SemanticDFL: Similarity-Aware Pull-based Personalized Decentralized Federated Learning

JAVAD DOGANI, IMDEA Networks Institute, Spain
MOSTAFA KHASTKHODAEI, Department of Computer Science and Engineering, Shiraz University, Iran
FARSHAD KHUNJUSH, Department of Computer Science and Engineering, Shiraz University, Iran
NIKOLAOS LAOUTARIS, IMDEA Networks Institute, Spain

Personalized decentralized federated learning (PDFL) seeks to tailor models to heterogeneous clients without a central coordinator, yet gossip-style mixing on large graphs dilutes minority signals and assumes any-to-any connectivity. We present *SemanticDFL*, a fully decentralized, *pull-based* personalization layer that organizes peers into a hierarchical *semantic overlay network* (SON). Each client publishes a compact top-P model signature; proximity-bounded discovery forms zones that are clustered using affinity propagation and stewarded by replica-backed super-peers that route bounded-fanout similarity queries. Clients then pull only the K most similar models for personalized aggregation, concentrating communication and computation where they matter most. We prove a lower bound that links spectral mixing and data heterogeneity to an irreducible mis-aggregation penalty for graph-oblivious, push-based overlays, thereby motivating the proposed similarity-aware pull method. A prototype and large-scale evaluation on FMNIST, Tiny ImageNet, Google Speech Commands, and 20 Newsgroups under Dirichlet and pathological splits (50–400 peers on the EU SLICES testbed) show that *SemanticDFL* improves final accuracy by 3–12% over strong decentralized personalized baselines, reaches target accuracy with ≈2.5× fewer rounds than FedAvg, and requires ≈1.3× fewer rounds than the best DPFL alternative. It adds only ≈1.7–12.6% per-round overhead across all settings while maintaining Recall@K ≈ 0.88–1.00, positioning similarity-aware pull over semantic overlays as a scalable path to high-quality personalization in decentralized FL.

CCS Concepts: • **Computing methodologies** → **Machine learning algorithms**; • **Networks** → *Peer-to-peer networks*; • **Computer systems organization** → *Distributed architectures.*

Additional Key Words and Phrases: personalized federated learning, decentralized federated learning, peer-to-peer learning, semantic overlay networks, similarity-aware aggregation, non-IID learning

ACM Reference Format:
Javad Dogani, Mostafa Khastkhodaei, Farshad Khunjush, and Nikolaos Laoutaris. 2026. SemanticDFL: Similarity-Aware Pull-based Personalized Decentralized Federated Learning. *Proc. ACM Meas. Anal. Comput. Syst.* 10, 2, Article 51 (June 2026), 37 pages. https://doi.org/10.1145/3805649

1 Introduction

Federated Learning (FL) allows multiple clients to train Machine Learning (ML) models collaboratively while keeping their data private [5, 20, 28, 38, 51, 70]. Centralized FL (CFL) approaches, exemplified by the FedAvg algorithm [51], rely on a central server for model aggregation but introduce

Authors' Contact Information: Javad Dogani, javad.dogani@networks.imdea.org, IMDEA Networks Institute, Madrid, Spain; Mostafa Khastkhodaei, Department of Computer Science and Engineering, Shiraz University, Shiraz, Iran, mostafa.khodaei@hafez.shirazu.ac.ir; Farshad Khunjush, Department of Computer Science and Engineering, Shiraz University, Shiraz, Iran, khunjush@shirazu.ac.ir; Nikolaos Laoutaris, IMDEA Networks Institute, Madrid, Spain, nikolaos.laoutaris@networks.imdea.org.

ACM 2476-1249/2026/6-ART51
https://doi.org/10.1145/3805649

scalability constraints, a single point of failure, and privacy threats if the centralized server becomes honest-but-curious [22, 73]. Decentralized FL (DFL) mitigates server bottlenecks via peer-to-peer (P2P) protocols that predominantly use push-based gossip for model exchange [25, 64]. However, conventional DFL methods typically produce a single global model, which degrades under non-IID data distributions [23, 29, 35]. In real-world heterogeneous deployments (e.g., healthcare [47], recommenders [43]), this biases performance against minority clients [41, 48, 52]. Personalized Decentralized Federated Learning (PDFL) [32, 46] addresses this by tailoring models to individual clients or semantically cohesive clusters, thereby reconciling the trade-off between generalization and specialization [14, 17, 37]. By partitioning clients into latent representation spaces, PDFL ensures that model updates are aggregated only among statistically aligned participants, enhancing personalization [19, 69]. This paradigm shift enables fine-grained model specialization, bias mitigation, and dynamic adaptation to evolving data, critical for applications requiring responsiveness to user-specific patterns [37].

Challenges. While personalization has been shown to substantially improve FL performance [49, 58], existing PDFL solutions face several fundamental challenges:

- *Topology limitations.* Current PDFL techniques impose either fixed or dynamic peer graphs, each with constraints. *Fixed* methods (e.g., DisPFL [11]) keep a static communication graph with personalized masks; when neighbors are semantically mismatched, this can lock in bias and slow mixing, harming convergence under non-IID data [6, 39]. *Dynamic* methods (e.g., stochastic neighbor-graph rewiring [72]) maintain low node degrees and periodically drop weak links but they rely on reachability to discover better peers; at scale, as we will show later, stochastic discovery remains sample-inefficient.
- *Feasibility of all-to-all connectivity.* In many deployments, assuming that *any* client can directly reach *any* other client is unrealistic: Carrier-Grade NAT (CGNAT) and enterprise firewalls break end-to-end P2P [55]; security-driven network segmentation in OT/ICS blocks cross-segment links [63]; and cross-border data-transfer controls constrain parameter/model exchange [1, 10, 59]. As a result, systems must fall back to overlay/relay mechanisms, which increase bandwidth consumption and latency because traffic is relayed via a third server [31, 56]. This further strains budgets—especially in regions with persistent rural bandwidth/connectivity gaps [18].
- *Model attenuation under gossip.* Push-based DFL studies [12, 64] propagate updates blindly to neighbors via gossip protocols, where nodes blend updates before rebroadcasting. This induces mixing-driven attenuation that is tied to the overlay's spectral gap [3, 30, 54]: on large-scale non-IID graphs, iterative averaging dilutes information, so the influence of distant yet relevant peers attenuates with hop distance and becomes asymptotically negligible [3].
- *Centralization Risks in Decentralized Systems.* Many DFL frameworks rely on semi-centralized coordinators for topology management [64], peer selection [6], or cluster cardinality specification [45], which reintroduce single-point failure, trust bottlenecks, and limit scalability.
- *Similarity Overhead.* Calculating peer similarity often involves full model comparisons, growing communication and computation costs rapidly with network size [4, 7].
- *Operational stability.* Most PDFL works optimize *who* to aggregate with, but at Internet scale, a reliable design must stay stable despite coupled hotspot pressure, peer churn, and lossy/asymmetric links; without admission control, neighbor sets become stale, and time-to-accuracy expands.

Our Proposal. We draw on the idea of *semantic overlay networks* (SONs) from early P2P search, where peers self-organize by content similarity and queries are forwarded along topical links, rather than via DHT-style exact-key lookups [9]. By design, classical DHTs (e.g., Chord [62], Kademlia [50]) enforce uniform hashing and discard geometric locality, enabling only exact-key

retrieval; consequently, they are not designed for searching most similar neighbors over high-dimensional floating-point representations (e.g., embeddings or million-parameter model states). We therefore introduce *SemanticDFL* by tailoring SON ideas to PDFL via a self-organizing SON built around three components:
i) *Compact Model Representation:* Each node keeps a sparse *semantic signature* of top-P critical parameters ranked by importance score. We find empirically that signature cosine similarity approximates full-model similarity well while reducing computation and communication costs.
ii) *Distributed SON Creation:* Initiator nodes first form bounded-size zones via proximity-limited discovery (e.g., via TTL). Within each zone, signatures drive clustering via Affinity Propagation (AP) to bridge physical and semantic levels. AP adapts the number of clusters based on pairwise similarities—avoiding centralized choices of *a priori* cluster counts [45]. SONs are formed by electing super-peers for clusters in each zone, which maintain lightweight local metadata (no global state) and route only similarity queries up/down the hierarchy, preserving decentralization and avoiding single points of failure. Zone initiators elect rotating super-peers that solely route similarity metadata; model aggregation remains end-to-end between peers. Iterating zone→cluster summaries yields an $O(\log N)$-depth overlay for low-latency similarity search.

iii) *Similarity-aware pull:* For each round, a client issues a similarity query to its zone's super-peer; queries traverse only promising branches, and the client *pulls* parameters from the K most similar peers. A lightweight control plane exchanges signatures and routes queries, while the data plane transfers models only for the selected neighbors. Replicated, rotating initiators and super-peers provide failover and resilience without introducing a central coordinator. This enables similarity-aware discovery and personalized aggregation with bounded control traffic, without the infeasibility of DHT-style exact-key routing for drifting high-dimensional model states.

SemanticDFL directly addresses the earlier challenges: proximity-bounded zones and AP-based clustering align neighbors with data similarity and adapt to drift (resolving topology mismatch); the SON with hierarchical super-peers removes any-to-any connectivity assumptions and avoids single points of failure; similarity-aware *pull* from the top-K peers prevents gossip-induced attenuation; sparse signatures bound similarity cost; and bounded zones with replica-backed control preserve stability under churn and lossy/asymmetric links.

Our Contribution. To our knowledge, *SemanticDFL* is the first fully decentralized, pull-based overlay network for PDFL. By computing similarity over pruned models, *SemanticDFL*'s pull-based mechanism achieves low bandwidth and latency overhead during both SON construction and K-nearest neighbor searches. Our theoretical analysis (Thm. 2, App. A) shows that, under clustered non-IID distributions, graph-oblivious mixing/smoothing induces a non-vanishing mis-aggregation penalty that increases monotonically with the smoothing weight and the graph's algebraic connectivity. Thus, overlays with spectral-gap mixing (e.g., expander-like topologies and Metropolis-weighted graphs) can incur higher bias as data heterogeneity grows. By addressing this, *SemanticDFL* shows fast convergence while outperforming existing PDFL studies in terms of precision, even in the most skewed distributions. We evaluate *SemanticDFL* in a real-world testbed using synthetic non-IID FMNIST and Tiny ImageNet, and real Google Speech and 20Newsgroup datasets across physical servers of the EU SLICES project [61] over the public Internet.
Our Findings. We demonstrate the following performance over a wide range of parameters:

- **Accuracy & speed.** Across four tasks and non-IID splits (Dir $\alpha \in \{0.3, 0.1\}$, Patho $\{20\%, 30\%\}$), *SemanticDFL* is best in all settings and reaches target accuracy with ~2.5× fewer rounds than FedAvg and ~1.3× fewer than the strongest PDFL baselines (pFedGraph/DFedPGP).

- **Efficiency & search fidelity.** Our pull-based design is practical at P2P scale: at N=400 on *Google Speech* with K=10%–25%, per-node SON *search* accounts for only ∼1.7–12.6% of the round time across all experiments, while Recall@K=0.88–1.00—supporting real-world deployability.
- **Scalability & robustness.** On the EU SLICES testbed with up to N=200 for 20Newsgroup, SON build/search remains sublinear with *Recall@K* = 0.88–0.99; under 0–40% churn, *SemanticDFL* accuracy drops by only ≈6–7 points, versus ≈12–14 for FedAvg/FedProx and ≈9–14 for decentralized graph/gossip baselines—making it practical to deploy at scale for PDFL.

2 Background and Related Work

2.1 Preliminaries

DFL. Let $G = (V, E)$ be the peer graph with $|V| = n$, and define the neighbor set $\mathcal{N}_i \triangleq \{\, j \in V \mid (i, j) \in E \,\}$. Client i holds data D_i and local risk $F_i(w) = \frac{1}{|D_i|} \sum_{(x,y)\in D_i} \ell(w; x, y)$, with sampling weight $p_i = \frac{|D_i|}{\sum_{j=1}^{n} |D_j|}$. DFL solves $\min_{\{w_i\}_{i=1}^n} \sum_{i=1}^n p_i F_i(w_i) \quad \text{s.t.} \quad w_i = w_j \ \forall (i, j) \in E$, implemented via local SGD and neighbor mixing. With a row-stochastic $A^{(t)}$ respecting G ($\sum_j A_{ij}^{(t)} = 1$, $A_{ij}^{(t)} = 0$ if $j \notin \mathcal{N}_i \cup \{i\}$), $v_i^{(t+1)} = w_i^{(t)} - \eta_t \nabla F_i\big(w_i^{(t)}\big), \qquad w_i^{(t+1)} = \sum_j A_{ij}^{(t)} v_j^{(t+1)}$.

PDFL. PDFL relaxes global consensus so each client learns a personalized w_i while coupling only to *similar* neighbors [40, 45, 72]. We adopt the standard neighbor-regularized (proximal / multi-task-style) PDFL instantiation; other personalization families (e.g., shared/private layer factorization, meta-learning, mixture models) are complementary and orthogonal to our focus on scalable decentralized collaborator discovery. At round t, define similarities $s_{ij}^{(t)} = \mathrm{sim}\big(w_i^{(t)}, w_j^{(t)}\big)$ (e.g., cosine on parameters/gradients) and let $S_i^{(t)} = \mathrm{TopK}_K\big(\{s_{ij}^{(t)}\}_{j\in\mathcal{N}_i}\big)$, be the index set of the $\min\{K, |\mathcal{N}_i|\}$ largest values. If $\mathcal{N}_i = \varnothing$, then $S_i^{(t)} = \varnothing$ and the support defaults to $\{i\}$. Set $s_{ii}^{(t)} := \mathrm{sim}\big(w_i^{(t)}, w_i^{(t)}\big)$ (e.g., 1 for cosine) so the self-weight is well defined. Let $c_i^{(t)} := \max_{k\in S_i^{(t)}\cup\{i\}} s_{ik}^{(t)}$. With temperature $\kappa > 0$, set row-stochastic weights (restricted to $S_i^{(t)} \cup \{i\}$) [26]:

$$W_{ij}^{(t)} = \begin{cases} \dfrac{\exp\big((s_{ij}^{(t)} - c_i^{(t)})/\kappa\big)}{\sum_{k\in S_i^{(t)}\cup\{i\}} \exp\big((s_{ik}^{(t)} - c_i^{(t)})/\kappa\big)}, & j \in S_i^{(t)} \cup \{i\}, \\ 0, & \text{otherwise.} \end{cases}$$

Define the *neighbor anchor* $m_i^{(t)} := \sum_j W_{ij}^{(t)} w_j^{(t)}$. Note that $m_i^{(t)}$ is *client-specific*: it is computed locally from i's Top-K set and is not a global model. Let $\psi_i \geq 0$ denotes the *personalization weight* that trades off local risk and attraction to the anchor. A proximal objective toward $m_i^{(t)}$ is

$$\min_{w_i} \; p_i F_i(w_i) + \frac{\psi_i}{2} \|w_i - m_i^{(t)}\|^2,$$

whose one-step proximal gradient (step size η) yields $w_i^{(t+1)} = (1 - \alpha_i)\, \widetilde{w}_i^{(t+1)} + \alpha_i\, m_i^{(t)}$, $\alpha_i = \frac{\eta\psi_i}{1+\eta\psi_i} \in [0, 1]$, with $\widetilde{w}_i^{(t+1)}$ produced by local SGD steps [13, 14]. Here, η is the proximal step size. Larger ψ_i (equivalently larger α_i) increases the influence of similar neighbors via $W_{ij}^{(t)}$.

2.2 Related Work

Serverless DFL with a single global model. Early decentralized FL (no server) replaces server aggregation with peer mixing. D-PSGD/DFedAvg [44] and expander/ring constructions [25] preserve CFL-level accuracy under IID but degrade on non-IID. Communication-oriented variants improve bandwidth use (e.g., segmented gossip) [24] or reduce inconsistency via sharper local minima and

multi-gossip steps [21, 60], yet target a single consensus model. Gossip-style mixing suffers from signal attenuation; rare but informative signals fade over hops in large, non-IID graphs [3, 30, 54].
Fully decentralized PFL via sparse/partial sharing. To personalize without a server, DisPFL [11] learns sparse masks and keeps sparse-to-sparse training across P2P exchange, cutting similarity and communication cost. DFedPGP [46] personalizes only a partial model while pushing directed gradients over asymmetric topologies, improving convergence and robustness under heterogeneity.
Fully decentralized PFL via learned collaboration graphs/neighbor selection. Vanhaesebrouck et al. [65] and FDJL [72] jointly learn personalized models and collaboration graphs in a P2P network. PENS [53] discovers semantically similar peers by probing neighbor loss and latching onto high-performers. Recent PDFL frameworks build an explicit budgeted collaboration graph: Kharrat et al. [32] optimize a bi-level objective with a constrained greedy selector, while PFedDST [15] performs decentralized peer selection under resource constraints. P2P clustered FL with adaptive neighbor matching further refines local topologies without a server [42, 43]. *For reference, server-based PFL baselines such as Ditto [37], Per-FedAvg [14], and pFedGraph [71] achieve strong personalization under a central coordinator but do not address fully decentralized operation.*

Shared/private split personalization. A complementary PFL line personalizes by *architecturally partitioning* parameters into a shared component trained collaboratively and a private component trained locally, mitigating negative transfer under heterogeneity without explicit neighbor discovery (e.g., FedPer [2], FedRep [8]). Factorized-FL [27] factorizes parameters and performs similarity matching, but its objective is *what-to-share* via parameter decomposition (typically assumes a coordinator), not scalable *who-to-collaborate-with* retrieval in large serverless P2P networks.
Beyond state-of-the-art. *SemanticDFL* departs from prior serverless DFL that pushes all clients toward one consensus model (e.g., D-PSGD, expander/ring overlays, segmented/multi-gossip variants) by learning who should collaborate with whom based on lightweight, similarity-preserving semantic signatures extracted from models, then wiring those peers into a hierarchical semantic overlay (SON) that is resilient to NATs and churn. Unlike sparse/partial-sharing PFL (DisPFL, DFedPGP), Shared/private split personalization (FedRep, Factorized-FL), and budgeted collaboration graphs (DPFL, PFedDST, PANM), *SemanticDFL* targets scalable top-K collaborator discovery in large settings. It discovers high-recall neighbors using compressed top-P signatures—avoiding probes on raw data or full models—while super-peer clustering amortizes overhead and stabilizes routing under drift. Shared/private split methods are orthogonal to the similarity-aware pull of *SemanticDFL* by applying neighbor pull on the shared subspace while keeping private parts local.

2.3 Motivation & Open Challenges

Consensus-style mixing biases minority clients under clustered non-IID. Assume clients form C latent clusters $\{C_1, \ldots, C_C\}$. After s mixing steps with a fixed row-stochastic W, the *effective* gradient at node i is $\tilde{g}_i^{(t,s)} = \sum_j (W^s)_{ij} \nabla F_j(w_j^{(t)}) = \sum_{c=1}^{C} H_{ic}^{(s)} \bar{g}_c^{(t)}$, where $\bar{g}_c^{(t)} = \frac{1}{|C_c|} \sum_{j \in C_c} \nabla F_j(w_j^{(t)})$ and $H_{ic}^{(s)} = \sum_{j \in C_c} (W^s)_{ij}$. Let $c(i) \in \{1, \ldots, C\}$ denote the index of i's cluster. Define the L_1 mis-aggregation score $\tilde{\Delta}_i^{(s)} \triangleq \sum_{c \neq c(i)} H_{ic}^{(s)} \, \|\bar{g}_c^{(t)} - \bar{g}_{c(i)}^{(t)}\|_2$. With inter-cluster gradient separation $\Gamma = \min_{c \neq c'} \|\bar{g}_c^{(t)} - \bar{g}_{c'}^{(t)}\|_2$, $\tilde{\Delta}_i^{(s)} \geq \Gamma \left(1 - H_{i,c(i)}^{(s)}\right)$. If W is primitive, $W^s \to \mathbf{1}\pi^\top$, so $H_{ic}^{(s)} \to \pi(C_c)$; if W is also doubly-stochastic (e.g., Metropolis on an undirected graph), $H_{ic}^{(s)} \to |C_c|/n$, making $\tilde{\Delta}_i^{(\infty)}$ largest for minority clusters. Moreover, standard mixing bounds yield $|H_{ic}^{(s)} - \pi(C_c)| \leq C \, \sigma_2(W)^s$ for some constant $C > 0$, tying the approach to the spectral gap. These dynamics motivate *SemanticDFL*, which replaces oblivious consensus with similarity-aware *pull* over SON, concentrating aggregation on top-K in-cluster peers and thereby suppressing cross-cluster blending.

Table 1. Qualitative comparison of decentralized (P)FL methods. ✓ = present, X = absent, – = not applicable.

Method	Personalization	Non-IID robust	Fully decentralized	Attenuation mitigation	No pre-defined knowledge	Similarity-cost efficient
DFedAvg [44]	X	X	✓	X	✓	–
Expander / Ring DFL [25]	X	X	✓	X	✓	–
Segmented Gossip [24]	X	X	✓	X	✓	–
DFedSAM / MGS [21, 60]	X	✓	✓	✓	✓	–
DisPFL [11]	✓	✓	✓	X	✓	–
Vanhaesebrouck et al. [65]	✓	✓	✓	X	✓	X
FDJL [72]	✓	✓	✓	X	✓	X
PENS [53]	✓	✓	✓	X	✓	X
DFedPGP [46]	✓	✓	✓	✓	✓	–
DPFL [32]	✓	✓	✓	X	✓	✓
Ditto [37]	✓	✓	X	X	✓	–
Per-FedAvg [14]	✓	✓	X	X	✓	–
pFedGraph [71]	✓	✓	X	X	✓	X
PFedDST [15]	✓	✓	✓	X	✓	✓
P2P Clustered FL [43]	✓	✓	✓	X	✓	X
Adaptive P2P Matching [42]	✓	✓	✓	X	✓	✓
FedRep [8]	✓	✓	X	X	✓	✓
Factorized-FL [27]	✓	✓	X	X	✓	X
SemanticDFL (ours)	✓	✓	✓	✓	✓	✓

Control/data-plane budgets bite. With model dimension M and neighborhood size $|\mathcal{N}_i|$, computing $s_{ij}^{(t)} = \mathrm{sim}(w_i^{(t)}, w_j^{(t)})$ over candidates costs $\Theta(M\,|\mathcal{N}_i|)$ time and $\Theta(M)$ bytes per comparison when parameters/gradients must be exchanged; tighter evaluation cadence to track drift multiplies both compute and traffic, and device heterogeneity turns similarity maintenance into a straggler bottleneck. On the wire, gossip-style DFL keeps control negligible but pays $\Theta(M\,|E|)$ bytes/round via pushes; naive similarity-pull trims data to $\Theta(kM)$ per round yet inflates control to $\Theta(n \times M)$ for global search. This is prohibitive and does not scale with N. Without strict time/space scoping (cadence, candidate sets, fanout), either the control plane dominates (global k-NN) or the data plane does (broad flooding), squeezing the budget for latency- and bandwidth-constrained deployments.

Generic overlays are ill-suited for metric k-NN on drifting models. DHTs and exact-key overlays support equality lookups, not metric proximity over high-dimensional, fast-drifting $w_i^{(t)}$. Order-preserving embeddings need frequent re-placement, and privacy limits exporting raw states for index upkeep. So, proximity search is either stale (hurting $S_i^{(t)}$) or costly (continuous reindexing).

2.4 Assumptions

SemanticDFL operates under the following assumptions: *Network model.* We assume a connected, partially synchronous overlay: messages may be delayed, reordered, or dropped, but are eventually delivered with bounded delay w.h.p.; links can be asymmetric, and nodes may join/leave with per-round churn rate λ_{ch}. *Overlay bounds. Model compatibility.* Clients share a common model family with layer shapes sufficient to compute a sparse signature $u_i^{(t)} = \mathrm{sig}(w_i^{(t)}) \in \mathbb{R}^M$ with $\|u_i^{(t)}\|_0 = P$ (from top-P parameter indices). They pull/aggregate all layers and use cosine similarity $\cos(u_i, u_j)$ for SON search. *Data model.* Client distributions form latent clusters with inter-cluster separation $\Gamma > 0$ in gradient space (Sec. 2.3); in App. A we use (δ, ϵ) for parameter-space separation/dispersion. *Trust model.* Honest clients: no Byzantine or Sybil behavior; participants follow the protocol.

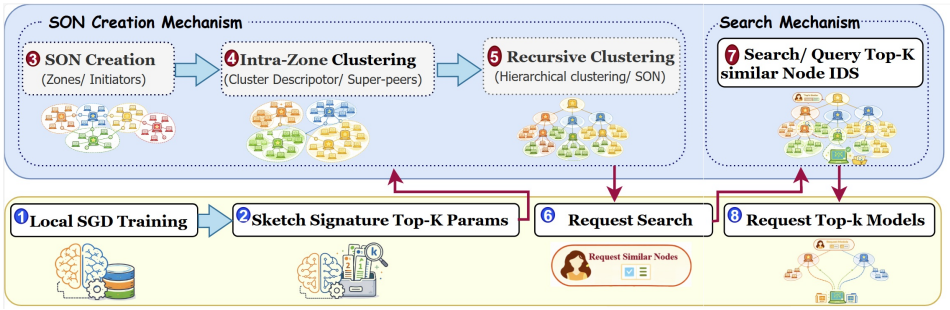


Fig. 1. Workflow of *SemanticDFL* via a Semantic Overlay Network (SON)

2.5 Scope and Limitations

SemanticDFL targets *topology optimization* and *efficient similarity-aware model dissemination* for PDFL under churn and bandwidth constraints. Out of scope are adversarial robustness (e.g., poisoning, Byzantine) and formal privacy guarantees (e.g., DP, HE, MPC). Our signatures expose only sparse magnitude information and are sent over authenticated, encrypted channels. Integrating DP noise and Byzantine-resilient mechanisms is orthogonal future work.

3 SemanticDFL

We present *SemanticDFL*, a fully decentralized, pull-based semantic overlay for PDFL. It specifies: (i) compact model signatures, (ii) fully P2P SON construction, (iii) bounded-fanout similarity search, and (iv) lightweight synchronization, maintenance under churn. These components enable a scalable, robust PDFL system that adapts to data heterogeneity in a fully P2P setting.

3.1 Design Overview

SemanticDFL transforms a flat P2P network into a shallow semantic overlay network (SON). Figure 1 illustrates the end-to-end workflow of *SemanticDFL*. Each client trains the model on its local data and then exports a fixed-cardinality sparse signature of its model. SON creation scales per zone, per-query control traffic is limited, and the full model is transferred only for the final neighbors. At bootstrap, clients elect zone initiators within TTL R that build proximity-bounded zones (size $\leq S_Z$). Within each zone, the initiator clusters peers by signature similarity and elects a replica set of super-peers per cluster. Super-peers store lightweight prototypes and member pointers and participate in the next level; no node keeps global state. Recursively clustering super-peers builds a global SON of depth $O(\log N)$. Training is pull-then-aggregate. A client refreshes its signature and issues a bounded-fanout similarity query to its top-level (level H) super-peer. The query descends iteratively to the smallest level that passes a similarity threshold, then ascends along the top branches; only signatures are routed. The client pulls full models from the final top-K matches for personalized aggregation. Robustness comes from replica sets with local repairs. The design remains fully decentralized and operates under partial synchrony, churn, and relayed paths.

3.2 Compact Model Signatures

We represent each model by a *fixed-cardinality* sparse signature of size $P \ll M$ (with M parameters). For client i and coordinate r, we maintain a magnitude-based importance score $s_{i,r}^{(t)} = \beta\, s_{i,r}^{(t-1)} + (1-\beta)\left|w_{i,r}^{(t)}\right|$, with $\beta \in [0,1)$ providing temporal smoothing ($\beta = 0$ uses raw magnitudes). Let $\mathrm{TopP}_r(\cdot)$ return the indices of the P largest entries (ties broken by index), and set $I_i^{(t)} = \mathrm{TopP}_r\left(s_{i,r}^{(t)}\right)$.

We define the *signature* as the raw masked vector $u_i^{(t)} \triangleq w_i^{[P]}(t) = \text{mask}\big(w_i^{(t)}, I_i^{(t)}\big)$, i.e., non-selected coordinates are zeroed with no further normalization. The similarity used for routing and clustering is the cosine between sparse masked vectors: $\text{sim}\big(u_i^{(t)}, u_j^{(t)}\big) = \frac{\sum_{r \in I_i^{(t)} \cap I_j^{(t)}} w_{i,r}^{(t)}\, w_{j,r}^{(t)}}{\big\|w_i^{[P]}(t)\big\|_2 \big\|w_j^{[P]}(t)\big\|_2}$, which we compute in $O(P)$ time by merging (sorted) index lists. Each signature transmission carries *indices and values* (optionally quantized), yielding a payload of $O\big(P\,(b_{\text{idx}}+b_{\text{val}})\big)$ bits.

Selecting P. At round t, define $w_i^{[P]}(t) = \text{mask}\big(w_i^{(t)}, \text{Top}_r(s_{i,r}^{(t)})\big)$. Since Top-$P$ supports are nested in P, $\cos\Big(w_i^{(t)},\, w_i^{[P]}(t)\Big)$ is non-decreasing, which enables a binary search over $P \in [1, M]$ to find the minimal P meeting a target directional similarity $\tau_{\min}$ (Alg. 1, Appendix B). Empirically (Fig. 2), pruning 84%–93% of parameters retains $\cos \geq 0.8$ in the models we evaluate.

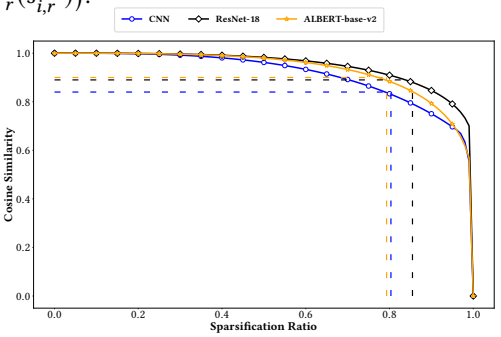


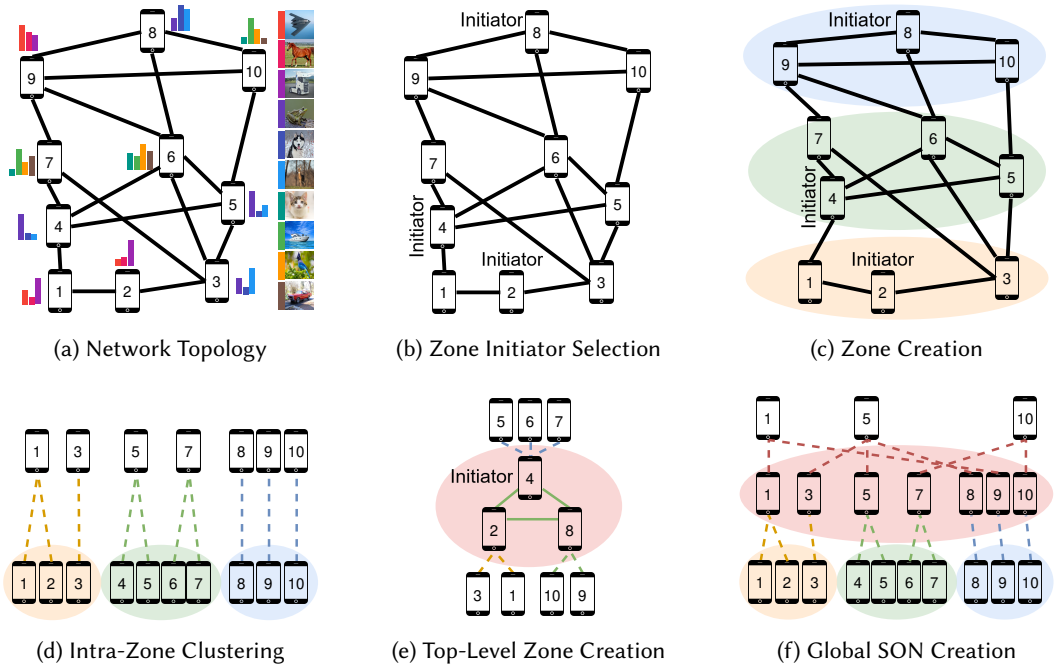
Fig. 2. Cosine between the full model and its Top-P Signature; sparsification $1-P/M$ grows.

Global P **for all clients.** To keep signatures comparable during search, we fixes a network-wide P. At bootstrap, client i computes local minimum $P_i^{\min}$ such that $\cos\big(w_i^{(t)}, w_i^{[P_i^{\min}]}(t)\big) \geq \tau_{\min}$. Zone super-peers aggregate $\{P_i^{\min}\}$ and set $P := \min\Big\{\text{Quantile}_q(\{P_i^{\min}\}),\ P_{\max}\Big\}$, ensuring at least a q-fraction of clients reach the target while keeping a uniform signature length. During training, the *values* in $u_i^{(t)}$ always reflect the current $w_i^{(t)}$; the *indices* $I_i^{(t)}$ refresh every rounds.

3.3 SON Creation

SemanticDFL constructs SON in three stages using the same raw Top-P (magnitude-based) signatures as both the input to clustering and the key for routing. First, clients run a *zone-local* election to select *initiators* that assemble proximity-bounded zones via TTL probes and deterministic splitting when over capacity. Second, each zone's initiator collects member signatures and performs *in-zone clustering*, and elects a small *replica set* of *super-peers* that maintain lightweight descriptors and serve as ingress points. Third, *recursive clustering* of super-peers' prototypes builds higher levels with bounded fan-out, yielding a global SON of expected depth $O(\log N)$. Nodes keep only parent/child descriptors—no global snapshot—so per-zone costs stay minimal, while replica sets and initiator rotation provide local failover under churn. Figure 3 illustrates this from an overlay-free state (a), to zone formation (b–c), intra-zone clustering and super-peer election (d), higher-level clustering and query routing (e–f). The complete algorithmic design and implementation details of the SON creation process are provided in Appendix B.

3.3.1 Distributed Zone Formation. We form proximity-bounded *zones* once per overlay epoch via a zone-local initiator election followed by TTL-bounded discovery. Zones have bounded TTL radius R and size $|Z| \leq S_Z$; each node caps its neighbor fanout by $d_{\max}$. At epoch e, each node p computes a verifiable random score $\sigma_p = \text{VRF}_p(\text{seed}(e-1))$; within its R-hop neighborhood, the node with the minimum (σ_p, id_p) becomes the initiator (ties by ID), ensuring a dispersed set of leaders at granularity R. Each initiator floods a PROBE with TTL= R; a NOT_ASSIGNED peer adopts the *first* valid probe it sees (first-wins), registers with that initiator, and forwards while TTL> 0, yielding disjoint, proximity-aligned membership with per-zone message cost $O(S_Z)$ and latency $O(R)$ hops. Peers that are already ASSIGNED emit adjacency notifications to both initiators



(a) Network Topology (b) Zone Initiator Selection (c) Zone Creation

(d) Intra-Zone Clustering (e) Top-Level Zone Creation (f) Global SON Creation

Fig. 3. SON construction process. (a) Initial network topology with non-IID data (b) Zone initiator selection based on network structure (c) Zone Formation around initiators (d) Intra-zone clustering by signature similarity (e) high-level zones construction (f) Final global SON enabling scalable P2P learning.

so neighboring zones discover their boundary without extra probing. After a diameter timeout of $R \cdot t_a$, each initiator checks its membership cap; if $|\mathcal{Z}_I| > S_Z$, it performs a deterministic *split* (e.g., BFS order): keep the first S_Z members and hand the remainder to the next-lowest VRF score in-zone, which re-probes to seed a new zone. This mechanism provides low latency via TTL scoping, load balancing via the hard cap S_Z, and deterministic splitting, and adaptivity via initiator rotation and purely local repairs under churn. Figures 3b and c illustrate initiator selection (nodes 2,4,8) and final zones (1,2,3, 4,5,6,7, 8,9,10) with geographical shading.

3.3.2 Intra-Zone Clustering. Within a zone of size $n_z \leq S_Z$, the initiator issues a lightweight `SigProbe` to collect the raw signatures $\{u_i\}$. It then runs *Affinity Propagation* (AP) [16] under cosine similarity, allowing the number of clusters c to be discovered from the data rather than fixed a priori. The control overhead to gather signatures is $O(n_z P)$ bytes; constructing the $n_z \times n_z$ similarity matrix costs $O(n_z^2 P)$ operations, which remains tractable under the hard zone cap. AP returns clusters $\mathcal{C} = \{C_1, \ldots, C_c\}$ with *exemplars* $e_j \in C_j$. For routing and summary, we maintain a *prototype* in the same signature space $\bar{u}_j = TopP(\frac{1}{|C_j|} \sum_{i \in C_j} u_i)$. For each C_j, the initiator elects a replica set $R_j \subseteq C_j$ of *super-peers* and publishes a compact descriptor

$$CD_j = \big(j,\ e_j,\ \bar{u}_j,\ R_j,\ |C_j|,\ \text{zoneID}\big).$$

Super-peers keep only a lightweight state (the descriptors and member pointers) and serve as ingress points for updates and search, pruning queries using $\bar{u}_j$ before forwarding to members. After CD_j dissemination to R_j and member notifications, the zone is ready for the recursive SON

construction in the next stage. As illustrated in Fig. 3d, peers $\{4, 5, 6, 7\}$ in the green zone form clusters $\{4, 5\}$ and $\{6, 7\}$, with peers 5 and 7 as (sole) super-peers in this toy example ($|R_j| = 1$).

3.3.3 Inter-Zone Recursive Clustering. Building on the intra-zone results, *SemanticDFL* constructs level ℓ by *recursively clustering the prototype signatures of clusters from level* $\ell-1$. Zone initiators elected at level $\ell-1$ act as ordinary participants in level ℓ: each initiator contributes a *set* of compact prototype signatures $\{\bar{u}_j\}$, where each $\bar{u}_j$ represents one lower-level cluster in its level $\ell-1$ zone. The only distinction between the first level($\ell = 1$) and higher levels ($\ell > 1$) is therefore explicit: at $\ell = 1$, each initiator contributes the intra-zone cluster prototypes $\{\bar{u}_j\}$; for $\ell > 1$, it contributes the prototypes it owns from level $\ell - 1$. Clustering at level ℓ proceeds identically to the in-zone case—Affinity Propagation under cosine on $\{\bar{u}_j\}$—yielding meta-clusters and a new replica set of *level-ℓ super-peers*. Each node maintains only parent/children descriptors without a global snapshot; state is confined to parent/children pointers, reducing failure blast radius. The recursion continues until the number of level-ℓ super-peers falls below a small threshold (we reuse S_Z), producing a hierarchy whose depth is $O(\log N)$ under bounded fanout. Intuitively, each level abstracts *groups of groups*: level-ℓ+1 initiators summarize their child clusters, allowing the overlay to operate at coarser granularity while preserving semantic structure for routing. The resulting overlay supports efficient search: queries ascend to the smallest ancestor whose prototype passes the similarity filter and then descend along the selected branches to the most relevant child clusters. Figures 3e–f illustrate the transition beyond the initial zones, where level 1 initiators (nodes 2, 4, 8) become vertices of a level-2 overlay; node 4 is elected as initiator and reclusters the prototypes advertised by nodes 2 and 8 together with those of its own child clusters, yielding semantically coherent meta-clusters.

3.4 SON-Based Similarity Search

PDFL on non-IID data requires *efficiently* finding peers with semantically compatible models. Push-based unstructured gossip incures prohibitive traffic and unreliable convergence at scale. *SemanticDFL* performs a topology-aware similarity search that *starts at the highest level of the hierarchy*, descends with bounded fan-out using only fixed-cardinality signatures, and aggregates results bottom-up to the first layer. Peer i forms a query $Q_i = \langle u_i^{(t)}, K, \texttt{qid} \rangle$ with signature $u_i^{(t)}$, target size K, and a unique identifier `qid`. The query is forwarded (via parent pointers) to a replicated *top-level* super-peer on level H, which keeps only children descriptors $\{CD_j\}$ including signature prototype $\{\bar{u}_j\}$ for its domain, not global state. The search process is performed in a top-down manner: At each visited super-peer v on level ℓ (starting from $\ell = L$ down to $\ell = 1$), let $C(v)$ denote its child super-peers at level $\ell - 1$ with prototypes $\{\bar{u}_c\}_{c \in C(v)}$. The node computes scores $\left\{\text{sim}(u_i^{(t)}, \bar{u}_c)\right\}_{c \in C(v)}$, keeps the top B_ℓ children whose scores exceed a round-level threshold $\tau^{(t)}$: $\hat{s}_{ij}^{(t)} \geq \tau^{(t)}$, forwards Q_i in parallel to those B_ℓ children. The procedure repeats until either (i) a leaf level is reached, or (ii) no branch passes the filter (then the *best* B_ℓ are kept to avoid dead-ends). During descent, only *signatures and cluster descriptors* are transmitted. At a leaf cluster $\mathcal{L}$, each member $j \in \mathcal{L}$ computes $\text{sim}_u(u_i^{(t)}, u_j^{(t)})$ and returns a local candidate list $R_{\mathcal{L}} = \{(j, \text{sim}_u(u_i^{(t)}, u_j^{(t)}))\}$ to its parent. Each internal super-peer merges the candidate lists received from children via a size-K max-heap keyed by similarity (deduplicating peer-IDs) to produce a condensed top-K list. These lists propagate *upward* to the *Top-level* super-peer(s) (level H), which returns the final top-K *peer IDs/addresses* to i.

All routing and pruning stages exchange *signatures* only; full models are transferred *only* at the end; the data-plane cost is thus incurred once and only for the selected neighbors. This design accepts minor, transient clustering inaccuracies due to churn/model drift to keep maintenance costs

low; correctness is recovered by the end-to-end ranking at leaves and the top-K merge. Figure 4(a–c) shows the similarity search process, where a query is initiated by peer 3. In (a), 3 scores the level-1 prototypes and forwards to the single best branch, the super-peer rooted at node 5. In (b), the beam widens to the top three branches—super-peers 3, 5, and 8—and the query is forwarded to each. In (c), each branch descends to its most relevant leaf cluster: 3 reaches cluster 3, 5 reaches 4,5, and 8 reaches 8; members in these leaves compare their signatures with 5's. In (d), leaves 3, 4,5, and 8 send their local candidates upstream. In (e), super-peers 3, 5, and 8 deduplicate and retain per-branch top-K lists. In (f), super peer 5 receives the condensed lists from super peers 3, 5, and 8, merges them into the global top-K, and only then pulls full models from the selected peers. Throughout (a–f), only signatures flow during routing; full models move at the end.

Cost. At a visited super-peer on level ℓ, scoring m_ℓ child prototypes costs $O(m_\ell P)$ and forwarding expands to at most B_ℓ children. If v_ℓ nodes are visited at level ℓ (with $v_\ell \le \prod_{r=\ell+1}^{H} B_r$), the total signature work per query is $O\Big(P \sum_{\ell=1}^{H} v_\ell m_\ell\Big)$ and messages are $O\Big(\sum_{\ell=1}^{H} v_\ell B_\ell\Big)$. Under bounded fanout $F = \max_\ell m_\ell$, beam $B = \max_\ell B_\ell$, and depth $H = O(\log N)$, this becomes $O(PFBH)$ signature comparisons and $O(BH)$ messages. Bottom-up merging adds $O(K \log K \cdot H)$, and only the final K pulls move full models, i.e., $O(KM)$ once. We quantify the SON control-plane cost by a concrete example (200 Mbps ≈ 25 MB/s). Using ResNet-18 (≈11.7M params ⇒ ∼47 MB/model in FP32), a Top-P signature contains PM entries; encoding each entry with a 4-byte index and a 2-byte value gives $\approx 6(PM)$ bytes, i.e., $S_{\text{sig}} \approx 6.7$ MB at P=10% (and scales linearly with P). Using the dataset-specific operating points from our runs (P=10.8%M for Tiny-ImageNet and P=13.3%M for Google Speech) yields $S_{\text{sig}} \approx 7.24$ MB and 8.91 MB, respectively. With depth H≈3, beam B=2, visiting ∼ 7 super-peers and probing ∼ 4 leaves of size $n_{\mathcal{L}}$=2, a query fetches $(7 + 4n_{\mathcal{L}})$=15 signatures, i.e., ≈ 109 MB (≈ 4.4 s) for Tiny-ImageNet and ≈ 133 MB (≈ 5.3 s) for Google Speech. Pulling and uploading (serving others) K=10 full models costs 935 MB (≈ 38.5 s), so search/control traffic remains smaller than the mandatory model transfers. When including local training, search accounts for at most ≈ 12.59% of the full FL round for all models (See Table 6). This overhead lets aggregation from Top-K most similar neighbors rather than arbitrary graph neighbors, yielding higher accuracy in fewer rounds (Table 3).

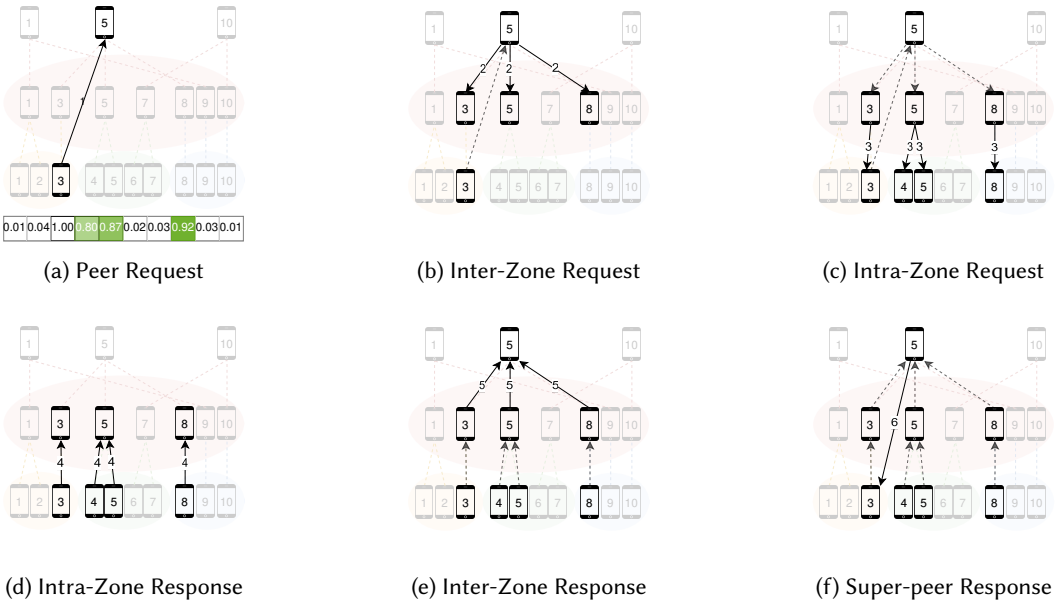
3.5 Adaptive Thresholding for Top-K

Model representations drift across rounds, making fixed similarity thresholds unstable. We therefore maintain a *single, round-level* similarity threshold $\tau^{(t)}$ that is used *uniformly at all overlay levels* for routing and candidate inclusion. In round t, any comparison (prototype or leaf) with $\text{sim}(\cdot,\cdot) \ge \tau^{(t)}$ is accepted; beam caps $\{B_\ell\}$ still bound branching. To keep search efficient while preserving recall, we adapt $\tau^{(t)}$ using the observed *global* acceptance rate. Let $\widehat{a}^{(t)}$ be the fraction of scored children (across all visited super-peers in round t) whose similarity to the query signature $u_i^{(t)}$ meets the threshold: $\widehat{a}^{(t)} = \frac{\#\text{ accepted edges with sim}\ge\tau^{(t)}}{\#\text{ scored edges}}$. We update the threshold with a small-step controller:

$$\tau^{(t+1)} \leftarrow \text{clip}_{[-1,1]}\Big(\tau^{(t)} + \eta_\tau \left(\widehat{a}^{(t)} - \phi\right) - \gamma\left(1 - \rho_i^{(t)}\right)\Big),$$

where $\phi \in (0,1)$ is the target global acceptance (roughly controlling the effective beam), η_τ is the step size, and $\rho_i^{(t)} = \cos\big(u_i^{(t)}, u_i^{(t-1)}\big)$ provides drift-aware slack (larger drift ⇒ lower $\tau^{(t+1)}$).

We initialize $\tau^{(0)}$ to a streaming estimate of the $(1-\phi)$ similarity quantile from recent probes (aggregated across levels), then refine it online via the controller above. After the bottom-up merge, if the total distinct candidates $C^{(t)} < K$, we relax the *same* global threshold and retry the descent under the same beam caps $\tau^{(t)} \leftarrow \tau^{(t)} - \delta$ (e.g., δ=0.02), preferentially re-expanding branches



(a) Peer Request (b) Inter-Zone Request (c) Intra-Zone Request

(d) Intra-Zone Response (e) Inter-Zone Response (f) Super-peer Response

Fig. 4. Six-step illustration of the search process in *SemanticDFL*.

where few comparisons previously passed. The process stops when $C^{(t)} \geq K$ or a small visit budget is reached. The same $\tau^{(t)}$ is then used throughout the hierarchy for the remainder of round t.

3.6 Synchronization and Freshness

We use a soft barrier with bounded staleness in a near-synchronous setting with roughly equal-capability peers. In each round t, a peer issues a similarity query and aggregates when either (i) K replies arrive, or (ii) a timeout Δ_t fires, whichever occurs first. Late replies from rounds $t' \in [t - \Delta_{\max},\ t)$ may be used if their signatures satisfy the freshness guard $\mathrm{sim}\big(u_i^{(t)}, u_j^{(t')}\big) \geq T_{\mathrm{fresh}}$. This provides consistent round boundaries while preserving the pull-based design.

3.7 Dynamics and Maintenance

Failure Resilience. *SemanticDFL* uses replica sets, lease-based leadership, and local repairs, without any global coordinator. Each zone elects an initiator that maintains zone descriptors and membership. The initiator replicates its state to a replica set R_Z of size r. Leadership is governed by short leases; upon lease expiry or missed renewals, replicas elect a successor deterministically (e.g., by (VRF, id) order) and restore state from their copies. Within each zone, every cluster C_j has a super-peer replica set R_j that stores the cluster descriptor CD_j (exemplar, prototype $\bar{u}_j$, member pointers). If a super-peer fails, a replica in R_j takes over immediately, and the zone initiator backfills to keep $|R_j| = r$. Live replicas also balance query load. Parent/child pointers are kept in replica sets; failures are healed locally by reassigning a replica and notifying only adjacent nodes (parent and children). No global state is required, and repairs do not affect unrelated zones or clusters. With replication $r \geq 2$, single-node failures are masked, and the overlay remains operational with minimal disruption.
Join, Leave, and Dropout. A newcomer attaches to a nearby peer, which forwards `join` to the nearest ancestor super-peer via parent pointers. Placement is top-down: at each level ℓ, compare u_{new} to child prototypes $\{\bar{u}_j\}$, keep the best branch with $\mathrm{sim}_u(u_{\mathrm{new}}, \bar{u}_j) \geq \tau^{(t)}$ (else take the best

child), and descend to a leaf cluster $\mathcal{L}$. The chosen cluster updates CD_j (membership, size) and adjusts R_j to maintain replication r. For graceful leave, the peer notifies its cluster super-peer; CD_j is updated and, if the leaver was in R_j, a replacement is promoted from C_j to keep $|R_j|$=r. Links are pruned locally; no global state is touched. If a peer disappears (dropout), lease expiry triggers local repair: promote a replica in R_j, refresh CD_j, and patch only adjacent parent/child pointers.

3.8 Analysis: Practical Scalability, Efficiency, and Robustness of SemanticDFL

SemanticDFL is a fully decentralized, pull-based semantic overlay for PDFL: clients expose compact, top-P model signatures, self-organize into proximity-bounded zones with replica-backed super-peers, and recursively cluster zone prototypes into a shallow hierarchy. Bounded-fanout, cosine-based search routes *signatures only*; the client then pulls the top-K full models for personalized aggregation. This yields **(i) *scalability*** via caps on zone size S_Z and per-level fan-out B_ℓ, giving depth $O(\log N)$ while $P \ll M$ keeps control traffic compact and limits full transfers to the final top-K; **(ii) *load balancing*** as TTL-scoped formation and hard caps avoid hot spots, deterministic splits bound growth, and replica-backed super-peers share routing/merge work with beam caps B_ℓ; **(iii) *efficient search*** through pruning—prototype filtering with thresholds $\tau^{(t)}$, so only signatures flow until the last hop; **(iv) *robustness*** using short leases and heartbeats for fast re-elections with repairs confined to parent/child pointers (no global state); and **(v) *adaptation to non-IID*** by clustering on signature similarity, preserving proximity and regulating acceptance via the global threshold $\tau^{(t)}$, after which the top-K pull personalizes aggregation over the most relevant neighbors.

4 Theoretical Analysis

4.1 Setup and assumptions

Let $F_i : \mathbb{R}^d \to \mathbb{R}$ be client i's objective and $w_i^{(t)}$ its parameter at round t. We impose standard conditions and make explicit the three sources of gradient mismatch: alignment within the *ideal* neighbor set, selection errors due to approximate retrieval, and (iii) staleness/partial replies.

Assumption 1 (Smoothness)**.** Each F_j is L_s-smooth: for all x, y, $\|\nabla F_j(x) - \nabla F_j(y)\| \le L_s\|x - y\|$.

Assumption 2 (Stochastic gradients)**.** For all w, $\mathbb{E}[g_j(w;\xi)] = \nabla F_j(w)$, $\mathbb{E}\left\|g_j(w;\xi) - \nabla F_j(w)\right\|^2 \le \sigma^2$, and $\mathbb{E}\left\|g_j(w;\xi)\right\|^2 \le \mathrm{G}^2$.

At round t, client i (i) runs E_{loc} local SGD steps on F_i to obtain $\tilde{w}_i^{(t+1)}$, (ii) uses the SON to retrieve a Top-K neighbor set $S_i^{(t)}$ by signature similarity, then *pulls* the neighbors' post-local-update models $\{\tilde{w}_j^{(t+1)}\}_{j\in S_i^{(t)}}$, forms the anchor $m_i^{(t)} = \sum_{j\in S_i^{(t)}} W_{ij}^{(t)}\, \tilde{w}_j^{(t+1)}$, $\sum_{j\in S_i^{(t)}} W_{ij}^{(t)} = 1$, and (iii) mixes $w_i^{(t+1)} = (1-\alpha_i)\tilde{w}_i^{(t+1)} + \alpha_i m_i^{(t)}$. Thus, the SON affects optimization only through the quality/staleness of the retrieved set $S_i^{(t)}$ (and hence the anchor), not through any gradient-oracle at neighbors. Let $O_i^{(t)} := \mathrm{TopK}\big(\{\mathrm{sim}_w(w_i^{(t)}, w_j^{(t)})\}_{j\neq i}\big)$ denote the ideal (oracle) Top-K neighbors for client i at round t under the cosine similarity in weights. To capture imperfect retrieval and staleness, we bound the deviation from using $S_i^{(t)}$ instead of $O_i^{(t)}$ via a mismatch term evaluated at a common reference point. This is *not* a protocol requirement (no neighbor computes gradients at $w_i^{(t)}$); it is used only for analysis. Define the neighbor-averaged gradient $\mu_S(w) := \frac{1}{|S|}\sum_{j\in S}\nabla F_j(w)$.

Remark 1. (Surrogate mismatch used for analysis). *SemanticDFL* performs one-step proximal update toward the pulled anchor: it forms $m_i^{(t)} = \sum_{j\in S_i^{(t)}} W_{ij}^{(t)}\tilde{w}_j^{(t+1)}$ and mixes $w_i^{(t+1)} = (1-\alpha_i)\tilde{w}_i^{(t+1)} + \alpha_i m_i^{(t)}$. To isolate the effects of *neighbor retrieval quality* and *staleness*, we analyze the equivalent surrogate descent $w_i^{(t+1)} = w_i^{(t)} - \eta g_i^{(t)}$ and bound how replacing the oracle set $O_i^{(t)}$ with the

retrieved set $S_i^{(t)}$ perturbs the update via the mismatch $\|\nabla F_i(w) - \mu_{S_i^{(t)}}(w)\|$ at a common reference point. Standard proximal-gradient arguments yield the same stationarity bound up to constants (absorbing α_i and ψ_i); we keep the surrogate to make the dependence on r_t and $\Delta_{\max}$ explicit.

Assumption 3 (Alignment of ideal neighbors)**.** The within-oracle alignment is bounded: $\zeta_i^{\text{align}} := \sup_w \|\nabla F_i(w) - \mu_{O_i(w)}(w)\| < \infty$.

Assumption 4 (Selection quality via recall@K)**.** At round t, the SON returns $S_i^{(t)}$ with recall $r_t \in [0,1]$ relative to $O_i(w_i^{(t)})$, i.e., $|S_i^{(t)} \cap O_i(w_i^{(t)})| = r_t K$.

Assumption 5 (Staleness guard and active fan-in)**.** Replies older than $\Delta_{\max}$ rounds are discarded. The random fan-in satisfies $K_t \geq 1$ almost surely and $K_{\text{eff}} := (\mathbb{E}[1/K_t])^{-1} \in (0, K]$.

Assumptions 1–5 are standard in nonconvex stochastic optimization and P2P learning, we make the *selection* (Assumption 4) and *staleness* (Assumption 5) explicit to reflect SON retrieval and timing.

4.2 Convergence with explicit mismatch decomposition

We quantify the deviation between the used average $\mu_{S_i^{(t)}}(w)$ and the target $\nabla F_i(w)$ through three components: ζ_i^{align} reflects intrinsic heterogeneity even under oracle neighbors, approximate retrieval with recall r_t, and reply staleness/partial fan-in.

Lemma 1 (Selection deviation under recall@K)**.** *Under Assumptions 2 and 4, for any w, $\left\|\mu_{S_i^{(t)}}(w) - \mu_{O_i(w)}(w)\right\| \leq 2(1-r_t)\,\mathrm{G}$.*

Proof. The set $S_i^{(t)}$ differs from $O_i(w)$ by at most $(1-r_t)K$ elements. Replacing at most $(1-r_t)K$ vectors in a K-average changes the mean by at most $\frac{(1-r_t)K}{K} \cdot 2\mathrm{G} = 2(1-r_t)\mathrm{G}$ using the triangle inequality and $\|\nabla F_j(w)\| \leq \mathrm{G}$ from Assumption 2. □

Lemma 2 (Bias–variance for the aggregated estimator)**.** *Conditioned on $w_i^{(t)}$ and $S_i^{(t)}$, $\mathbb{E}[g_i^{(t)}] = \mu_{S_i^{(t)}}(w_i^{(t)})$ and $\mathbb{E}\|g_i^{(t)} - \mu_{S_i^{(t)}}(w_i^{(t)})\|^2 \leq \sigma^2/K_t$. So, $\mathbb{E}\|g_i^{(t)} - \mathbb{E}[g_i^{(t)}]\|^2 \leq \sigma^2\,\mathbb{E}[1/K_t] = \sigma^2/K_{\text{eff}}$.*

Proof. Unbiasedness and independence across neighbors under Assumption 2 yield the mean; averaging K_t i.i.d. terms gives the variance bound. □

Lemma 3 (Staleness deviation)**.** *Under Assumptions 1 and 5, the SON may return (and client i may pull) models that are up to $\Delta_{\max}$ rounds stale relative to the round-t reference. Let $\Delta_{ij}^{(t)} \in \{0, 1, \ldots, \Delta_{\max}\}$ denote the stalenes of the pulled model from neighbor j and used in forming $m_i^{(t)}$. Then for a universal constant c_Δ, the induced surrogate mismatch satisfies $\zeta_i^{\text{stale}} \;\leq\; c_\Delta L_s \eta \Delta_{\max}\Big(\sigma/\sqrt{K_{\text{eff}}} + 2\mathrm{G}\Big)$.*

Proof. By L_s-smoothness, for any j and any $\delta \leq \Delta_{\max}$, $\|\nabla F_j(w_i^{(t)}) - \nabla F_j(w_i^{(t-\delta)})\| \leq L_s\|w_i^{(t)} - w_i^{(t-\delta)}\|$. Telescoping the updates over at most $\Delta_{\max}$ rounds gives $w_i^{(t)} - w_i^{(t-\delta)} = -\eta \sum_{s=t-\delta}^{t-1} g_i^{(s)}$, hence by Jensen and Cauchy–Schwarz, $\mathbb{E}\left\|w_i^{(t)} - w_i^{(t-\delta)}\right\| \leq \eta \sum_{s=t-\delta}^{t-1} \mathbb{E}\|g_i^{(s)}\| \leq \eta\Delta_{\max}\sqrt{\mathbb{E}\|g_i^{(s)}\|^2}$. Using Assumption 2 and Lemma 2, $\mathbb{E}\|g_i^{(s)}\|^2 \leq 2\|\mu_{S_i^{(s)}}(w_i^{(s)})\|^2 + 2\,\mathrm{Var}(g_i^{(s)}) \leq 2\mathrm{G}^2 + 2\sigma^2/K_{\text{eff}}$, so $\mathbb{E}\left\|w_i^{(t)} - w_i^{(t-\delta)}\right\| \leq \eta\Delta_{\max}\left(\frac{\sigma}{\sqrt{K_{\text{eff}}}} + 2\mathrm{G}\right)$, where constants are absorbed into c_Δ. □

We compress alignment, selection, and staleness into one quantity used by the descent bound. By $(a+b+c)^2 \leq 3(a^2+b^2+c^2)$, define

$$\zeta_i^2 \;:=\; 3\Big((\zeta_i^{\text{align}})^2 + (\zeta_t^{\text{sel}})^2 + (\zeta_i^{\text{stale}})^2\Big), \qquad \zeta_t^{\text{sel}} := 2(1-r_t)\mathrm{G}, \tag{1}$$

where r_t is the (expected) recall@K of the SON retrieval at round t.

Lemma 4 (One-step descent with mismatch). *Under Assumption 1 and $\eta \le 1/(2L_s)$,*

$$\mathbb{E}\Big[F_i(w_i^{(t+1)})\Big] \le \mathbb{E}\Big[F_i(w_i^{(t)})\Big] - \frac{\eta}{2}\,\mathbb{E}\Big\|\nabla F_i(w_i^{(t)})\Big\|^2 + \eta\,\mathbb{E}\Big\|\nabla F_i(w_i^{(t)}) - \mu_{S_i^{(t)}}(w_i^{(t)})\Big\|^2 + \frac{\eta^2 L_s}{2}\cdot\frac{\sigma^2}{K_{\text{eff}}}.$$

Proof. Apply smoothness to $w_i^{(t+1)} = w_i^{(t)} - \eta g_i^{(t)}$, take expectations, use $\eta \le 1/(2L_s)$, and plug in Lemma 2. □

Lemma 5 (Bounding the mismatch term). *Under Assumptions 3–5, $\mathbb{E}\Big[\Big\|\nabla F_i(w_i^{(t)}) - \mu_{S_i^{(t)}}(w_i^{(t)})\Big\|^2\Big] \le \zeta_i^2$, with ζ_i defined in (1).*

Proof. The triangle inequality with the three components and Jensen's inequality yield the factor 3. □

Theorem 1 (Stationarity rate with explicit error floor). *Suppose Assumptions 1–5 hold and choose a constant step size $\eta \le 1/(2L_s)$. Then for any $T \ge 1$,*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\Big\|\nabla F_i(w_i^{(t)})\Big\|^2 \;\le\; \frac{2\big(F_i(w_i^{(0)}) - F_i^\star\big)}{\eta T} \;+\; 2\eta\,\frac{\sigma^2}{K_{\text{eff}}} \;+\; 8\,\zeta_i^2. \tag{2}$$

Proof. Sum Lemma 4 over $t = 0, \dots, T-1$, telescope F_i, and apply Lemma 5 and $\mathbb{E}[1/K_t] = 1/K_{\text{eff}}$. Rearranging gives (2). (Full proof in Appendix C.) □

Remark 2 (Choosing the step size). Setting $\eta = \Theta(1/\sqrt{T})$ yields the usual $O(1/\sqrt{T})$ stationarity rate for nonconvex stochastic optimization, up to an additive floor $8\zeta_i^2$. Larger K and better timeout handling increase K_{eff} and shrink the stochastic term; stronger clustering (larger P, wider beam), higher recall, and tighter staleness bounds reduce ζ_i.

Remark 3 (Gradients computed at neighbors' own parameters). If neighbors respond with $\nabla F_j(w_j^{(t)})$ instead of $\nabla F_j(w_i^{(t)})$, then an additional disagreement term bounded by $L_s\,\mathbb{E}\|w_j^{(t)} - w_i^{(t)}\|$ enters Lemma 4, which translates into an extra $O(L_s^2\eta^2)$ additive term in (2) under bounded steps. This strengthens the motivation for computing gradients at the requester's parameter.

4.3 Control traffic, creation latency, and memory

We now formalize the control-plane costs of SON creation and maintenance. At each level ℓ of the hierarchy (with cap S_Z nodes per zone, $H = \lceil \log_{S_Z} N \rceil$ levels), each node exchanges (i) overlay headers of size S_H, (ii) signatures of size S_P, and (iii) CDs of size S_D. Let $\bar{d}$ be the average overlay degree, $\overline{C}$ the average number of clusters per zone, and r the replication factor for super-peers.

Proposition 1 (Per-node control cost is $O(1)$; total $O(N)$). *Across all levels, $C_{\text{tot}} = \Theta\Big(N\big(\bar{d}\,S_H + S_P + \overline{C}\,S_D\big)\Big)$, $C_{\text{avg}} := \frac{C_{\text{tot}}}{N} = \Theta\big(\bar{d}\,S_H + S_P + \overline{C}\,S_D\big) = O(1)$. with $S_Z = \Theta(N)$ (as in our runs), So $H = O(1)$. A root-level synchronization among F super peers is $\Theta(F^2 S_D)/\Theta(F S_D)$ over a mesh/tree; remains lower order for bounded F.*

Proof. At any level, the per-node header and signature messages contribute $N\bar{d}S_H + NS_P$ bytes. Cluster descriptors contribute $(N/S_Z)\cdot\overline{C}S_D$ bytes; super-peer replication multiplies by replication factor $r = O(1)$. With Summing across $H = O(1)$ levels in terms of constants (since S_Z and the per-level factors are capped) gives the expression. The root term is additive and lower order for fixed F. □

Let D_Z denote the diameter of a zone-level overlay. Under bounded-degree random overlays we have $D_Z = O(\log S_Z)$. Flood/probe and clustering at each level proceed in parallel across zones (See full derivation with message-type constants in Appendix D.1).

Proposition 2 (Logarithmic Overlay creation latency)**.** *With per-zone bounded computation (fixed S_Z and P), the wall-clock time per level is $O(D_Z) + O(1) = O(\log S_Z)$. $H = \lceil \log_{S_Z} N \rceil$ and the final super peer sync over a shallow tree adds $O(\log F)$ rounds, the end-to-end build time is $T_{\text{build}} = \Theta\big(\log_{S_Z} N\big) + \Theta(\log F) = \Theta(\log N)$ for fixed S_Z, $F \ll N$. See details in Appendix D.2.*

Proposition 3 (Worst-case super-peer memory is logarithmic in N)**.** *If a node acts as a super-peer at all H levels, storing up to S_Z child entries per level (each entry consisting of one signature and one descriptor, with replication factor r), then $M_{\max} \le H S_Z \big(S_P + r S_D\big) = O\big((S_P + S_D)\, S_Z \log N\big)$, which is typically far below caching K full model copies.*

Proof. Summing S_Z entries per level across H levels yields $HS_Z(S_P + rS_D)$. Since $H = \Theta(\log N)$ and S_Z, r are fixed caps, the bound follows. (See Appendix D.3 for the complete procedure). □

5 Experimental Evaluation

5.1 Experimental Setup

Datasets And Models. We evaluate *SemanticDFL* on four heterogeneous benchmarks to demonstrate applicability across *modalities* and model *inductive biases/parameter geometries*: (i) *CNN* on **FMNIST** [68] (two 5×5 conv layers, one 512-unit FC, 10-class output); (ii) *ResNet-18* on **Tiny ImageNet** [34] (stem adapted for 64×64 inputs, 200 classes); (iii) *ResNet-18* on **Google Speech** [67] (log-mel spectrogram front-end); (iv) *Albert-base-v2* on **20Newsgroup** [33] (12-layer Transformer encoder, 768-dim hidden, 12 heads). We consider two non-IID families: Dirichlet splits with $\alpha \in \{0.3, 0.1\}$ and pathological label-skew where each client observes only {20%, 30%} of classes (following [23]). For evaluation, each client is tested both on a held-out local test split with the same partition proportions as its training data. These settings amplify client-specific support mismatch and drift—precisely where the objective is the *personalized DFL*. Extended experiments, including evaluating each personalized model on a shared global test set, are in Appendix E.

Parameter/Hyperparameters selection and stability knobs. *SemanticDFL* exposes a small set of knobs with clear roles. *Auto/online*: the routing threshold $\tau^{(t)}$ is auto-tuned (Sec. 3.5) and the signature size P is set by the cosine-retention rule (App. A). *Budget knobs*: K and the visit budget (B, F) (Sec. 3.4) are fixed to meet a target per-round overhead. *Robustness/load knobs*: the zone radius and cap S_Z bound discovery and super-peer load (Sec. 3.3), replication r (Sec. 3.7), and AP is used with standard settings. Overall, only $\tau^{(t)}$ and P require adaptation (handled by the above procedures); the rest are deployment budgets rather than data-dependent hyperparameters. Unless otherwise stated, we fix per-model tuples [lr, batch, local epochs] for comparability: CNN [0.01, 64, 1], ResNet-18 [0.05, 64, 1], ResNet-18 [0.01, 64, 1], Albert-base [2×10^{-5}, 16, 1]. The optimal P are 12.3%, 9.6%, 11.2%, and 12.8% of M for FMNIST, Tiny ImageNet, Google Speech, and ALBERT-base-v2, respectively. All configurations use $K = 0.10\,N$ and $S_Z = 0.05\,N$, rounded to the nearest integer; $r = 3$; $(B, F) = (8, 8)$, and $\psi_i = \psi = 2$ for all clients. FL rounds are 100.

Settings. We run on the **SLICES** testbed [61], which exposes heterogeneous, WAN-connected machines and thus realistic P2P conditions. We study increasing *resource heterogeneity* via four machine classes (highest→lowest capacity): *GPU 1080* (6 vCPU, 64 GB RAM + 16 GB VRAM, 512 GB disk), *Xlarge* (8 vCPU, 16 GB RAM, 40 GB disk), *Large* (8 vCPU, 10 GB RAM, 20 GB disk), *Medium* (4 vCPU, 4 GB RAM, 10 GB disk). Each workload is pinned to the *smallest* class that meets its memory/compute footprint: (i) *CNN/FMNIST* → Medium; (ii) *Google Speech* → Large; (iii) *Tiny ImageNet* → Xlarge; (iv) *20Newsgroup* → Xlarge/GPU 1080 (mixed, reflecting limited GPU

Table 2. Test accuracy (%) on FMNIST, Tiny ImageNet, Google Speech, and 20Newsgroup under non-IID partitions (Dirichlet $\alpha \in \{0.3, 0.1\}$; Pathological $\{20\%, 30\%\}$). Mean±std over 3 runs;

Dataset	Partition	FedAvg	FedProx	Per-FedAvg	Ditto	pFedGraph	DPFL	DFedPGP	SemanticDFL
FMNIST	Dir (0.3)	66.4 ± 1.4	69.9 ± 1.7	73.4 ± 1.5	76.1 ± 0.8	75.4 ± 0.9	76.8 ± 1.5	77.9 ± 0.9	**79.6 ± 0.3**
	Dir (0.1)	62.0 ± 1.4	66.0 ± 1.8	69.7 ± 1.3	70.8 ± 1.9	70.4 ± 1.5	71.0 ± 1.1	71.3 ± 1.7	**73.1 ± 0.4**
	Path (30%)	57.0 ± 1.1	61.5 ± 1.7	65.0 ± 1.5	67.2 ± 1.5	66.7 ± 1.0	68.4 ± 1.6	69.8 ± 1.8	**71.5 ± 0.9**
	Path (20%)	53.5 ± 1.2	58.0 ± 1.5	61.5 ± 1.4	63.8 ± 1.0	63.2 ± 1.7	65.7 ± 1.6	67.4 ± 1.1	**68.4 ± 0.5**
Tiny ImageNet	Dir (0.3)	38.3 ± 2.0	42.3 ± 2.3	46.3 ± 1.6	47.6 ± 1.2	48.2 ± 1.7	47.0 ± 1.4	48.6 ± 1.9	**48.8 ± 0.3**
	Dir (0.1)	36.0 ± 2.1	40.0 ± 2.8	44.0 ± 1.4	46.1 ± 1.5	46.8 ± 1.4	45.2 ± 1.7	47.1 ± 1.2	**47.3 ± 0.9**
	Path (30%)	36.7 ± 2.8	41.0 ± 2.3	44.7 ± 1.6	46.6 ± 1.5	47.2 ± 1.7	45.8 ± 1.4	47.8 ± 1.6	**48.0 ± 0.1**
	Path (20%)	36.2 ± 2.2	40.5 ± 1.6	44.2 ± 1.3	45.3 ± 1.2	45.6 ± 1.5	44.9 ± 1.5	45.8 ± 1.8	**45.9 ± 0.4**
Google Speech	Dir (0.3)	79.3 ± 1.8	83.5 ± 1.6	87.3 ± 1.1	88.5 ± 1.7	88.2 ± 1.3	88.8 ± 1.5	89.0 ± 1.2	**89.1 ± 0.9**
	Dir (0.1)	76.6 ± 1.5	81.0 ± 1.8	84.6 ± 1.5	85.7 ± 1.3	85.3 ± 1.6	86.2 ± 1.8	86.6 ± 1.1	**86.8 ± 1.1**
	Path (30%)	72.4 ± 1.4	76.4 ± 1.5	80.4 ± 1.4	81.9 ± 1.2	81.6 ± 1.8	82.5 ± 1.3	83.0 ± 1.3	**83.3 ± 0.9**
	Path (20%)	68.2 ± 1.6	72.0 ± 1.7	76.2 ± 1.6	77.9 ± 1.0	77.3 ± 1.8	78.9 ± 1.5	79.5 ± 1.2	**79.8 ± 0.5**
20Newsgroup	Dir (0.3)	50.8 ± 1.9	55.0 ± 1.3	58.8 ± 1.7	63.5 ± 1.4	61.6 ± 1.8	61.0 ± 1.6	60.1 ± 1.2	**64.2 ± 0.9**
	Dir (0.1)	49.4 ± 1.5	53.6 ± 1.7	57.4 ± 1.3	62.7 ± 1.8	60.3 ± 1.4	59.1 ± 1.1	58.2 ± 1.0	**63.3 ± 0.6**
	Path (30%)	48.0 ± 1.5	52.0 ± 1.9	56.0 ± 1.3	63.2 ± 1.7	59.7 ± 1.4	58.3 ± 1.7	57.2 ± 1.3	**63.9 ± 1.2**
	Path (20%)	48.6 ± 1.6	52.6 ± 1.1	56.6 ± 1.9	64.1 ± 1.7	59.8 ± 1.2	58.4 ± 1.8	57.5 ± 1.4	**64.5 ± 0.7**

availability). As budgets shrink (higher heterogeneity), progressively smaller networks fit entirely in memory. We scale the network size N within the limits of our EU SLICES deployments: for the *Medium* setting, we scale up to N=400, for the *XLarge* and *GPU* settings we scale up to N=200.

Baselines. We compare *SemanticDFL* against (i) centralized FL (*FedAvg* [51], *FedProx* [38]), (ii) strong *server-based* PFL (*Ditto* [37], *Per-FedAvg* [14]), (iii) *graph-based* PFL (*pFedGraph* [71]), and (iv) the closest *PDFL* state of the art (*DPFL* [32], *DFedPGP* [46]).

5.2 Accuracy Evaluation Unde non-IID partitions

5.2.1 Comparing Accuracy with baselines. Table 2 reports test accuracy (%) for four datasets across Dir(0.3), Dir(0.1), Patho(20%), Patho(30%). In this experiment, FMNIST and Google Speech use $N = 200$ clients, while Tiny ImageNet and 20Newsgroup use $N = 100$. The results indicate that *SemanticDFL* is best in all settings, with gains that generally widen as heterogeneity increases. Centralized personalization (Per-FedAvg, Ditto) improves over FedAvg/FedProx, and graph-aware methods (pFedGraph, DFedPGP); however, their push/topology-driven mixing remains content-agnostic. In contrast, our semantics-guided *pull* concentrates aggregation on distribution-aligned neighbors, reducing heterogeneity-induced mixing bias. For *Patho*(20%), *SemanticDFL* improves accuracy over FedAvg by 27.9%, 26.8%, 17.0%, and 32.7%, and over DFedPGP by 1.5%, 0.2%, 0.4%, and 12.2%, for FMNIST, Tiny ImageNet, Google Speech, and 20Newsgroup, respectively. See Appendix E.1, E.2, and E.3 for Ablations on signature size, zone size, and personalization level, respectively.

5.2.2 Time-to-Target Accuracy. We measure the *earliest FL round* $r \leq 100$ at which test accuracy first reaches a fixed target set strictly below the FedAvg terminal accuracy for each dataset/partition so that all methods achieve it. As reported in Table 3, *SemanticDFL* consistently attains the target with the fewest rounds. Median speedups are 2.5× over FedAvg and 1.3× over the strongest decentralized baselines (pFedGraph/DFedPGP). These are consistent with semantics-guided *pull*, reducing gradient mismatch and accelerating early-round alignment under non-IID.

Table 3. Rounds to first reach *Target accuracy*.

	FMNIST		Tiny ImageNet		Google Speech		20Newsgroup	
Method	Dir(0.1)	Path(20%)	Dir(0.1)	Path(20%)	Dir(0.1)	Path(20%)	Dir(0.1)	Path(20%)
Target: Recall@K	@60	@50	@34	@33	@74	@66	@46	@45
FedAvg	24	28	60	58	22	30	36	34
FedProx	22	25	54	52	20	28	32	30
Per-FedAvg	16	19	46	44	16	22	26	25
Ditto	14	17	42	40	14	20	22	21
pFedGraph	12	14	38	36	12	17	19	18
DPFL	18	22	48	46	13	18	27	26
DFedPGP	13	16	40	38	12	17	21	20
SemanticDFL	**9**	**11**	**33**	**31**	**9**	**12**	**14**	**13**

Table 4. Scalability at fixed K=10%: each entry is Recall@K / $\text{Acc}_\star$ / $\text{Acc}_{SemanticDFL}$ under $Dir(0.1)$ and $Patho(20\%)$ for varying N. Here $\text{Acc}_\star$ uses oracle Top-K, while $\text{Acc}_{SemanticDFL}$ uses SON-retrieved Top-K.

Dataset	N=50		N=100		N=200		N=400	
	$Dir(0.1)$	$Patho(20\%)$	$Dir(0.1)$	$Patho(20\%)$	$Dir(0.1)$	$Patho(20\%)$	$Dir(0.1)$	$Patho(20\%)$
FMNIST	0.95/80.2/80.0	0.91/76.4/76.2	0.96/79.1/78.9	0.92/75.1/74.9	0.96/77.8/77.5	0.93/73.6/73.2	0.97/76.5/76.1	0.92/72.0/71.5
Tiny ImageNet	0.93/47.3/47.0	0.88/42.0/41.6	0.94/46.0/45.7	0.89/40.8/40.3	0.94/44.4/44.0	0.90/39.0/38.4	0.95/42.8/42.2	0.90/37.2/36.4
Google Speech	0.96/89.4/89.2	0.93/86.2/86.0	0.97/88.6/88.4	0.94/85.1/84.9	0.97/87.3/87.0	0.94/83.6/83.2	0.98/85.9/85.5	0.95/82.0/81.4
20Newsgroup	0.97/65.8/65.6	0.94/62.3/62.0	0.98/64.6/64.4	0.95/61.2/60.9	0.98/63.0/62.6	0.95/59.6/59.1	0.99/61.5/61.0	0.96/58.1/57.4

5.3 Search Efficiency Evaluation

To benchmark *SemanticDFL* against a centralized topology, we define an *ideal centralized oracle* that, at each round t, has visibility of all client states and uses cosine similarity on full gradient vectors to compute the Top-K models for every client by scoring all clients: $S_{i,\star}^{(t)} = \text{TopK}_K(\{s_{ij}^{(t)}\}_{j\in[N]\setminus\{i\}})$. Running the same PDFL update (Sec. 2, same ψ and weighting) with $S_{i,\star}^{(t)}$ yields the oracle accuracy $\text{Acc}_\star$. The only difference is exact Top-K selection vs. SON-retrieved approximate Top-K. We quantify retrieval quality by $\text{Recall@}K = \mathbb{E}_{i,t}\left[|\widehat{S}_i^{(t)} \cap S_{i,\star}^{(t)}|/K\right]$.

5.3.1 Centralized-oracle comparison: Recall@K and accuracy gap. Table 4 reports, for each dataset/split and network size, the triple $\text{Recall@}K/\text{Acc}_\star/\text{Acc}_{SemanticDFL}$ at fixed K=10%. Across all settings, *SemanticDFL* attains high Recall@K (typically ≥ 0.93 for Dir(0.1) and ≥ 0.88 for Patho(20%)) and, consequently, its accuracy closely tracks the centralized oracle: the gap $\Delta\text{Acc} = \text{Acc}_\star - \text{Acc}_{SemanticDFL}$ remains sub-pp at small/medium networks and increases only mildly at N=400. For example, on 20Newsgroup under Dir(0.1), Recall improves 0.97→0.99 as N increases while $\text{Acc}_{SemanticDFL}$ stays within ≈0.3–0.8pp of $\text{Acc}_\star$. Overall, the consistently low oracle gap indicates that SON-based retrieval recovers nearly the same collaborators as centralized selection, yielding near-oracle accuracy while decentralized operating under bounded fanout.

5.3.2 Recall vs. K at Fixed Network Size. We fix network size at N=200 and vary $K \in \{5\%, 10\%, 15\%, 20\%\}$ of N. Table 5 shows Recall@K for all datasets under $Dir(0.1)$ and $Patho(20\%)$. Across the board, recall lies between **88**% and **100**%, and is consistently higher for the easier $Dir(0.1)$ splits, reflecting tighter latent clusters; e.g., FMNIST/$Dir(0.1)$ reaches 0.96–0.99 as K grows from 10% to 20%, while the harder Tiny ImageNet/$Patho(20\%)$ stays in the 0.90–0.94 band. This explains the faster time-to-target results reported earlier: since most of the true top-K neighbors are found early, the aggregation mixes semantically aligned updates rather than gossiping with mismatched peers.

Table 5. Recall@K vs. K at fixed N=200.

	$Dir(0.1)$, N=200				$Patho(20\%)$, N=200			
Dataset/K	5%	10%	15%	20%	5%	10%	15%	20%
FMNIST	0.94	0.96	0.97	0.99	0.90	0.93	0.95	0.97
Tiny ImageNet	0.92	0.94	0.96	0.98	0.88	0.90	0.92	0.94
Google Speech	0.95	0.97	0.98	0.99	0.91	0.94	0.96	0.98
20Newsgroup	0.96	0.98	0.99	1.00	0.93	0.95	0.97	0.99

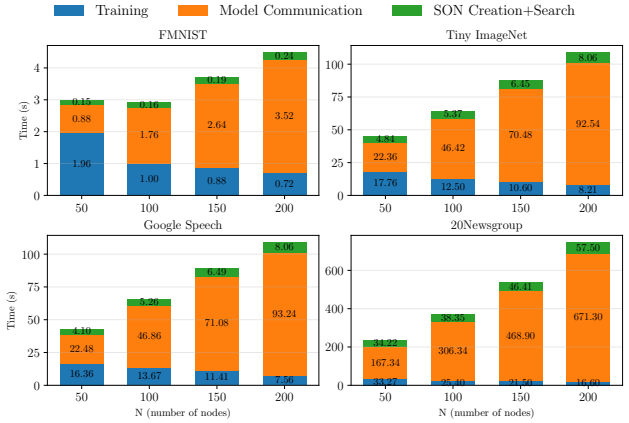


Fig. 5. Per-node FL-round time breakdown at BW=200 Mbps and K=0.1N.

Table 6. Per-node time for a single FL round. Each entry shows FL-round time / SON search time (search share % of the FL round) in seconds. Pulls per node are $\lfloor K\% \cdot N \rfloor$, $K \in \{10, 15, 20, 25\}$%.

FMNIST (CNN)					**Tiny ImageNet (ResNet-18)**				
N	K=10%	K=15%	K=20%	K=25%	N	K=10%	K=15%	K=20%	K=25%
100	2.92/0.16 (5.56%)	3.82/0.18 (4.68%)	4.71/0.19 (3.97%)	5.59/0.19 (3.45%)	50	44.21/3.03 (6.86%)	56.22/3.34 (5.94%)	68.08/3.49 (5.13%)	79.90/3.61 (4.52%)
200	4.26/0.24 (5.71%)	6.05/0.27 (4.43%)	7.82/0.28 (3.58%)	9.59/0.29 (3.02%)	100	59.09/3.37 (5.71%)	82.84/3.71 (4.48%)	106.42/3.88 (3.64%)	129.97/4.01 (3.09%)
300	5.90/0.26 (4.40%)	8.57/0.29 (3.34%)	11.22/0.30 (2.66%)	13.87/0.31 (2.23%)	150	80.21/4.05 (5.04%)	115.73/4.45 (3.85%)	151.05/4.65 (3.08%)	186.33/4.81 (2.58%)
400	7.59/0.28 (3.64%)	11.13/0.30 (2.73%)	14.67/0.32 (2.16%)	18.20/0.33 (1.81%)	200	103.15/5.06 (4.90%)	150.48/5.56 (3.70%)	197.56/5.82 (2.94%)	244.59/6.02 (2.46%)
Google Speech (ResNet-18)					**20Newsgroup (Albert-base)**				
N	K=10%	K=15%	K=20%	K=25%	N	K=10%	K=15%	K=20%	K=25%
100	59.06/3.93 (6.66%)	82.86/4.33 (5.22%)	106.47/4.52 (4.25%)	130.04/4.68 (3.60%)	50	229.52/28.89 (12.59%)	316.00/31.78 (10.06%)	401.04/33.22 (8.28%)	485.79/34.38 (7.08%)
200	103.71/5.90 (5.69%)	151.13/6.49 (4.29%)	198.25/6.78 (3.42%)	245.31/7.02 (2.86%)	100	383.28/32.10 (8.38%)	553.68/35.31 (6.38%)	722.47/36.92 (5.11%)	890.94/38.20 (4.29%)
300	149.54/6.29 (4.21%)	220.40/6.92 (3.14%)	290.95/7.24 (2.49%)	361.44/7.49 (2.07%)	150	551.28/38.52 (6.99%)	805.91/42.37 (5.26%)	1058.62/44.30 (4.19%)	1310.94/45.84 (3.50%)
400	196.07/6.69 (3.41%)	290.39/7.36 (2.53%)	384.37/7.69 (2.00%)	478.28/7.96 (1.66%)	200	725.31/48.15 (6.64%)	1064.50/52.97 (4.98%)	1401.28/55.37 (3.95%)	1737.58/57.30 (3.30%)

5.4 SON Creation and Search Overhead Analysis

5.4.1 FL round Time decomposition for Different N. We first quantify its *end-to-end* overhead relative to the standard FL data plane (local training + model exchange) under BW=200 Mbps and K=0.1N. Fig. 6 decomposes the per-node round time into Training, Model Communication (downloading the K pulled models *and* uploading the local update to others), and SON Creation+Search. We amortize SON creation over rounds. The results show that the round is dominated by the data plane—especially communication—while SON overhead remains bounded: SON contributes 5.2% (FMNIST), 8.5% (Tiny ImageNet), 8.1% (Google Speech), and 10.3% (20Newsgroup/Albert-base) of the total round time (about 8.0% overall). Moreover, as model/transfer cost increases, the *relative* SON share shrinks because communication grows faster than lookup. Overall, SemanticDFL's control plane adds a small overhead while enabling semantic (Top-K) collaboration, which is repaid in higher accuracy and faster convergence, as shown in Table 2 and Table 3.

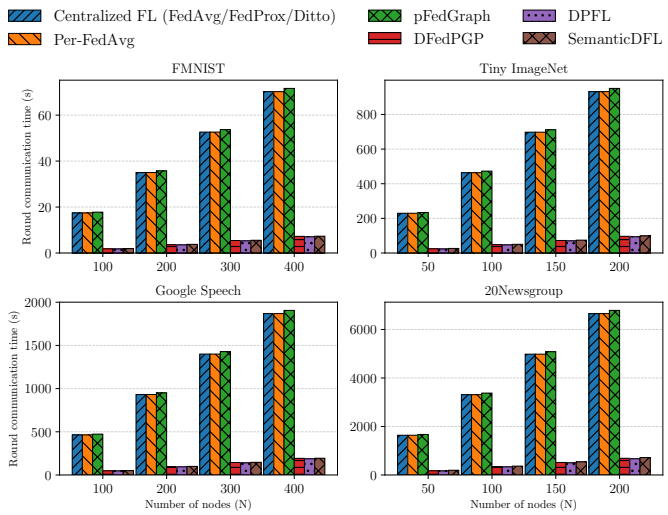


Fig. 6. Per-node round communication time vs. number of nodes (N) under 200 Mb/s bandwidth.

5.4.2 Scaling with collaboration breadth K and network size N. Table 6 quantifies the per-node cost of widening collaboration from K=10% to 25%. Across all, increasing K inflates the total round time approximately proportionally (more models pulled and more incoming pull-serves), while SON search grows only mildly. Consequently, the relative search overhead decreases monotonically with K: for FMNIST it drops from 5.56%→3.45% at N=100 and 3.64%→1.81% at N=400; for 20Newsgroup from 12.59%→7.08% (N=50) and 6.64%→3.30% (N=200). Overall, widening semantic collaboration increases data-plane communication, while the control-plane remains a bounded fraction of the round, enabling larger K when the accuracy/convergence gains justify the additional bandwidth. Moreover, FL-round time breakdowns under varying bandwidth are reported in Appendix E.4.

5.4.3 Communication overhead vs. baselines across FL categories (centralized, graph-PFL, and decentralized). Fig. 6 reports the per-node *round communication time* under fixed BW=200 Mb/s as the population N scales. Centralized methods scale worst because each round requires full-model uplink/downlink through a server bottleneck (fan-in/fan-out), so per-node time grows steeply with N and model size. Per-FedAvg remains in the same regime (often slightly higher) since it adds peer exchange on top of a global synchronization path. pFedGraph, while personalized via an inferred collaboration graph, is effectively centralized: clients still pay the dominant server-mediated transfer cost each round, so its scaling tracks centralized FL as N grows. Decentralized baselines are locality-bound: DPFL (one-hop gossip) is lowest as it exchanges only with immediate neighbors, while DFedPGP is higher due to additional coordination but still far below centralized regimes. SemanticDFL is near-decentralized yet non-local: it pays a small premium over one-hop gossip to pull Top-K aligned peers, while avoiding server bottlenecks and dense mixing via bounded SON routing. Thus, it is much faster than centralized FL/pFedGraph and richer than DPFL; more overlay hops would add communication, but remain budget-bounded.

5.5 Robustness Under Client Leave (Dropout)

5.5.1 Availability-Aware robustness to churn for different methods. Real edge/P2P deployments operate under continual churn. We set N=200 and vary the steady-state churn level $\in \{0\%, 10\%, 25\%, 40\%\}$ per round, where churn denotes the fraction of peers that depart/arrive between consecutive rounds. Table 7 reports test accuracy for four datasets under $Dir(0.1)$. Across all settings, accuracy degrades

Table 7. Accuracy (%) under Dir(0.1) with churn ∈ {0, 10, 25, 40}% (fixed N=200).

	FMNIST				Tiny ImageNet				Google Speech				20Newsgroup			
Method/ Churn level	0%	10%	25%	40%	0%	10%	25%	40%	0%	10%	25%	40%	0%	10%	25%	40%
FedAvg	62.0	59.0	54.0	48.0	36.0	33.0	28.0	23.0	76.6	74.6	70.6	64.6	49.4	47.4	42.4	35.4
FedProx	66.0	63.0	59.0	53.0	40.0	37.0	32.0	26.0	81.0	79.0	75.0	69.0	53.6	51.6	46.6	39.6
Ditto	70.8	68.8	64.8	59.8	46.1	44.1	40.1	34.1	85.7	83.7	80.7	75.7	62.7	60.7	56.7	49.7
Per-FedAvg	69.7	67.7	63.7	58.7	44.0	42.0	38.0	32.0	84.6	82.6	78.6	73.6	57.4	55.4	51.4	44.4
pFedGraph	70.4	68.4	65.4	60.4	46.8	44.8	40.8	35.8	85.3	84.3	81.3	76.3	60.3	59.3	55.3	49.3
DPFL	71.0	68.0	64.0	58.0	45.2	42.2	37.2	31.2	86.2	84.2	80.2	74.2	59.1	57.1	53.1	46.1
DFedPGP	71.3	69.3	65.3	59.3	47.1	45.1	41.1	35.1	86.6	84.6	81.6	75.6	58.2	56.2	52.2	45.2
SemanticDFL	**73.1**	**72.1**	**70.1**	**67.1**	**47.3**	**46.3**	**44.3**	**40.3**	**86.8**	**85.8**	**83.8**	**80.8**	**63.3**	**62.3**	**60.3**	**56.3**

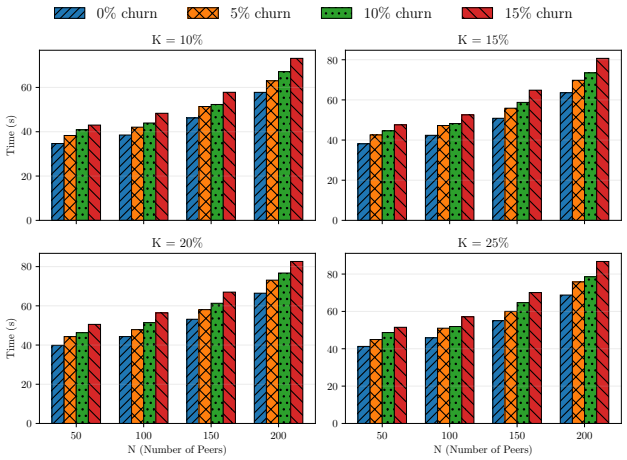


Fig. 7. Control-Plane Overhead for 20Newsgroups under different churn levels.

monotonically with churn, but SemanticDFL exhibits the *slowest* decay. Static or push/gossip overlays lose accuracy faster because neighborhood information becomes partially stale under churn. Under $Dir(0.1)$, SemanticDFL drops by only $\approx 6-7$ points from 0% to 40% churn (e.g., FMNIST $73.1 \rightarrow 67.1$, Tiny ImageNet $47.3 \rightarrow 40.3$), whereas centralized baselines such as FedAvg/FedProx deteriorate by $\approx 12-14$ points and decentralized graph/gossip baselines by $\approx 9-14$ points.

5.5.2 Control-Plane Overhead under Churn for Varying N, K. Fig. 7 reports the per-node *control-plane* time (SON creation+search) under steady-state churn (0–15% of peers join/leave between consecutive rounds), where churn events trigger SON maintenance/refresh; as in prior overhead results, creation is *amortized* over FL rounds, so the bars represent a per-round equivalent cost. Across all (N, K), the overhead increases smoothly with churn but remains modest: SON *search* stays approximately constant (fixed routing/visit budgets and signature comparisons), while the small uplift comes primarily from the additional amortized creation work needed to update zone/replica state as membership changes.

6 Conclusion

We presented *SemanticDFL*, a personalized decentralized federated learning framework that replaces graph-oblivious, push-based mixing with a similarity-aware, pull-based semantic overlay. By exposing compact model signatures, routing bounded-fanout similarity queries through replica-backed super-peers, and pulling only the most relevant models for aggregation, *SemanticDFL* concentrates

communication where it matters while preserving full decentralization and avoiding any-to-any reachability. Experiments across vision, speech, and text benchmarks under challenging non-IID regimes show consistent accuracy gains, faster convergence than decentralized and centralized baselines, low overlay overhead, high neighbor recall, and robustness under churn—establishing semantic overlays with similarity-aware pull as a practical, scalable path to high-quality personalization in real-world PDFL. For future work, we will integrate lightweight structured pruning and quantization tailored to *SemanticDFL*'s signatures and pull pipeline to reduce communication time and make deployments more suitable for low-resource edge devices.

Acknowledgement

Javad Dogani and Nikolaos Laoutaris were supported by the European Union's HORIZON project GenAI4ED (101178648), funded by the Ministry of Economic Affairs and Digital Transformation and the European Union-NextGenerationEU/PRTR.

References

[1] 2020. Case C-311/18: *Data Protection Commissioner v Facebook Ireland Limited and Maximillian Schrems* ("Schrems II"). Judgment of the Court (Grand Chamber), Court of Justice of the European Union. https://curia.europa.eu/juris/liste.jsf?num=C-311/18 ECLI:EU:C:2020:559.

[2] Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Singh, and Sunav Choudhary. 2019. Federated Learning with Personalization Layers. *arXiv preprint arXiv:1912.00818* (2019). arXiv:1912.00818 [cs.LG]

[3] Enrique T. Maté Beltrán, Miguel Q. Pérez, Pedro M. S. Sánchez, Sergio L. Bernal, Gianluca Bovet, Miguel G. Pérez, Germán M. Pérez, and Antonio H. Celdrán. 2023. Decentralized Federated Learning: Fundamentals, State-of-the-Art, Frameworks, Trends, and Challenges. arXiv:2211.08413.

[4] Michael Blot, David Picard, Matthieu Cord, and Nicolas Thome. 2015. Gossip training with deep neural networks. In *Proceedings of the European Symposium on Artificial Neural Networks (ESANN)*. 573–578.

[5] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. 2019. Practical secure aggregation for privacy-preserving machine learning. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*. 1175–1191.

[6] Suo Chen, Yang Xu, Hongli Xu, Zhenguo Ma, and Zhiyuan Wang. 2024. Enhancing decentralized and personalized federated learning with topology construction. *IEEE Transactions on Mobile Computing* (2024).

[7] Yushan Chen, Zhihua Zhang, and Mingyi Hong. 2021. Tighter coupling, faster convergence: Communication-efficient topology for decentralized learning. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*. 1570–1580.

[8] Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. 2021. FedRep: Federated Representation Learning with Separate Local and Global Parameters. In *Proceedings of the 38th International Conference on Machine Learning (ICML) (Proceedings of Machine Learning Research, Vol. 139)*. PMLR, 1958–1968.

[9] Arturo Crespo and Hector Garcia-Molina. 2002. Routing Indices for Peer-to-Peer Systems. In *Proceedings of the 22nd International Conference on Distributed Computing Systems (ICDCS)*. IEEE, Vienna, Austria, 23–32.

[10] Cyberspace Administration of China. 2022. Outbound Data Transfer Security Assessment Measures (). http://www.cac.gov.cn/2022-07/07/c_1658811536396503.htm Promulgated July 7, 2022; effective September 1, 2022.

[11] Rong Dai, Li Shen, Fengxiang He, Xinmei Tian, and Dacheng Tao. 2022. DisPFL: Towards Communication-Efficient Personalized Federated Learning via Decentralized Sparse Training. In *International Conference on Machine Learning (ICML), Proceedings of Machine Learning Research*, Vol. 162. 4587–4604.

[12] Martijn De Vos, Sadegh Farhadkhani, Rachid Guerraoui, Anne-Marie Kermarrec, Rafael Pires, and Rishi Sharma. 2023. Epidemic learning: Boosting decentralized learning with randomized communication. *Advances in Neural Information Processing Systems* 36 (2023), 36132–36164.

[13] Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. 2020. Adaptive Personalized Federated Learning. *arXiv preprint arXiv:2003.13461* (2020).

[14] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. 2020. Personalized Federated Learning: A Meta-Learning Approach. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 33.

[15] Mingyu Fan et al. 2025. PFedDST: Personalized Federated Learning with Decentralized Peer Selection under Resource Constraints. arXiv:2502.07750.

[16] Brendan J Frey and Delbert Dueck. 2007. Clustering by passing messages between data points. *Science* 315, 5814 (2007), 972–976.

[17] Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. 2020. An efficient framework for clustered federated learning. *Advances in neural information processing systems* 33 (2020), 19586–19597.
[18] GSMA Intelligence. 2024. *The State of Mobile Internet Connectivity Report 2024.* Technical Report. GSMA. https://www.gsma.com/r/wp-content/uploads/2024/10/The-State-of-Mobile-Internet-Connectivity-Report-2024.pdf
[19] Xuming Han, Qiaohong Zhang, Zaobo He, and Zhipeng Cai. 2023. Confidence-Based Similarity-Aware Personalized Federated Learning for Autonomous IoT. *IEEE Internet of Things Journal* (2023).
[20] Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. 2020. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604* (2020).
[21] Abolfazl Hashemi, Anish Acharya, Rudrajit Das, Haris Vikalo, Sujay Sanghavi, and Inderjit Dhillon. 2022. On the Benefits of Multiple Gossip Steps in Communication-Constrained Decentralized Federated Learning. *IEEE Transactions on Parallel and Distributed Systems* 33, 11 (2022), 2727–2739.
[22] Chaoyang He, Songze Li, Jinhyun So, Xiaoqian Zhang, Hongyi Wang, Xiaoyang Wang, Praneeth Vepakomma, Abhishek Singh, Hang Qiu, Xinghua Zhu, Jianzong Wang, Li Shen, Peilin Zhao, Yan Kang, Yang Liu, Ramesh Raskar, Qiang Yang, Murali Annavaram, and Salman Avestimehr. 2020. FedML: A Research Library and Benchmark for Federated Machine Learning. In *Proceedings of Machine Learning and Systems (MLSys)*, Vol. 2. 1–12.
[23] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. 2019. Measuring the Effects of Non-Identical Data Distribution for Federated Learning. *arXiv preprint arXiv:1909.06335* (2019). NeurIPS FL Workshop.
[24] Chenghao Hu, Jingyan Jiang, and Zhi Wang. 2019. Decentralized Federated Learning: A Segmented Gossip Approach. arXiv:1908.07782.
[25] Yifan Hua, Kevin Miller, Andrea L. Bertozzi, Chen Qian, and Bao Wang. 2021. Efficient and Reliable Overlay Networks for Decentralized Federated Learning. *arXiv preprint arXiv:2112.15486* (2021).
[26] Yutao Huang, Lingyang Chu, Zirui Zhou, Lanjun Wang, Jiangchuan Liu, Jian Pei, and Yong Zhang. 2020. Personalized Cross-Silo Federated Learning on Non-IID Data. *arXiv preprint arXiv:2007.03797* (2020). FedAMP.
[27] Wonyong Jeong, Jaehee Yoon, and Eunho Yang. 2022. Factorized-FL: Personalized Federated Learning with Parameter Factorization and Similarity Matching. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 35.
[28] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. 2019. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977* (2019).
[29] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. 2020. Scaffold: Stochastic controlled averaging for federated learning. *Proceedings of the 37th International Conference on Machine Learning (ICML)* (2020), 5132–5143.
[30] David Kempe, Alin Dobra, and Johannes Gehrke. 2005. Gossip-based computation of aggregate information. *IEEE Transactions on Information Theory* 51, 7 (2005), 2641–2652.
[31] Ari Keränen, Christer Holmberg, et al. 2018. Interactive Connectivity Establishment (ICE). RFC 8445.
[32] Salma Kharrat, Marco Canini, and Samuel Horváth. 2025. DPFL: Decentralized Personalized Federated Learning. In *Proceedings of The 28th International Conference on Artificial Intelligence and Statistics (Proceedings of Machine Learning Research, Vol. 258)*, Yingzhen Li, Stephan Mandt, Shipra Agrawal, and Emtiyaz Khan (Eds.). PMLR, 5086–5094.
[33] Ken Lang. 1995. 20 Newsgroups Dataset. http://qwone.com/~jason/20Newsgroups/. Originally collected by Ken Lang and widely used for text classification benchmarks.
[34] Ya Le and Xuan Yang. [n. d.]. Tiny ImageNet Visual Recognition Challenge. https://tinyimagenet.cs231n.stanford.edu/. Accessed: 2025-10-03.
[35] Royson Lee, Minyoung Kim, Da Li, Xinchi Qiu, Timothy Hospedales, Ferenc Huszár, and Nicholas Lane. 2023. Fedl2p: Federated learning to personalize. *Advances in Neural Information Processing Systems* 36 (2023), 14818–14836.
[36] Minghao Li, Dmitrii Avdiukhin, Rana Shahout, Nikita Ivkin, Vladimir Braverman, and Minlan Yu. [n. d.]. FIELDING: Clustered Federated Learning with Data Drift. In *The 29th International Conference on Artificial Intelligence and Statistics*.
[37] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. 2021. Ditto: Fair and robust federated learning through personalization. In *International conference on machine learning*. PMLR, 6357–6368.
[38] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2020. Federated Optimization in Heterogeneous Networks. In *Proceedings of Machine Learning and Systems (MLSys)*, Vol. 2.
[39] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. 2020. On the Convergence of FedAvg on Non-IID Data. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*. Addis Ababa, Ethiopia.
[40] Zhuangdi Li, Yongxin Li, Heng Huang, and Jing Liu. 2022. Mining Latent Relationships among Clients: Peer-to-peer Federated Learning with Adaptive Neighbor Matching. *arXiv preprint arXiv:2203.12285* (2022).

[41] Zexi Li, Jiaxun Lu, Shuang Luo, Didi Zhu, Yunfeng Shao, Yinchuan Li, Zhimeng Zhang, Yongheng Wang, and Chao Wu. 2022. Towards effective clustered federated learning: A peer-to-peer framework with adaptive neighbor matching. *IEEE Transactions on Big Data* (2022).

[42] Zexi Li, Jiaxun Lu, Shuang Luo, Didi Zhu, Yunfeng Shao, Yinchuan Li, Zhimeng Zhang, Yongheng Wang, and Chao Wu. 2022. Towards Effective Clustered Federated Learning: A Peer-to-Peer Framework with Adaptive Neighbor Matching. arXiv:2203.12285.

[43] Zexi Li, Jiaxun Lu, Shuang Luo, Didi Zhu, Yunfeng Shao, Yinchuan Li, Zhimeng Zhang, Yongheng Wang, and Chao Wu. 2024. Towards Effective Clustered Federated Learning: A Peer-to-Peer Framework with Adaptive Neighbor Matching (PANM). *IEEE Transactions on Big Data* 10, 6 (2024), 2200–2215. Extended version of arXiv:2203.12285.

[44] Xiangru Lian, Wei Zhang, Ce Zhang, and Ji Liu. 2017. Can Decentralized Algorithms Outperform Centralized Algorithms? A Case Study for Decentralized Parallel Stochastic Gradient Descent. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[45] I-Cheng Lin, Osman Yağan, and Carlee Joe-Wong. 2024. FedSPD: A Soft-clustering Approach for Personalized Decentralized Federated Learning. *arXiv preprint arXiv:2410.18862* (2024).

[46] Yuang Liu, Zhiyuan Zhang, Beichen Gao, Zhenzhong Lin, Yu Li, Xiaoqiang Wang, and Li Shen. 2024. Decentralized Directed Collaboration for Personalized Federated Learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 19600–19609.

[47] Zili Lu, Heng Pan, Yueyue Dai, Xueming Si, and Yan Zhang. 2024. Federated learning with non-iid data: A survey. *IEEE Internet of Things Journal* (2024).

[48] Guixun Luo, Naiyue Chen, Jiahuan He, Bingwei Jin, Zhiyuan Zhang, and Yidong Li. 2024. Privacy-preserving clustering federated learning for non-IID data. *Future Generation Computer Systems* 154 (2024), 384–395.

[49] Othmane Marfoq, Giovanni Neglia, Aurélien Bellet, Laetitia Kameni, and Richard Vidal. 2021. Federated Multi-Task Learning under a Mixture of Distributions. In *NeurIPS*.

[50] Petar Maymounkov and David Mazières. 2002. Kademlia: A Peer-to-Peer Information System Based on the XOR Metric. In *Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS '02) (Lecture Notes in Computer Science, Vol. 2429)*, Peter Druschel, Frans Kaashoek, and Antony Rowstron (Eds.). Springer, Cambridge, MA, USA, 53–65.

[51] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS) (Proceedings of Machine Learning Research, Vol. 54)*, Aarti Singh and Jerry Zhu (Eds.). PMLR, 1273–1282.

[52] Mahdi Morafah, Saeed Vahidian, Weijia Wang, and Bill Lin. 2023. Flis: Clustered federated learning via inference similarity for non-iid data distribution. *IEEE Open Journal of the Computer Society* 4 (2023), 109–120.

[53] Noa Onoszko, Gustav Karlsson, Olof Mogren, and Edvin Listo Zec. 2021. Decentralized Federated Learning of Deep Neural Networks on Non-IID Data. In *ICML 2021 Workshop on Federated Learning (FL-ICML)*. Method: PENS (Performance-Based Neighbor Selection).

[54] Daniele Pasquini, Matthieu Raynal, Lucas Cordeiro, and Luigi V. Mancini. 2023. On the Privacy of Decentralized Machine Learning. arXiv:2302.00618.

[55] Simon Perreault, Ikuhei Yamagata, Shin Miyakawa, Akira Nakagawa, and Hiroyuki Ashida. 2013. Common Requirements for Carrier-Grade NATs (CGNs). RFC 6888. https://doi.org/10.17487/RFC6888

[56] Marc Petit-Huguenin, Gonzalo Salgueiro, et al. 2020. Traversal Using Relays around NAT (TURN). RFC 8656.

[57] Raksha Ramakrishna and György Dán. 2022. Inferring class-label distribution in federated learning. In *Proceedings of the 15th ACM Workshop on Artificial Intelligence and Security*. 45–56.

[58] Yichen Ruan and Carlee Joe-Wong. 2022. FedSoft: Soft Clustering for Federated Learning. *IEEE Transactions on Neural Networks and Learning Systems* (2022).

[59] Russian Federation. 2014. Federal Law No. 242-FZ of July 21, 2014: On Amendments to Certain Legislative Acts of the Russian Federation for Clarification of the Procedure of Personal Data Processing in Information and Telecommunication Networks. https://wilmap.stanford.edu/entries/federal-law-no-242-fz Amends Federal Law No. 152-FZ "On Personal Data"; effective September 1, 2015.

[60] Yifan Shi, Li Shen, Kang Wei, Yan Sun, Bo Yuan, Xueqian Wang, and Dacheng Tao. 2023. Improving the Model Consistency of Decentralized Federated Learning. In *International Conference on Machine Learning (ICML), Proceedings of Machine Learning Research*, Vol. 202. 31637–31660.

[61] SLICES Consortium. 2023. SLICES Research Infrastructure. https://www.slices-ri.eu/. https://doi.org/10.23728/slices-ri European large-scale research infrastructure for end-to-end experimentation in networking, computing systems, and services.

[62] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan. 2001. Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications. In *Proceedings of the 2001 ACM SIGCOMM Conference*. ACM, San Diego, CA, USA, 149–160.

[63] Keith Stouffer, Michael Pease, CheeYee Tang, Timothy Zimmerman, Victoria Pillitteri, Suzanne Lightman, Adam Hahn, Stephanie Saravia, Aslam Sherule, and Michael Thompson. 2023. *Guide to Operational Technology (OT) Security*. NIST Special Publication 800-82r3. National Institute of Standards and Technology, Gaithersburg, MD. https://doi.org/10.6028/NIST.SP.800-82r3

[64] Zhenheng Tang, Shaohuai Shi, Bo Li, and Xiaowen Chu. 2022. GossipFL: A decentralized federated learning framework with sparsified and adaptive communication. *IEEE Transactions on Parallel and Distributed Systems* 34, 3 (2022), 909–922.

[65] Paul Vanhaesebrouck, Aurélien Bellet, and Marc Tommasi. 2017. Decentralized Collaborative Learning of Personalized Models over Networks. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, Vol. 54. 509–517.

[66] Lixu Wang, Shichao Xu, Xiao Wang, and Qi Zhu. 2021. Addressing class imbalance in federated learning. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 35. 10165–10173.

[67] Pete Warden. 2018. Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition. *arXiv preprint arXiv:1804.03209* (2018).

[68] Han Xiao, Kashif Rasul, and Roland Vollgraf. 2017. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. *arXiv preprint arXiv:1708.07747* (2017).

[69] Lei Yang, Jiaming Huang, Wanyu Lin, and Jiannong Cao. 2023. Personalized federated learning on non-IID data via group-based meta-learning. *ACM Transactions on Knowledge Discovery from Data* 17, 4 (2023), 1–20.

[70] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2019. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)* 10, 2 (2019), 1–19.

[71] Rui Ye, Zhenyang Ni, Fangzhao Wu, Siheng Chen, and Yanfeng Wang. 2023. Personalized Federated Learning with Inferred Collaboration Graphs. In *Proceedings of the 40th International Conference on Machine Learning (ICML) (Proceedings of Machine Learning Research, Vol. 202)*. PMLR, 39801–39817. https://proceedings.mlr.press/v202/ye23b.html

[72] Valentina Zantedeschi, Aurélien Bellet, and Marc Tommasi. 2020. Fully Decentralized Joint Learning of Personalized Models and Collaboration Graphs. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS) (PMLR, Vol. 108)*. 864–874.

[73] Ligeng Zhu, Zhijian Liu, and Song Han. 2019. Deep Leakage from Gradients. In *Advances in Neural Information Processing Systems*, Vol. 32. 14747–14756.

A Bias–Transient Lower Bound for Oblivious Overlays

We quantify the mis-aggregation penalty of oblivious push/mixing under the clustered non-IID assumptions introduced earlier. Let $\{w_i^\star\}$ denote the local minimizers, and suppose there exist at least two clusters with inter-cluster separation $\delta > 0$ and within-cluster dispersion $\epsilon \geq 0$ (as defined in the main text). We additionally use the standard Laplacian notation: $\lambda_2(L)$ is the algebraic connectivity and $d_{\max}$ the maximum degree.

Let G be the communication graph with adjacency A and degree matrix D. Define the random-walk matrix $\mathbf{P}_{\mathrm{rw}} := D^{-1}A$ and the random-walk normalized Laplacian $L_{\mathrm{rw}} := I - \mathbf{P}_{\mathrm{rw}}$. Let $\sigma_2(\mathbf{P}_{\mathrm{rw}})$ denote the second-largest eigenvalue (in magnitude) of $\mathbf{P}_{\mathrm{rw}}$.

Theorem 2 (Irreducible personalization bias of graph-regularized objectives)**.** *Assume each local loss f_i is L_s-smooth and μ-strongly convex. Consider the graph-regularized objective*

$$g_\gamma(w) \;=\; \sum_i f_i(w_i) \;+\; \gamma \sum_{(i,j)\in E} \|w_i - w_j\|^2,$$

and let $w^\gamma = \{w_i^\gamma\}$ be any minimizer for $\gamma > 0$. Then there exist absolute constants $c_1, c_2 > 0$ (depending only on μ, L_s and the cluster balance parameter, but not on $|V|$, p, or T) and a stepsize threshold $\bar\gamma > 0$ such that, for all $\gamma \in (0, \bar\gamma]$,

$$\frac{1}{|V|}\sum_i \left\|w_i^\gamma - w_i^\star\right\|^2 \;\geq\; c_1 \left(\frac{\gamma\,\lambda_2(L)}{\mu + 4\gamma\, d_{\max}}\right)^2 \delta^2 \;-\; c_2\,\epsilon^2. \tag{3}$$

For fixed $(\mu, \gamma, d_{\max})$, the prefactor $\frac{\gamma\lambda_2(L)}{\mu+4\gamma d_{\max}}$ is nondecreasing in $\lambda_2(L)$ and saturates when $4\gamma d_{\max} \gg \mu$. Across different graphs, however, $\lambda_2(L)$ and $d_{\max}$ may co-vary; thus the bound should be interpreted in terms of the ratio $\lambda_2(L)/(\mu + 4\gamma d_{\max})$ (or along graph families with controlled $d_{\max}$, e.g., d-regular graphs).

Notation for the sketch. Stack parameters as $\mathbf{w} = \mathrm{vec}(\{w_i\}) \in \mathbb{R}^{|V|p}$. Let $\Delta = \mathrm{vec}(\{w_i^\gamma - w_i^\star\})$ and let $\Delta_\perp$ denote the projection onto the graph disagreement subspace (orthogonal to the consensus direction). We write $Q(\mathbf{w}) = \mathbf{w}^\top(L\otimes I)\mathbf{w}$ for the Laplacian quadratic form and use $\|\cdot\|$ for the Euclidean norm.

Proof sketch (safe perturbation route). Define the monotone map $F_\gamma(\mathbf{w}) = \nabla f(\mathbf{w}) + 2\gamma(L\otimes I)\mathbf{w}$, where ∇f stacks the local gradients. By μ-strong convexity and L_s-smoothness, F_γ is μ-strongly monotone and $(L_s + 2\gamma\lambda_{\max}(L))$-Lipschitz. The stationarity conditions read $F_\gamma(\mathbf{w}^\gamma) = 0$ and $F_0(\mathbf{w}^\star) = \nabla f(\mathbf{w}^\star) = 0$.

(i) First-order expansion in γ. By the implicit function theorem, there exists $\bar\gamma > 0$ and a smooth path $\gamma \mapsto \mathbf{w}^\gamma$ with

$$\mathbf{w}^\gamma \;=\; \mathbf{w}^\star \;-\; 2\gamma\left(\nabla^2 f(\mathbf{w}^\star)\right)^{-1}(L\otimes I)\,\mathbf{w}^\star \;+\; O(\gamma^2).$$

Consequently,

$$\|\Delta\| \;\geq\; \frac{2\gamma}{L_s}\left\|(L\otimes I)\mathbf{w}^\star\right\| \;-\; C\gamma^2 \quad \text{for some constant } C = C(\mu, L_s, d_{\max}).$$

(ii) Cluster projection (geometry of $\mathbf{w}^\star$). Decompose $\mathbf{w}^\star = \bar{\mathbf{w}} + \mathbf{w}_\perp^\star$, where $\bar{\mathbf{w}}$ is the consensus component. Standard spectral bounds give

$$\left\|(L\otimes I)\mathbf{w}^\star\right\| \;\geq\; \lambda_2(L)\left\|\mathbf{w}_\perp^\star\right\|.$$

Under the separation/dispersion assumptions, and assuming each of the two largest clusters has fraction at least $\beta > 0$ of nodes, a variance decomposition yields

$$\left\|\mathbf{w}_{\perp}^{\star}\right\| \;\geq\; c(\beta)\,\delta \;-\; C'(\beta)\,\epsilon,$$

for absolute constants $c(\beta), C'(\beta) > 0$ (this is the usual lower bound of the between-cluster component in terms of the inter-cluster gap, up to within-cluster dispersion).

(iii) Conditioning with $d_{\max}$. Using the quadratic form bounds $\Delta^{\top}(L\otimes I)\Delta \leq 2d_{\max}\|\Delta\|^2$ and the Lipschitz/strong-convexity constants to control higher-order terms, we absorb the $O(\gamma^2)$ remainder and obtain, for all $\gamma \in (0, \bar{\gamma}]$,

$$\|\Delta\| \;\geq\; c\,\frac{\gamma\,\lambda_2(L)}{\mu + 4\gamma d_{\max}}\,\Big(\delta - c''\epsilon\Big),$$

for absolute $c, c'' > 0$. Squaring and averaging over i gives (3) with suitable c_1, c_2. □

Remark (Interpretation beyond strong convexity). Theorem 2 is stated for μ-strongly convex f_i to obtain a clean, closed-form lower bound. For nonconvex objectives (e.g., deep networks), the result can be read as a *local* statement: in any neighborhood where each f_i satisfies a PL/strong-convexity-type condition (or admits a quadratic approximation around a stable stationary point), the same perturbation route applies to that local model. We use Theorem 2 primarily as a *mechanistic motivation* that oblivious mixing can induce a topology-dependent, T-independent bias under clustered heterogeneity.

Optimization transient under mix–grad dynamics. Consider the coupled iteration

$$\mathbf{w}^{(t+1)} = (M \otimes I)\,\mathbf{w}^{(t)} - \eta\,\nabla f\big(\mathbf{w}^{(t)}\big), \tag{4}$$

with M symmetric, doubly-stochastic (spectral gap $1 - \lambda(M) > 0$) and stepsize $\eta \leq 1/L_s$. The bias bound above is a property of the *regularized minimizer* w^{γ} and is *algorithm-independent*. For the transient error relative to w^{γ}, a standard consensus/optimization decomposition (consensus vs. disagreement subspaces) yields the contraction rate

$$\lambda(M) \;=\; \max\{\,1 - \mu\eta,\; \lambda(M)\,\} \;<\; 1,$$

so that, for a constant C depending only on (μ, L_s),

$$\frac{1}{|V|}\sum_i \left\|w_i^{(T)} - w_i^{\star}\right\|^2 \;\leq\; \underbrace{2\,\frac{1}{|V|}\sum_i \|w_i^{\gamma} - w_i^{\star}\|^2}_{\text{steady bias from (3)}} \;+\; 2C\,\rho^{\,2T}\left\|\mathbf{w}^{(0)} - \mathbf{w}^{\gamma}\right\|^2. \tag{5}$$

Thus larger spectral gap (smaller $\lambda(M)$) accelerates the transient, while Theorem 2 shows the steady bias *increases* with $\lambda_2(L)$ and γ—exhibiting the fundamental trade-off.

Corollary 1 (Effect of expander-like connectivity)**.** *For a d-regular expander, $\lambda_2(L) = \Theta(d)$ and $\lambda_{\max}(L) \leq 2d$. Substituting into (3) (for $\gamma \in (0, \bar{\gamma}]$) gives*

$$\frac{1}{|V|}\sum_i \|w_i^{\gamma} - w_i^{\star}\|^2 \;\geq\; \Omega\!\left(\left(\frac{\gamma\, d}{\mu + 4\gamma d}\right)^2 \delta^2\right) \;-\; O(\epsilon^2),$$

i.e., increasing connectivity amplifies *the irreducible mis-aggregation under clustered non-IID (and saturates as $\gamma d \to \infty$), even as the transient in (5) contracts faster.*

Table 8. Essential notation used in *SemanticDFL*.

Symbol	Meaning	Symbol	Meaning
$G=(V,E)$, $N=\|V\|$	P2P graph; #peers	S_Z	Zone size cap
R	TTL hops for discovery	H	SON depth (#levels)
M	Model dimension (#params)	$w_i^{(t)}$	Model of client i at round t
$s_{i,r}^{(t)}$	Coord. importance (EMA)	β	Smoothing factor
$I_i^{(t)}$	Top-P index set	P	Signature size (nonzeros)
$u_i^{(t)}=w_i^{[P]}(t)$	Signature (raw mask)	$\mathrm{sim}_u(x,y)$	Cosine similarity on signatures
$\tau_{\min}$	Target $\cos(w_i, w_i^{[P]})$	$P_i^{\min}$	Min P meeting $\tau_{\min}$
q	Quantile to choose P	$P_{\max}$	Cap on P
$\mathcal{C}=\{C_j\}$	Zone clusters	e_j	Exemplar of C_j
$\bar{u}_j$	Prototype of C_j	R_j	Replica set of C_j
r	Replication factor	CD_j	Cluster descriptor
K	# neighbors to pull	B_ℓ	Beam at level ℓ
F_ℓ	Fan-out at level ℓ	$\mathrm{Ch}(v)$	child super-peers at level $\ell-1$
$Q_i=\langle u_i^{(t)}, K, \mathtt{qid}\rangle$	Query of client i	m_ℓ	Children scored at level ℓ
v_ℓ	Visited super-peers @ level ℓ	$\tau^{(t)}$	similarity threshold (round t)
ϕ	Global target acceptance	η_τ	Threshold step size
γ	Laplacian regularization weight	δ	Relax step when $< K$
$\rho_i^{(t)}$	Drift proxy $\cos(u_i^{(t)}, u_i^{(t-1)})$	T_{fresh}	Freshness threshold
$\Delta_{\max}$	Max staleness (rounds)	Δ_t	Soft-barrier timeout
κ	Softmax temperature in $W_{ij}^{(t)}$	λ_{ch}	Per-round churn rate

Discussion. Oblivious mixing (push/graph smoothing) trades faster consensus for a topology-induced, T-independent bias that scales up with both the smoothing weight γ and connectivity (via $\lambda_2(L)$), and is only offset when clusters are indistinguishable ($\delta \to 0$) or smoothing vanishes ($\gamma = 0$). This motivates the similarity-aware pull and structured search in *SemanticDFL* to avoid cross-cluster blending while preserving fast convergence in the transient through targeted communication.

B Algorithmic Specification of *SemanticDFL*

B.1 Binary Search for Selecting the Signature Cardinality P

Goal. Given the current client weights $w_i^{(t)}$ and magnitude-based importance scores $s_{i,r}^{(t)}$ (as defined in the main text), we seek the *smallest* signature cardinality $P_i^{\min}$ such that the directional similarity between the full model and its Top-P *raw masked* version exceeds a target $\theta \in (0,1]$ (we set $\theta \equiv \tau_{\min}$ in the main text):

$$\cos\Big(w_i^{(t)},\, w_i^{[P]}(t)\Big) \;=\; \frac{\big\|w_i^{[P]}(t)\big\|_2}{\big\|w_i^{(t)}\big\|_2} \;\geq\; \theta, \qquad w_i^{[P]}(t) \;=\; \mathrm{mask}\big(w_i^{(t)},\, \underset{r}{\text{Top-P}}\; s_{i,r}^{(t)}\big).$$

Because supports are nested as P grows, the above cosine is *non-decreasing* in P. Hence the feasible set $\{P : \cos(\cdot) \geq \theta\}$ is an interval $[P^\star, M]$, allowing logarithmic-time search.

Precomputation for $O(1)$ similarity tests. Sort indices once per search (or when the guard triggers in the main text), in descending score, with deterministic tie-breaking by index:

$$J = (j_1, \ldots, j_M) \;\leftarrow\; \mathrm{argsort}_r\big(s_{i,r}^{(t)}\big).$$

Let $\ell \leftarrow \|w_i^{(t)}\|_2$; if $\ell = 0$ then return $P_{\min}^i \leftarrow M$ and define prefix sums $c_k = \sum_{m=1}^{k} \left(w_{i,j_m}^{(t)}\right)^2$. Then, for any P,

$$\cos\left(w_i^{(t)},\, w_i^{[P]}(t)\right) \;=\; \frac{\sqrt{c_P}}{\nu},$$

so each similarity evaluation is $O(1)$ after sorting (or compare $c_P \geq \theta^2 \ell^2$ to avoid square roots).

Binary search (minimal P with $\cos \geq \theta$). We search over integers $P \in [\underline{P}, \overline{P}]$ where typically $\underline{P} = 1$ and $\overline{P} = M$. A tolerance $\varepsilon_P \in \mathbb{N}$ (default 1) controls termination.

Algorithm 1 BinarySearchForSignatureCardinality

Require: Weights $w_i^{(t)}$, scores $s_{i,r}^{(t)}$, threshold $\theta \in (0, 1]$, bounds $\underline{P}$=1, $\overline{P}$=M, tolerance ε_P=1
Ensure: Minimal $P_i^{\min}$ such that $\cos\left(w_i^{(t)}, w_i^{[P]}(t)\right) \geq \theta$

$J \leftarrow \text{argsort}_r\left(s_{i,r}^{(t)}\right)$ ▷ descending by score; ties by index
$\nu \leftarrow \|w_i^{(t)}\|_2$; **if** ν=0 **then return** $P_i^{\min} \leftarrow M$
$c_0 \leftarrow 0$; $c_k \leftarrow \sum_{m=1}^{k} \left(w_{i,j_m}^{(t)}\right)^2$ for k=1..M
FEASIBLE(P) $\leftarrow \left(c_P \geq \theta^2 \nu^2\right)$
$\ell \leftarrow \underline{P}$; $h \leftarrow \overline{P}$ ▷ note: FEASIBLE(M) is true
while $h - \ell > \varepsilon_P$ **do**
    $m \leftarrow \lfloor (\ell + h)/2 \rfloor$
    if FEASIBLE(m) **then**
        $h \leftarrow m$ ▷ feasible; try smaller P
    else
        $\ell \leftarrow m$+1 ▷ infeasible; need more coords
    end if
end while
return $P_i^{\min} \leftarrow h$

Correctness sketch. Let $S_P = \{j_1, \ldots, j_P\}$. Since $S_P \subset S_{P+1}$, we have $c_P \leq c_{P+1}$, hence $\sqrt{c_P}/\nu$ is non-decreasing in P. Therefore the feasible set is an interval in P, and standard binary search returns its left endpoint $P^\star$.

Complexity. Per search: sorting costs $O(M \log M)$ (or can be amortized across rounds when orders change slowly); prefix sums are $O(M)$; each feasibility test is $O(1)$. Total is $O(M \log M)$ time and $O(M)$ memory. We recompute the order and prefix sums only every T_{ref} rounds or when the guard $\cos\left(w_i^{(t)}, w_i^{[P]}(t)\right) < \theta - \epsilon$ (main text) fires.

Corner cases and robustness. (i) If θ is set too high, the algorithm returns $P_i^{\min} = M$. (ii) Optionally add a safety margin $\eta > 0$ and accept the smallest P with $c_P \geq (\theta{+}\eta)^2 \nu^2$ to buffer drift between refreshes. (iii) Quantizing P to packet-friendly granularities (e.g., multiples of 32) simplifies implementation with negligible impact on cosine. (iv) Deterministic tie-breaking in Top-P stabilizes $I_i^{(t)}$ and avoids jitter.

From local to global P. Each client outputs $P_i^{\min}$. Super-peers aggregate via a robust rule—e.g., $P := \min\{\text{Quantile}_q(\{P_{\min}^i\}), P_{\max}\}$ with $q \in [0.8, 0.9]$— to set a single network-wide cardinality, ensuring that at least a q-fraction of clients satisfy $\cos \geq \theta$ while keeping signatures comparable during search. If the chosen P exceeds a local budget, a client may *clip* to its device-specific $P_{\max}$; its search recall may degrade but routing remains consistent.

B.2 Distributed Zone Formation

Initiator election (per epoch). At epoch e, each node p computes a verifiable random score $\sigma_p = \mathrm{VRF}_p(\mathrm{seed}(e-1))$; within its R-hop neighborhood the node with the minimum $(\sigma_p, \mathrm{id}_p)$ becomes the initiator (ties by ID). This yields a spaced set of initiators at granularity R.

Algorithm 2 SelectInitiators($\mathcal{P}, R$)

Require: Peer set $\mathcal{P}$, hop radius R, epoch seed seed_{t-1}
Ensure: Initiator set $\mathcal{I}$
for each $p \in \mathcal{P}$ **in parallel do**
  $\sigma_p \leftarrow \mathrm{VRF}_{pk_p}(\mathrm{seed}_{t-1})$
  $m_p \leftarrow \min_{q \in \mathcal{N}_R(p)} (\sigma_q, \mathrm{id}_q)$
  if $(\sigma_p, \mathrm{id}_p) = m_p$ **then**
    mark p as Initiator
  end if
end for
return $\mathcal{I} \leftarrow \{ p \in \mathcal{P} \mid p$ marked as Initiator $\}$

TTL-bounded probing and first-wins assignment. Initiators flood a PROBE with TTL = R. A NOT_ASSIGNED peer that receives its *first* PROBE adopts that initiator and forwards the message (excluding the sender). Peers that are already ASSIGNED send adjacency notifications to both initiators to register zone neighbors. If $|\mathcal{Z}_I| > S_Z$, I hands a disjoint subset to the next-lowest VRF score in-zone, ensuring the cap.

Algorithm 3 ZoneFormationProtocol($\mathcal{V}, R, S_Z$)

Init.
for each $p \in \mathcal{V}$ **do**
  p.state $\leftarrow$ NOT_ASSIGNED
end for
$\mathcal{I} \leftarrow$ SelectInitiators($\mathcal{V}, R$)
Probe.
for each $I \in \mathcal{I}$ **do**
  I.state $\leftarrow$ ASSIGNED; $\mathcal{Z}_I$.members $\leftarrow \{I\}$
  BroadcastTTL(I, PROBE(I), R)
end for
Handlers.
procedure OnReceivePROBE($sender$, $initiator$, ttl)
  if $self$.state = NOT_ASSIGNED **then**
    $self$.state $\leftarrow$ ASSIGNED($initiator$)
    $\mathcal{Z}_{initiator}$.members $\leftarrow \mathcal{Z}_{initiator}$.members $\cup \{self\}$
    Send($initiator$, REGISTER($self$.cap))
    if ttl > 0 **then**
      ForwardTTL(PROBE($initiator$), ttl $-$ 1)
    end if
  else
    Send($initiator$, NEIGHBOR($self$.initiator))
    Send($self$.initiator, NEIGHBOR($initiator$))
  end if
end procedure
Termination and split.
Wait $R \cdot t_a$ time units (diameter timeout within TTL scope)
for each $I \in \mathcal{I}$ **do**
  if $|\mathcal{Z}_I$.members$| > S_Z$ **then**
    SplitZone($\mathcal{Z}_I$) ▷ hand a disjoint subset to next-lowest VRF in-zone
  end if
end for

L1–L3: Initialize all peers as NOT_ASSIGNED; elect initiators via VRF-min within R hops (Alg. 2). *L4–L7*: Each initiator seeds its zone with itself and launches a TTL= R flood; this bounds propagation radius and latency. *Handler L8–L16*: On the *first* probe, the peer atomically adopts that initiator

(first-wins), registers capabilities, and forwards while TTL>0. This ensures disjoint membership and proximity grouping. *Handler L17–L18*: If already assigned, the peer reports an inter-zone adjacency to both initiators, allowing zone-neighbor discovery without extra probes. *L19*: A fixed timeout $R \cdot t_a$ (hop time estimate t_a) safely concludes the wave. *L20–L22*: If size exceeds S_Z, split by delegating a disjoint member subset to the next-lowest VRF in-zone; the cap enforces load balancing and keeps per-zone control cost bounded.

Why it is correct (sketch). (i) *Coverage*: VRF election yields a dispersed set of initiators; TTL= R probing ensures every peer within R hops of at least one initiator is reached in the epoch; epochs rotate seeds, so uncovered peers (if any) are covered in subsequent epochs. (ii) *Disjointness*: The first-wins rule plus idempotent state transition to ASSIGNED prevents duplicate zone membership. (iii) *Bounded size*: The explicit check $|\mathcal{Z}_I| \le S_Z$ with SplitZone ensures the invariant holds each epoch. (iv) *Low latency/proximity*: TTL bounding restricts membership to an R-hop ball around each initiator, aligning zones with network proximity.

Complexity. Control messages per zone are $O(S_Z)$: each peer processes at most one first-wins PROBE and forwards to its neighbors once within TTL; adjacency reports add $O(\partial\mathcal{Z}_I)$ where $\partial\mathcal{Z}_I$ is the zone boundary. Each initiator stores $O(S_Z)$ member metadata; peers keep $O(1)$ state. End-to-end wall time is $O(R \cdot t_a)$.

C Convergence Proofs

This appendix provides the auxiliary steps invoked in Section 4.

C.1 Bias–variance decomposition (Lemma 2)

Under Assumption 2, conditioned on $w_i^{(t)}$ and $S_i^{(t)}$, $\mathbb{E}[g_i^{(t)}] = \frac{1}{K_t}\sum_{j\in S_i^{(t)}} \nabla F_j(w_i^{(t)}) = \mu_{S_i^{(t)}}(w_i^{(t)})$. Independence and equal weighting give $\mathrm{Var}(g_i^{(t)}) = \frac{1}{K_t^2}\sum_{j\in S_i^{(t)}} \mathrm{Var}(g_j) \le \sigma^2/K_t$. Taking expectations over K_t yields σ^2/K_{eff}.

C.2 One-step descent (Lemma 4)

By L_s-smoothness, $F_i(w-\eta g) \le F_i(w) - \eta\langle \nabla F_i(w), g\rangle + \frac{\eta^2 L_s}{2}\|g\|^2$. Take $w = w_i^{(t)}$, $g = g_i^{(t)}$, then take expectations and insert $\mathbb{E}[g_i^{(t)}] = \mu_{S_i^{(t)}}(w_i^{(t)})$, add/subtract $\nabla F_i(w_i^{(t)})$, and complete the square using $2ab \le a^2 + b^2$. With $\eta \le 1/(2L_s)$ the standard inequality yields the stated coefficients.

C.3 Mismatch bound (Lemma 5)

Triangle inequality gives $\|\nabla F_i - \mu_S\| \le \|\nabla F_i - \mu_O\| + \|\mu_O - \mu_S\| + \|\text{stale} - \text{fresh}\|$. Squaring and using $(a+b+c)^2 \le 3(a^2+b^2+c^2)$ together with Lemmas 1 and 3 yields ζ_i^2.

C.4 Proof of Theorem 1

Sum Lemma 4 over t. Telescoping yields

$$\frac{\eta}{2}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla F_i(w_i^{(t)})\|^2 \le F_i(w_i^{(0)}) - F_i^\star + \eta T\,\zeta_i^2 + \frac{\eta^2 L_s T}{2}\cdot\frac{\sigma^2}{K_{\text{eff}}}.$$

Divide by ηT, multiply by 2, and obtain (2).

D Control traffic, creation latency, and memory

D.1 Control-Plane Accounting

At level ℓ, there are $Z_\ell = \Theta(N/S_Z)$ zones. Each node emits $\bar{d}$ headers of size S_H and one signature of size S_P, for $N\bar{d}S_H + NS_P$ bytes. Each zone produces $\overline{C}$ CDs of size S_D, for $Z_\ell \overline{C} S_D$ bytes; super-peer replication scales the descriptor term by r. Summing levels multiplies by a constant (caps fixed). Root-level sync among F representatives adds either $\Theta(F^2 S_D)$ (mesh) or $\Theta(FS_D)$ (tree). This establishes Proposition 1.

D.2 Creation Latency

Within a level, bounded-degree overlays concentrate diameter at $D_Z = O(\log S_Z)$; floods and probes finish in $O(D_Z)$ rounds. Clustering under caps (S_Z, P) has bounded per-zone work and completes in $O(1)$ rounds (constant iterations). Since levels proceed sequentially but zones in parallel, the per-level wall-clock is $O(\log S_Z)$. With $H = \lceil \log_{S_Z}(N) \rceil$ levels and an $O(\log F)$ representative sync, Proposition 2 follows.

D.3 Memory Bound

A super-peer at level ℓ stores up to S_Z child entries consisting of one signature and one descriptor; replication factor r multiplies the descriptor term. Summing across H levels gives $M_{\max} \le HS_Z(S_P + rS_D)$, proving Proposition 3.

E Extended Experimental Results

E.1 Ablation on Signature Size P

Top-P signatures balance fidelity and cost: too small P under-represents salient directions; too large P admits noise/drift and bloats control traffic. We therefore ablate P per model/dataset around the rounded optimum reported by our base runs. The optimal P are 12.3%, 9.6%, 11.2%, and 12.8% of M for FMNIST, Tiny ImageNet, Google Speech, and ALBERT-base-v2, respectively. For each case, we evaluate P with some integer values near the optimal P.

The results in Table 9 show that accuracy is maximized at a narrow, dataset-specific signature budget P (FMNIST 12.3%M, Tiny ImageNet 9.6%M, Google Speech 11.2%M, 20Newsgroup 12.8%M), with a smooth but non-negligible sensitivity around the optimum. Rounding to the nearest integer used in the main runs (FMNIST 15, Tiny ImageNet 11, Google Speech 13, 20Newsgroup 14) remains near-optimal, typically within $\approx 0.2-0.3$ pp of the best setting. Deviating by ± 1 from the rounded operating point reduces accuracy by a median ~ 0.6 pp (range $0.4-1.3$ pp) across all datasets/partitions, while a ± 2 deviation causes a larger median drop of ~ 1.1 pp (range $0.8-1.8$ pp). Sensitivity is consistently higher under more heterogeneous splits (especially $Dir(0.1)$ and Patho 30%) than under $Dir(0.5)$, reflecting reduced overlap and higher update drift. Practically, when bandwidth is constrained, selecting the left-shoulder configuration $(P-1)$ often incurs only $\approx 0.4-0.9$ pp loss while shrinking signature payloads proportionally.

E.2 Ablation on Zone Size S_Z

The zone-size cap S_Z controls cluster granularity and the breadth of in-zone candidate discovery before similarity routing. We therefore vary $S_Z \in \{0.04N,\ 0.05N,\ 0.08N,\ 0.10N\}$ around our default $0.05N$ and measure test accuracy across all datasets/partitions while keeping P at the dataset-specific optimum. The results in Table 10 indicate that across different tasks, accuracy shifts relative to $0.05N$ are small and patternless: the median absolute change is $\sim$0.8 pp (IQR $\approx$0.4–1.3 pp), with a maximum of +1.5 pp/−2.1 pp. No setting shows a consistent monotone increase/decrease with zone

Table 9. Ablation on P (accuracy %, mean±std). Best per column within each dataset block in **bold**. The opt. P values are used in all other experiments; the $Dir(0.1)$ opt. entries match the corresponding SemanticDFL results reported in the main table.

Dataset	P (%) of M	Dir (0.1)	Dir (0.5)	Patho (20%)	Patho (30%)
FMNIST	10	71.6±0.9	77.3±0.6	72.8±0.8	74.6±0.9
	11	72.4±0.8	78.4±0.5	74.1±0.7	75.8±0.8
	12.3 (opt)	**73.1±0.7**	**79.2±0.4**	**75.0±0.6**	**76.6±0.7**
	13	72.7±0.8	78.8±0.5	74.6±0.7	76.2±0.7
	14	71.9±0.9	78.0±0.6	73.4±0.8	75.0±0.9
Tiny ImageNet	7	45.6±2.0	50.4±1.8	46.8±1.7	48.2±1.8
	8	46.7±1.9	51.6±1.7	48.1±1.6	49.6±1.7
	9.6 (opt)	**47.3±1.8**	**52.2±1.6**	**48.9±1.5**	**50.3±1.6**
	10	47.0±1.9	51.8±1.7	48.5±1.6	50.0±1.6
	11	46.0±2.0	50.8±1.8	47.2±1.7	48.7±1.8
Google Speech	9	84.8±1.1	87.8±1.0	82.6±1.1	84.0±1.1
	10	86.0±1.0	88.7±0.9	83.8±1.0	85.1±1.0
	11.2 (opt)	**86.8±0.9**	**89.4±0.9**	**84.6±0.9**	**85.9±0.9**
	12	86.4±0.9	89.0±0.9	84.2±0.9	85.5±0.9
	13	85.2±1.0	88.1±1.0	83.1±1.0	84.4±1.0
20Newsgroup	10	61.2±1.4	66.1±1.3	59.0±1.3	61.0±1.2
	11	62.6±1.3	67.4±1.2	60.4±1.2	62.3±1.1
	12.8 (opt)	**63.3±1.2**	**68.0±1.2**	**61.0±1.1**	**62.9±1.0**
	13	62.9±1.2	67.6±1.2	60.6±1.1	62.5±1.0
	14	61.8±1.3	66.6±1.3	59.4±1.2	61.4±1.1

Table 10. Zone-size ablation for **SemanticDFL**: accuracy (%, mean±std over 3 runs) vs. S_Z. The default $S_Z = 0.05N$.

Dataset	Partition	S_Z=0.04N	S_Z=0.05N **(base)**	S_Z=0.08N	S_Z=0.10N
FMNIST	Dir 0.3	79.0 ± 0.4	79.6 ± **0.3**	80.1 ± 0.4	78.7 ± 0.5
	Dir 0.1	72.5 ± 0.5	73.1 ± **0.4**	73.6 ± 0.5	72.0 ± 0.6
	Path 30%	71.8 ± 1.0	71.5 ± **0.9**	71.0 ± 0.9	70.1 ± 1.1
	Path 20%	67.7 ± 0.6	68.4 ± **0.5**	68.9 ± 0.6	66.9 ± 0.7
Tiny ImageNet	Dir 0.3	49.4 ± 0.5	48.8 ± **0.3**	48.2 ± 0.4	47.5 ± 0.6
	Dir 0.1	46.5 ± 1.0	47.3 ± **0.9**	48.7 ± 0.8	46.0 ± 1.0
	Path 30%	47.6 ± 0.3	48.0 ± **0.1**	48.9 ± 0.2	46.9 ± 0.4
	Path 20%	44.8 ± 0.6	45.9 ± **0.4**	47.0 ± 0.5	45.2 ± 0.6
Google Speech	Dir 0.3	88.4 ± 1.0	89.1 ± **0.9**	90.2 ± 0.8	87.2 ± 1.1
	Dir 0.1	87.1 ± 0.9	86.8 ± **0.9**	86.2 ± 0.9	85.0 ± 1.0
	Path 30%	82.0 ± 1.0	83.3 ± **0.9**	84.8 ± 0.8	81.2 ± 1.1
	Path 20%	80.5 ± 0.4	79.8 ± **0.1**	79.6 ± 0.3	78.6 ± 0.5
20Newsgroup	Dir 0.3	63.8 ± 1.0	64.2 ± **0.9**	65.4 ± 0.8	62.7 ± 1.1
	Dir 0.1	62.9 ± 0.7	63.3 ± **0.6**	64.6 ± 0.6	62.0 ± 0.7
	Path 30%	64.1 ± 0.6	63.9 ± **0.5**	63.2 ± 0.5	62.5 ± 0.7
	Path 20%	64.9 ± 0.4	64.5 ± **0.3**	65.6 ± 0.3	63.4 ± 0.5

size, confirming that *SemanticDFL*'s signature-based neighbor discovery reliably surfaces the top-K most similar peers for a broad range of S_Z. For reproducibility, we keep $S_Z = 0.05N$ as the default.

Table 11. Effect of personalization weight ψ under Dir(α=0.3). We report mean test accuracy (%).

ψ	$\alpha_i^{\text{mix}} = \frac{\eta\psi}{1+\eta\psi}$	FMNIST	Tiny ImaegeNet	Google Speech	20Newsgroup
FedAvg	–	66.4	38.3	79.3	50.8
0	0.00	71.0	41.5	83.5	55.5
0.1	0.09	74.2	44.0	86.0	58.5
0.5	0.33	77.8	47.0	88.0	61.8
1	0.50	79.0	48.0	88.7	63.5
2	0.67	79.6	48.8	89.1	64.2
5	0.83	79.1	48.2	88.5	63.4

E.3 Ablation on ψ (Personalization vs. globalization)

Personalized decentralized FL (PDFL) relaxes the global-consensus constraint of classical FL/DFL by maintaining a distinct model w_i per client while still enabling collaboration through a controlled coupling term. In our formulation (Sec. 2), client i performs a local update and then interpolates its intermediate model $w_{e,i}^{(t+1)}$ with a neighbor anchor $m_i^{(t)} = \sum_{j\in\mathcal{N}_i^K} W_{ij}^{(t)} w_j^{(t)}$, where $\mathcal{N}_i^K$ is the Top-K similarity set and $W_{ij}^{(t)}$ are normalized similarity weights. Here, η is the proximal/mixing step size (distinct from the local SGD step size η_t). The scalar $\psi_i \geq 0$ (or equivalently $\alpha_i \in [0, 1)$) explicitly controls the *globalization–personalization* tradeoff: $\psi_i \to 0$ recovers purely local learning, while a larger ψ_i increases attraction to the collaborative anchor and thus moves toward more global/cluster-level behavior. This neighbor-regularized objective is a standard PDFL instantiation (graph-/multi-task-style personalization). Other personalization mechanisms exist (e.g., shared/private layer factorization, meta-learning, mixture models), but they are orthogonal to our focus on scalable decentralized collaborator discovery and similarity-aware pull.

A ψ-sweep is essential because the optimal coupling strength is heterogeneity-dependent: too small ψ under-utilizes collaboration (high variance, limited transfer), whereas too large ψ over-regularizes toward the neighborhood anchor (reduced client specialization). Reporting performance across ψ therefore makes the personalization–globalization spectrum explicit and identifies the operating regime where collaboration improves accuracy without collapsing solutions. Table 11 summarizes this sensitivity under Dir(0.3).

Table 11 shows that moderate coupling ($\psi\approx2$) consistently dominates both global FedAvg and near-local training. Relative to FedAvg, the best operating point improves accuracy by +13.2pp (FMNIST: 66.4→79.6), +10.5pp (Tiny ImageNet: 38.3→48.8), +9.8pp (Google Speech: 79.3→89.1), and +13.4pp (20Newsgroup: 50.8→64.2). Relative to purely local training (ψ=0), the same point adds +8.6pp (71.0→79.6), +7.3pp (41.5→48.8), +5.6pp (83.5→89.1), and +8.7pp (55.5→64.2), indicating that controlled neighbor attraction yields substantial positive transfer. Over-coupling slightly degrades performance (e.g., ψ=5 vs. ψ=2: −0.5pp FMNIST, −0.6pp Tiny ImageNet, −0.6pp Google Speech, −0.8pp 20Newsgroup), consistent with reduced client specialization under excessively strong regularization.

E.4 Overhead sensitivity to bandwidth (communication-dominated scaling).

Real deployments span heterogeneous and time-varying links, so we stress-test how SemanticDFL's per-round overhead behaves as bandwidth degrades. Table 12 reports a per-node FL-round breakdown into training (bandwidth-independent) and *Comm/Search* for $BW \in \{50, 100, 150, 200\}$ Mbps; to avoid idealized $1/BW$ scaling, we inject independent 0–5% jitter into communication and search.

Table 12. Per-node FL-round time breakdown with bandwidth variation. Training time scales sublinearly with N (60–70% reduction when doubling N) and is independent of bandwidth. Each entry reports *Comm/Search* (Search % of total round time) in seconds for $BW \in \{50, 100, 150, 200\}$ Mbps. Communication and search include an independent 0–5% jitter per cell.

Model/Dataset	N	Train (s)	BW=50	BW=100	BW=150	BW=200
FMNIST (CNN)	100	1.00	3.58/0.32 (6.48%)	1.78/0.16 (5.31%)	1.20/0.11 (4.79%)	0.89/0.08 (3.92%)
	200	0.66	7.33/0.50 (5.89%)	3.58/0.24 (5.03%)	2.37/0.16 (4.96%)	1.77/0.12 (4.74%)
	300	0.50	11.03/0.53 (4.58%)	5.41/0.27 (4.57%)	3.55/0.17 (4.33%)	2.75/0.13 (4.27%)
	400	0.42	14.75/0.56 (3.65%)	7.39/0.29 (3.69%)	4.78/0.19 (3.61%)	3.60/0.14 (3.67%)
TinyImageNet (ResNet-18)	50	17.76	48.36/6.32 (8.53%)	23.61/3.13 (6.93%)	16.21/2.06 (5.66%)	11.72/1.59 (5.09%)
	100	11.18	97.40/6.82 (6.09%)	47.10/3.45 (5.82%)	31.64/2.35 (5.49%)	23.73/1.71 (5.01%)
	150	8.10	147.13/8.15 (5.07%)	73.61/4.07 (4.87%)	48.59/2.72 (4.76%)	36.63/2.10 (4.71%)
	200	7.27	196.39/10.13 (4.84%)	95.82/5.19 (4.96%)	64.03/3.44 (4.82%)	47.50/2.65 (4.88%)
Google Speech (ResNet-18)	100	8.30	95.51/8.10 (7.28%)	49.10/4.09 (6.69%)	31.73/2.67 (6.28%)	24.27/1.98 (5.75%)
	200	5.56	193.57/12.29 (5.96%)	94.65/5.96 (5.79%)	63.77/3.94 (5.58%)	47.43/3.02 (5.61%)
	300	4.10	290.57/12.79 (4.20%)	143.99/6.31 (4.14%)	97.25/4.25 (4.09%)	72.72/3.17 (4.05%)
	400	3.56	381.48/13.41 (3.41%)	193.89/7.01 (3.48%)	127.02/4.62 (3.48%)	94.26/3.35 (3.39%)
20Newsgroup (ALBERT-base)	50	33.44	350.28/60.29 (13.52%)	174.10/29.88 (12.55%)	115.93/20.10 (11.83%)	84.99/14.57 (10.93%)
	100	20.73	699.95/65.19 (8.45%)	347.18/33.40 (8.48%)	233.93/22.37 (8.25%)	172.25/16.35 (8.02%)
	150	16.00	1046.64/78.72 (6.98%)	508.52/40.00 (7.19%)	346.40/25.99 (6.80%)	253.82/19.42 (6.86%)
	200	14.30	1399.55/99.84 (6.64%)	699.56/48.52 (6.43%)	467.87/33.38 (6.57%)	342.48/24.69 (6.59%)

As expected, communication dominates the bandwidth sensitivity: e.g., for ResNet-18/Tiny ImageNet at N=200, communication drops from 196.39 s (50 Mbps) to 47.50 s (200 Mbps), and for ALBERT/20Newsgroup at N=200 from 1399.55 s to 342.48 s. In contrast, SON search is far less bandwidth-sensitive and remains a small fraction of the round: for CNN/FMNIST, the search share stays within ≈3.6–6.5% across all N and bandwidths; for ResNet-18 (Tiny ImageNet/Google Speech) it stays around ≈3.4–8.5%; and for ALBERT it decreases with N from ≈13.5% (50 Mbps, N=50) to ≈6.4–6.6% (N=200), since the round becomes increasingly communication-dominated. Training time is independent of bandwidth and decreases sublinearly with N (about 60–70% reduction when doubling N), so the main effect of poorer links is to inflate the data plane rather than the control plane. Overall, the takeaway is that bandwidth heterogeneity primarily changes the cost of model transfer, while SemanticDFL's SON search remains a bounded, predictable overhead across bandwidth regimes, making Top-K semantic collaboration viable even under constrained links.

E.5 Evaluation on Local and Common Global Test Sets

To clarify the evaluation protocol, we report two complementary test views. For each client, we first measure accuracy on a held-out *local test set* generated with the same client-wise partition proportions as its training data, while keeping the samples disjoint from training. This is the primary metric for personalized federated learning, since the goal is to optimize each client model for its own target distribution. We additionally evaluate every learned client model on a *common global test set* sampled from the overall population distribution and shared across all clients. This second metric serves as a diagnostic of cross-distribution generalization.

Figure 8 shows a consistent pattern on FMNIST: local accuracy increases steadily over rounds, whereas global-test accuracy improves more modestly and remains substantially lower. This behavior is expected for a personalized method such as SemanticDFL. Rather than driving all clients toward a single global consensus model, SemanticDFL encourages each client to aggregate only with its Top-K most similar peers, thereby learning a model that is better matched to its local or cluster-level distribution.

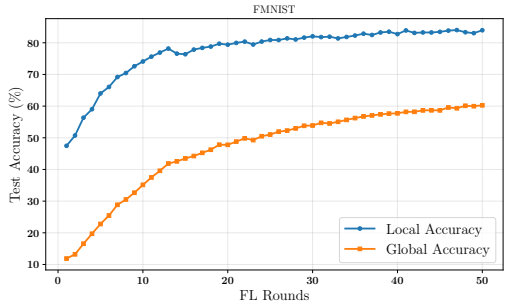


Fig. 8. Average test accuracy over FL rounds on CNN/FMNIST under local and common global evaluation with $N = 100$ clients and $K = 10$.

The growing gap between local and global accuracies should therefore be interpreted as evidence of increasing personalization. Similarity-aware aggregation improves client-specific fit by reducing mixing with statistically dissimilar peers, but this specialization does not necessarily maximize performance on a pooled global distribution. This is an inherent personalization–generalization trade-off, not a contradiction of the method's objective. The common global test results thus clarify the operating point of SemanticDFL: stronger adaptation to heterogeneous local data, with less implicit globalization than consensus-based alternatives.

E.6 Comparison with Data-Similarity-Based Peer Selection

We additionally compare our peer-selection strategy based on compact model similarity against a data-similarity baseline that selects peers using class-distribution similarity. Figure 9 shows that class-distribution similarity achieves slightly higher test accuracy on FMNIST, which is expected since direct access to class-composition statistics provides a strong signal of client relatedness under static label-skew settings. However, the margin over compact model similarity remains small.

This comparison should be interpreted together with privacy and adaptivity considerations. Prior published work has explicitly noted that the composition of a client's training data can itself be sensitive information in federated learning, and that label-distribution leakage may be particularly problematic in applications such as fraud detection, claim prediction, default prediction, churn prediction, spam detection, anomaly detection, intrusion detection, and conversion prediction [57]. This makes class-distribution-based peer selection less attractive from a privacy perspective, since it relies on exposing or inferring data-side statistics. By contrast, compact model similarity operates on lightweight model-side information already available during training, avoiding explicit exchange of local class-distribution statistics. Related work on class imbalance in federated learning has similarly explored indirect proxies based on gradient information rather than direct access to local label composition [66].

Recent work further suggests that class-distribution representations are inherently limited under drift. FIELDING [36] observes that label-distribution vectors mainly capture label shift, but do not capture changes in the input–output relationship, i.e., concept drift. By contrast, model-side or loss-based representations, such as gradients or gradient directions, can better reflect such changes, and FIELDING further reports that gradient-based clustering improves as training progresses and the model becomes more stable [36]. In decentralized settings, this distinction is particularly important, since a class-distribution-based graph must be refreshed by re-estimating and re-sharing updated statistics, whereas compact model similarity can be recomputed directly from the current

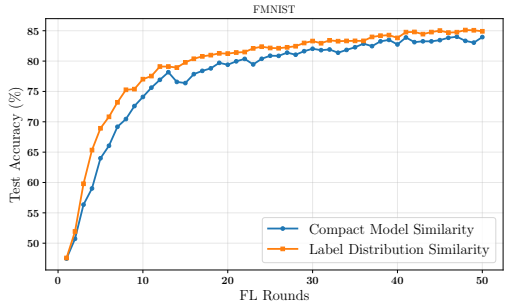


Fig. 9. Comparison of peer selection using compact model similarity and class-distribution similarity on CNN/FMNIST with $N = 100$ clients and $K = 10$.

model state already available during training. Overall, class-distribution similarity serves as a useful reference baseline, whereas compact model similarity offers a more practical privacy–adaptivity trade-off for the dynamic decentralized setting targeted by SemanticDFL.