zantedeschi20a.html

[11] S. Li, T. Zhou, X. Tian, and D. Tao, "Learning to collaborate in decentralized learning of personalized models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 9766–9775.

[12] R. Ye, Z. Ni, F. Wu, S. Chen, and Y. Wang, "Personalized federated learning with inferred collaboration graphs," in *International conference on machine learning*. PMLR, 2023, pp. 39 801–39 817.

[13] M. Sensoy, L. Kaplan, and M. Kandemir, "Evidential deep learning to quantify classification uncertainty," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[14] A. Koloskova, N. Loizou, S. Boreiri, M. Jaggi, and S. Stich, "A unified theory of decentralized sgd with changing topology and local updates," in *International conference on machine learning*. PMLR, 2020, pp. 5381–5393.

[15] A. Nedić and A. Olshevsky, "Distributed optimization over time-varying directed graphs," *IEEE Transactions on Automatic Control*, vol. 60, no. 3, pp. 601–615, 2014.

[16] A. Nedić and A. Olshevsky, "Stochastic gradient-push for strongly convex functions on time-varying directed graphs," *IEEE Transactions on Automatic Control*, vol. 61, no. 12, pp. 3936–3947, 2016.

[17] B. Le Bars, A. Bellet, M. Tommasi, E. Lavoie, and A.-M. Kermarrec, "Refined convergence and topology learning for decentralized sgd with heterogeneous data," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2023, pp. 1672–1702.

[18] Y. Lu, Z. Yu, and N. Suri, "Privacy-preserving decentralized federated learning over time-varying communication graph," *ACM Transactions on Privacy and Security*, vol. 26, no. 3, pp. 1–39, 2023.

[19] Y. Shi, Q. Ma, Y. Xu, J. Zhou, M. Hu, Y. Liao, and H. Xu, "Dystop: Dynamic staleness control and topology construction for asynchronous decentralized federated learning," *IEEE Transactions on Mobile Computing*, 2026.

[20] B. Li, W. Gao, X. Deng, J. Xie, Z. Xiong, M. Siew, B. Guo, S. Mao, and Z. Han, "Decentralized federated learning over time-varying and heterogeneous mobile computing networks," *IEEE Transactions on Mobile Computing*, vol. 25, no. 5, pp. 6688–6704, 2026.

[21] C. Fang, Z. Yang, and W. U. Bajwa, "Bridge: Byzantine-resilient decentralized gradient descent," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 8, pp. 610–626, 2022.

[22] S. Guo, T. Zhang, H. Yu, X. Xie, L. Ma, T. Xiang, and Y. Liu, "Byzantine-resilient decentralized stochastic gradient descent," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 6, pp. 4096–4106, 2021.

[23] Z. Wu, T. Chen, and Q. Ling, "Byzantine-resilient decentralized stochastic optimization with robust aggregation rules," *IEEE transactions on signal processing*, vol. 71, pp. 3179–3195, 2023.

[24] L. He, S. P. Karimireddy, and M. Jaggi, "Byzantine-robust decentralized learning via clippedgossip," *arXiv preprint arXiv:2202.01545*, 2022.

[25] M. Fang, Z. Zhang, Hairi, P. Khanduri, J. Liu, S. Lu, Y. Liu, and N. Gong, "Byzantine-robust decentralized federated learning," in *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security*, 2024, pp. 2874–2888.

[26] C. Che, X. Li, C. Chen, X. He, and Z. Zheng, "A decentralized federated learning framework via committee mechanism with convergence guarantee," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 12, pp. 4783–4800, 2022.

[27] Y. Tao, S. Cui, W. Xu, H. Yin, D. Yu, W. Liang, and X. Cheng, "Byzantine-resilient federated learning at edge," *IEEE Transactions on Computers*, vol. 72, no. 9, pp. 2600–2614, 2023.

[28] H. Perrey, M. Landsmann, O. Ugus, T. C. Schmidt, and M. Wählisch, "Trail: Topology authentication in rpl," *arXiv preprint arXiv:1312.0984*, 2013.

[29] C. Li, M. Hurfin, and Y. Wang, "Reputation propagation and updating in mobile ad hoc networks with byzantine failures," in *2015 IEEE Trustcom/BigDataSE/ISPA*, vol. 1, 2015, pp. 111–118.

[30] Y. Miao, Z. Liu, H. Li, K.-K. R. Choo, and R. H. Deng, "Privacy-preserving byzantine-robust federated learning via blockchain systems," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 2848–2861, 2022.

[31] M. Rangwala, F. Azzedin, R. O. Sinnott, and R. Buyya, "Sketchguard: Scaling byzantine-robust decentralized federated learning via sketch-based screening," *arXiv preprint arXiv:2510.07922*, 2025.

[32] A. Gupta and G. Pandurangan, "Fully-distributed construction of byzantine-resilient dynamic peer-to-peer networks," *arXiv preprint arXiv:2506.04368*, 2025.

[33] Z. Wu, T. Chen, and Q. Ling, "Byzantine-resilient decentralized stochastic optimization with robust aggregation rules," *IEEE Transactions on Signal Processing*, vol. 71, pp. 3179–3195, 2023.

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

[35] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "A public domain dataset for human activity recognition using smartphones," in *European Symposium on Artificial Neural Networks (ESANN)*, vol. 3, 2013, p. 3.

[36] A. Reiss and D. Stricker, "Introducing a new benchmarked dataset for activity monitoring," in *Proceedings of the 2012 16th Annual International Symposium on Wearable Computers (ISWC)*, ser. ISWC '12. USA: IEEE Computer Society, 2012, p. 108–109.

Shubham Vaishnav received a Ph.D. from the Department of Computer and Systems Science, Stockholm University, Sweden, in June, 2026. He received Bachelor's and Master's degrees in Computer Science & Engineering from IIT (ISM), Dhanbad in 2017 and 2019, respectively. His research interests include reinforcement learning, federated Learning, distributed systems, optimization, and adaptive decision-making in dynamic environments such as IoT.

Murtaza Rangwala received his B.Eng. degree (Honours) in Software Engineering from Monash University in 2022. He is currently a Ph.D. candidate with the Quantum Cloud Computing and Distributed Systems (qCLOUDS) Laboratory, School of Computing and Information Systems, University of Melbourne, Australia. His research interests include distributed privacy-preserving machine learning, Byzantine-robust distributed systems, and blockchain-enabled trust frameworks for IoT and cloud computing environments.

Ali Beikmohammadi received the B.Sc. degree in electrical engineering from Bu-Ali Sina University, Hamedan, Iran, in 2017, and the M.Sc. degree in electrical engineering from the Amirkabir University of Technology, Tehran, Iran, in 2019. He is currently working toward the Ph.D. degree in computer and systems sciences at Stockholm University, Stockholm, Sweden. His research interests include reinforcement learning, deep learning, and federated learning, both in theory and applications.

Sindri Magnússon is an Associate Professor in the Department of Computer and Systems Science at Stockholm University, Sweden. He received a PhD in Electrical Engineering from KTH Royal Institute of Technology, Stockholm, Sweden, in 2017. He was a postdoctoral researcher 2018-2019 at Harvard University, Cambridge, MA. His research interests include large-scale distributed/parallel optimization, machine learning, and control.

Rajkumar Buyya is a Redmond Barry Distinguished Professor and Director of the Quantum Cloud Computing and Distributed Systems (qCLOUDS) Laboratory at the University of Melbourne, Australia. Recognized as one of the world's most highly cited researchers in computer science and software engineering (h-index: 181; g-index: 394; 170,000+ citations), Dr. Buyya is a Fellow of IEEE, a Foreign Fellow of Academia Europaea, and a Fellow of ACM. He co-founded five major IEEE/ACM international conferences—CCGrid, Cluster, Grid, e-Science, and UCC—and served as the Chair of their inaugural meetings. He served as the founding Editor-in-Chief of the IEEE Transactions on Cloud Computing. He is currently serving as Co-Editor-in-Chief of the Journal of Software: Practice and Experience, which was established 55+ years ago.