Supplementary Material: Operationalizing the European Health Data Space: A Governance Framework for Privacy-Preserving Cross-Border Health Analytics

Fabio Liberti
Department of Computer Science
Universitas Mercatorum, Rome, Italy
fabio.liberti@unimercatorum.it
ORCID: 0000-0003-3019-5411

***Abstract*—This document provides comprehensive supplementary material for the EHDS governance paper, including formal governance algorithm specifications, detailed FHIR/OMOP interoperability components, extended privacy-enhancing technology analysis, Federated Learning algorithm catalogue, infrastructure specifications, and clinical validation details. The open-source reference implementation (∼40K lines, 159 modules) is available at https://github.com/FabioLiberti/FL-EHDS-FLICS2026.**

I. Governance Algorithm Specifications

This section provides formal algorithmic descriptions of all EHDS governance components implemented in the framework.

A. EHDS-Compliant FL Training Procedure

Algorithm S1 presents the core training procedure with governance checkpoints integrated at every step.

Algorithm S1: EHDS-Compliant FL Training
Input: Hospitals $\mathcal{H} = \{h_1, \dots, h_K\}$, permit P, rounds T
Output: Global model $\theta^{(T)}$

Server executes:
  Initialize $\theta^{(0)}$
  for round $t = 1$ to T **do**
    // Governance check (Layer 1)
    if not ValidatePermit(P, t) **then abort**
    $\mathcal{H}_t \leftarrow$ SelectParticipants($\mathcal{H}$)
    for each hospital $h \in \mathcal{H}_t$ **in parallel do**
      $\Delta_h^{(t)}, n_h \leftarrow$ LocalTrain(h, $\theta^{(t-1)}$)
    // Privacy-preserving aggregation (Layer 2)
    $\theta^{(t)} \leftarrow \theta^{(t-1)} + \frac{1}{\sum n_h} \sum n_h \cdot \Delta_h^{(t)}$
    LogCompliance(t, $\mathcal{H}_t$)
LocalTrain(h, θ):
  $\mathcal{D}_h \leftarrow$ FilterOptedOut($\mathcal{D}_h$, Registry) *// Art. 71*
  $\theta_h \leftarrow \theta$; train E epochs on $\mathcal{D}_h$
  $\Delta_h \leftarrow$ ClipGradient($\theta_h - \theta$, C) *// DP bound*
  return Δ_h, $|\mathcal{D}_h|$

B. Data Permit Lifecycle Management

Algorithm S2 implements the complete permit lifecycle from application through revocation.

Algorithm S2: Permit Lifecycle Manager
Phases:
1. APPLICATION:
  app ← CreateApplication(purpose, categories, MS)
  app.fl_params ← (algorithm, rounds, ε-budget)
  SubmitToHDAB(app)
2. EVALUATION (at HDAB):
  if app.purpose $\notin$ Art53_Purposes **then** DENY
  if app.ε-budget < MinPrivacyThreshold **then** DENY
  permit ← ApproveWithConstraints(app, duration, budget)
3. EXECUTION:
  for each FL round:
    ValidatePermit(permit)
    CheckOptOuts(permit.MS)
    TrackPrivacyBudget(permit.ε)
4. AUDIT:
  GenerateGDPRArt30Report(permit, logs)
  ArchiveForRegulatory(permit, 5_years)

C. Data Permit Validation

Algorithm S3 ensures compliance at every training round.

Algorithm S3: Permit Validation (Art. 53)
Input: Permit P, round t, categories $\mathcal{C}$
Output: Boolean validity
// Temporal validity
if CurrentTime() > P.valid_until **then**
  raise PermitExpiredError
// Purpose alignment (Article 53)
if P.purpose $\notin$ AllowedPurposes **then**
  raise PurposeMismatchError
// Category authorization
for each $c \in \mathcal{C}$: **if** $c \notin P$.categories **then raise** Error
// Privacy budget check
if P.ε_remaining < ε_per_round **then**
  raise PrivacyBudgetExhaustedError
// GDPR Article 30 audit
AuditTrail.log(permit=P, round=t, categories=$\mathcal{C}$)
return True

D. Article 71 Opt-Out Protocol

Algorithm S4 implements citizen opt-out with fine-grained scope support.

Algorithm S4: Opt-Out Filtering (Art. 71)
Input: Dataset $\mathcal{D}_h$, purpose p, categories $\mathcal{C}$
Output: Filtered dataset $\mathcal{D}'_h$

OptOutRecs ← FetchRegistry(MS) // *LRU-cached, TTL config.*
$\mathcal{D}'_h \leftarrow \emptyset$
for each record $r \in \mathcal{D}_h$ **do**
  opted_out ← False
  // *Blanket opt-out check*
  if (r.id, "ALL") ∈ OptOutRecs: opted_out ← True
  // *Purpose-specific*
  if (r.id, p) ∈ OptOutRecs: opted_out ← True
  // *Category-specific*
  for each $c \in \mathcal{C}$:
    if (r.id, c) ∈ OptOutRecs: opted_out ← True
  if not opted_out: $\mathcal{D}'_h \leftarrow \mathcal{D}'_h \cup \{r\}$
AuditLog.record(total=$|\mathcal{D}_h|$, filtered=$|\mathcal{D}'_h|$, purpose=p)
return $\mathcal{D}'_h$

Opt-out granularity: (1) Blanket—all secondary use; (2) Purpose-specific—e.g., commercial use only; (3) Category-specific—e.g., genomics only. Registry caching: configurable TTL, <10ms latency impact, periodic refresh ensures timely propagation.

E. Cross-Border HDAB Consensus Protocol

Algorithm S5 coordinates multi-Member State studies.

Algorithm S5: Multi-HDAB Coordination
Input: Study S, Member States $\mathcal{M}$
Output: Coordination status, unified permit

permits ← {}
for each MS $m \in \mathcal{M}$ **in parallel do**
  P_m ← SubmitPermitRequest(HDAB$_m$, S)
  permits[m] ← AwaitApproval(P_m)
// *Consensus: ALL HDABs must approve*
if $\exists m$: permits[m] = DENIED **then**
  NotifyAll($\mathcal{M}$, "Study denied by" + m)
  return DENIED
// *Harmonize constraints*
P_u.duration ← $\min_m$(permits[m].duration)
$P_u.\varepsilon$ ← $\min_m$(permits[m].ε)
P_u.categories ← $\bigcap_m$(permits[m].categories)
// *Monitor for mid-study revocation*
StartRevocationMonitor($\mathcal{M}$, permits)
return APPROVED, P_u

Graceful degradation: If one HDAB revokes mid-study, the coordinator: (1) pauses FL training; (2) notifies all parties; (3) removes the revoking MS's clients; (4) optionally continues with remaining MS if study objectives can be met; (5) logs the event for audit.

Algorithm S6: Audit Trail Persistence
Input: FL round context

record ← AuditRecord(
  timestamp = ISO8601_UTC(),
  permit_id = current_permit.id,
  purpose = current_permit.purpose,
  participating_MS = list(active_clients.MS),
  data_categories = list(accessed_categories),
  privacy_budget_consumed = ε_this_round,
  privacy_budget_remaining = ε_total − ε_spent,
  records_processed = count_after_optout,
  records_excluded_optout = count_opted_out,
  model_metrics = {accuracy, loss, AUC},
  anomalies = list(detected_anomalies)
)
AuditStore.persist(record) // *Append-only, immutable*
if record.anomalies: AlertRegulator(record)

Algorithm S7: FHIR R4 Data Harmonization
Input: Raw EHR records $\mathcal{R}$, feature spec $\mathcal{F}$
Output: Training tensors (X, y)

// *Stage 1: Format detection*
format ← DetectFormat($\mathcal{R}$)
parser ← GetParser(format) // *HL7v2, CDA, CSV*
records ← parser.parse($\mathcal{R}$)
// *Stage 2: Terminology mapping*
for each $r \in$ records **do**
  r.diagnoses ← MapToICD10(r.diagnoses)
  r.medications ← MapToATC(r.medications)
  r.labs ← MapToLOINC(r.labs)
// *Stage 3: FHIR transformation*
fhir_bundle ← ToFHIR(records)
ValidateFHIR(fhir_bundle) // *Structural + terminology*
// *Stage 4: ML tensor extraction*
X ← ExtractFeatures(fhir_bundle, $\mathcal{F}$)
X ← StandardScaler.fit_transform(X)
y ← ExtractLabels(fhir_bundle)
return (X, y)

F. GDPR Article 30 Audit Trail

II. FHIR R4 Preprocessing Pipeline

A. Data Harmonization

Supported FHIR R4 Resources: Patient, Observation, Condition, MedicationRequest, Procedure, DiagnosticReport.

Coding Systems: SNOMED-CT, LOINC, ICD-10, ATC, UCUM.

EHDS Data Categories (Article 33): Patient Summary, E-Prescription, Laboratory Results, Medical Imaging, Hospital Discharge, Rare Disease.

B. OMOP CDM Integration

OMOP CDM v5.4 provides an alternative harmonization path for observational research networks (EHDEN, OHDSI).

ETL Pipelines: Transform source EHR to OMOP. **Vocabulary Mapping**: Standard concepts (SNOMED, ICD10, LOINC, RxNorm). **Cohort Definitions**: ATLAS-compatible SQL generation. **Feature Extraction**: FeatureExtraction package for ML-ready datasets.

FL Integration: (1) Each hospital transforms local EHR to OMOP; (2) Feature extraction produces identical schema; (3) FL training proceeds on homogeneous feature spaces across institutions.

III. Extended Interoperability Standards

A. IHE Integration Profiles

ATNA (Audit Trail and Node Authentication):

- TLS mutual authentication between FL nodes
- Syslog audit messages for all data access events (RFC 5424)
- Maps directly to GDPR Article 30 record-keeping

BPPC (Basic Patient Privacy Consents):

- Maps Article 71 opt-out to BPPC consent documents
- XDS.b integration for consent document sharing
- Consent enforcement at FL training initiation

XCA (Cross-Community Access):

- Cross-border document query and retrieve
- Initiating/Responding Gateway implementation
- Patient identity correlation across communities

PIX/PDQ (Patient Identifier Cross-referencing / Demographics Query):

- Patient matching across institutional boundaries
- Pseudonymization-aware identity management
- Integration with national eHealth infrastructures

XUA (Cross-Enterprise User Assertion):

- SAML 2.0 assertions for federated authentication
- Role-based access control integration
- HDAB authorization token propagation

B. Cross-Border Data Exchange

Message Formats: EHDS Data Permit Exchange Format (JSON-LD), Federated Query Protocol (SPARQL Federation), Model Update Message Format (Protocol Buffers).

Security Requirements: eIDAS-compliant electronic signatures for permits, TLS 1.3 for all cross-border communication, certificate-based node authentication (EU trust framework).

Metadata Standards: DCAT-AP Health extension for dataset cataloging, W3C PROV-O provenance, EMA data quality indicators.

C. Interoperability Architecture

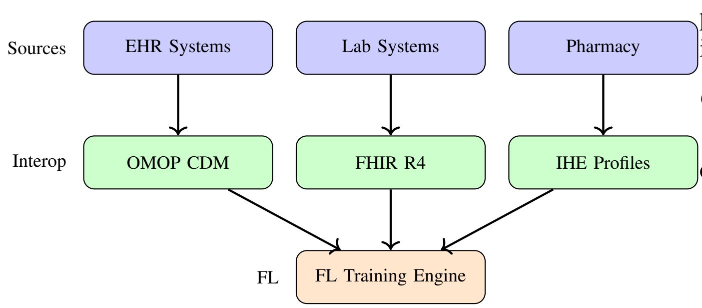

Fig. 1. Interoperability layer integrating heterogeneous data sources for FL training.

IV. Privacy-Enhancing Technology Details

A. Differential Privacy

Algorithm S8: Gaussian DP with Rényi Accounting
Input: Gradient Δ, clip norm C, budget ε, δ
Output: Noisy gradient $\tilde{\Delta}$
$\sigma \leftarrow C \cdot \sqrt{2\ln(1.25/\delta)}/\varepsilon$
for each parameter $w \in \Delta$ **do**
  $\tilde{w} \leftarrow w + \mathcal{N}(0, \sigma^2)$
// Rényi DP moment accounting
PrivacyAccountant.record_step(σ, sampling_rate)
$\varepsilon_{spent} \leftarrow$ PrivacyAccountant.get_epsilon(δ)
if $\varepsilon_{spent} > \varepsilon_{total}$ **then**
  raise BudgetExhaustedError *// Hard stop*
return $\tilde{\Delta}$

Rényi DP (RDP) provides 5–6× tighter composition bounds for the 20+ rounds typical of EHDS studies. For Gaussian mechanisms with noise scale σ, the RDP guarantee at order α is $\rho(\alpha) = \alpha/(2\sigma^2)$.

Governance implications: The ε-budget must be specified in the data permit application, approved by the HDAB, and tracked throughout training. Budget exhaustion triggers automatic training termination—preventing "privacy bankruptcy."

B. Secure Aggregation

Algorithm S9: Pairwise Masking Protocol
Input: Gradients $\{\Delta_1, \ldots, \Delta_K\}$, threshold t
Output: Aggregate Δ_{agg}
// Phase 1: ECDH key exchange
for each pair (j, k):
  $s_{jk} \leftarrow$ ECDH(pk_j, sk_k)
  $r_{jk} \leftarrow$ HKDF-SHA256(s_{jk}, round_id)
// Phase 2: Mask gradients
for each client k:
  $\hat{\Delta}_k \leftarrow \Delta_k + \sum_{j<k} r_{jk} - \sum_{j>k} r_{kj}$
// Phase 3: Aggregate (masks cancel)
$\Delta_{agg} \leftarrow \sum_k \hat{\Delta}_k = \sum_k \Delta_k$
// Dropout: Shamir reconstruction of missing masks
if $|\text{ActiveClients}| < t$: **raise** SecureAggError
return Δ_{agg}

Security guarantee: The aggregation server learns only $\Delta_{agg} = \sum_k \Delta_k$, never individual Δ_k. Combined with DP, this provides defense-in-depth: even if the server is compromised, individual hospital contributions remain protected.

C. Byzantine Resilience

Six defense methods protect model integrity against malicious participants:

- **Krum**: Selects gradient closest to $n-f-2$ nearest neighbors
- **Multi-Krum**: Selects top-m Krum scores, then averages
- **Trimmed Mean**: Removes β-fraction extremes per coordinate
- **Coordinate-wise Median**: Robust estimator per dimension

- **Bulyan**: Two-stage: Krum selection + trimmed mean
- **FLTrust**: Server-guided trust using small trusted dataset

Governance relevance: In cross-border EHDS federations, Byzantine resilience protects against compromised institutions, ensuring that a malicious or malfunctioning participant in one Member State cannot corrupt the global model used by all others.

V. FL Algorithm Catalogue

The framework implements 17 FL algorithms spanning 2017–2025:

TABLE I
Complete FL Algorithm Catalogue

Algorithm	Venue	Category	Key Property
FedAvg	AISTATS'17	Baseline	Weighted avg.
FedProx	MLSys'20	Non-IID	Proximal reg.
SCAFFOLD	ICML'20	Non-IID	Variance red.
FedNova	NeurIPS'20	Non-IID	Normalized avg.
FedDyn	ICLR'21	Non-IID	Dynamic reg.
FedAdam	ICLR'21	Adaptive	Server momentum
FedYogi	ICLR'21	Adaptive	Sparse stability
FedAdagrad	ICLR'21	Adaptive	Grad. accum.
Ditto	ICML'21	Personal.	Dual models
Per-FedAvg	NeurIPS'20	Personal.	MAML-based
FedLC	ICML'22	Label skew	Logit calibration
FedSAM	ICML'22	Generalize	Flat minima
FedDecorr	ICLR'23	Represent.	Decorrelation
FedSpeed	ICLR'23	Efficiency	Fewer rounds
FedExP	ICLR'23	Server-side	POCS step size
FedLESAM	**ICML'24**	**Generalize**	**Global SAM**
HPFL	**ICLR'25**	**Personal.**	**Local classif.**

Bold: 2024–2025 additions. All implemented in the open-source reference.

A. Algorithm Selection for EHDS Governance

Table II maps governance scenarios to recommended algorithms.

TABLE II
Algorithm Selection for EHDS Governance Scenarios

Governance Scenario	Algorithm	Rationale
Homogeneous MS	FedAvg	Simple, auditable
Heterogeneous MS	SCAFFOLD	Handles data skew
Privacy-critical permits	FedAvg + DP	Best-studied bounds
Label-imbalanced data	FedLC	Class calibration
Per-hospital needs	HPFL	Local classifiers
Comm.-constrained	FedSpeed	Fewer rounds
Rapid deployment	FedExP	Server-side only

MS = Member States. Algorithm choice should be specified in the data permit application for HDAB evaluation.

VI. Infrastructure and Deployment

A. Communication Layer

gRPC: Bidirectional streaming, Protocol Buffers serialization (30% bandwidth reduction vs. JSON), HTTP/2 multiplexing. Suitable for data center deployments with low-latency requirements.

WebSocket: Browser-compatible, firewall-friendly (HTTP upgrade), event-driven. Suitable for edge deployments and browser-based participation.

Compression: GZIP, LZ4, ZSTD, Snappy—configurable per deployment.

B. Orchestration

Kubernetes: FL clients/aggregators as pods, HPA for elastic scaling, ConfigMaps for hyperparameters, Secrets for HDAB API keys.

Ray: Actor-based FL with automatic fault tolerance, Ray Tune for federated hyperparameter optimization.

EHDS-Specific: Data residency constraints (gradients processed within national boundaries), permit-aware deployment, regional restrictions.

C. Monitoring and Alerting

Prometheus Metrics: rounds_total, permits_validated, privacy_budget_remaining, active_clients, round_duration, communication_latency.

Governance Alerts: Privacy budget exhaustion warning, permit expiration alerts, opt-out rate spikes, cross-border consensus failures, model divergence detection.

D. User Interfaces

Streamlit Dashboard (15 modules): EHDS governance workflow screens, real-time FL monitoring, permit management, dataset exploration, paper experiment runner.

Terminal UI (11 screens): Algorithm configuration, dataset management, Byzantine settings, hierarchical FL, continual learning, multi-task FL, vertical FL, privacy settings, cross-border coordination.

VII. Experimental Validation Details

A. Datasets

Tabular:

- Heart Disease UCI: 920 patients from 4 international hospitals (Cleveland, Hungarian, Swiss, VA Long Beach). 13 clinical features, binary cardiac diagnosis. Natural non-IID from geographical variation.
- Diabetes 130-US: 101,766 encounters from 130 US hospitals. 22 features, binary 30-day readmission ($\sim$11% positive rate). Partitioned via Dirichlet α=0.5.

Imaging (V2 experiments):

- Chest X-ray: 5,860 pediatric radiographs (NORMAL/PNEUMONIA)
- Brain Tumor MRI: 3,064 T1-weighted CE slices (3-class)
- Skin Cancer: 3,297 dermoscopy images (binary)

B. Governance Workflow Executed

1) Permit application: "scientific research" (Art. 53(1)(b))
2) HDAB evaluation: auto-approval with 20-round budget, ε=10
3) Per-round: permit validation + opt-out filtering + DP
4) FL training: 5 algorithms compared (FedAvg, FedProx, SCAFFOLD, FedNova, Ditto)
5) Audit trail: 100% GDPR Art. 30 field coverage

C. Governance Overhead

- Permit validation: <50ms/round
- Opt-out registry lookup: <10ms/round (LRU cached)
- Cross-border consensus: <200ms for 4-country study
- Audit trail write: <5ms/round
- **Total governance overhead**: <0.3% of training time

D. Reproducibility

```
cd fl-ehds-framework
# Full experiments (7 algo x 5 datasets x 3 seeds)
python -m benchmarks.run_full_experiments
# Quick validation (~1-2h)
python -m benchmarks.run_full_experiments --quick
# Resume after interruption
python -m benchmarks.run_full_experiments --resume
```

E. Supplementary Experimental Figures

The following figures from the benchmark suite provide additional insights:

- Hospital data distribution showing demographic heterogeneity
- Per-client training time variation across hospitals
- Client participation matrix over 50 rounds
- Gradient norm evolution (convergence indicator)
- Communication cost analysis (cumulative)
- Learning rate sensitivity ($\eta \in \{0.01, 0.05, 0.1, 0.2, 0.5\}$)
- Batch size impact ($\{8, 16, 32, 64, 128\}$)
- Per-client accuracy trajectories

All figures are available in the repository under `paper/figures/`.

VIII. Reference Implementation Summary

The open-source codebase (~40,000 lines, 159 Python modules):

TABLE III
CODEBASE MODULE SUMMARY

Module	Files	Key Components
core/	36+	FL algorithms, security, governance
terminal/	15	CLI with 11 specialized screens
dashboard/	15	Streamlit web interface
data/	7	FHIR, OMOP, dataset loaders
models/	3	ResNet-18, MLP, CNN
tests/	6	Governance, DP, config tests
benchmarks/	2+	Paper experiment suite
docs/	6	Architecture, algorithm docs

Key implementation components:

- `core/hdab_api.py`: 1,900+ lines implementing the complete HDAB governance API including DataPermitApplication, DataPermit, OptOutRecord, HDABServiceSimulator, FLEHDSPermitManager, and CrossBorderHDABCoordinator
- `core/fl_algorithms.py`: All 17 FL algorithms with metadata
- `core/secure_aggregation.py`: Pairwise masking with ECDH
- `core/byzantine_resilience.py`: 6 defense methods + attack simulation
- `core/fhir_integration.py`: FHIR R4 resources and coding systems
- `dashboard/ehds_tab.py`: EHDS governance workflow UI

Repository: https://github.com/FabioLiberti/FL-EHDS-FLICS2026

References

[1] European Commission, "Regulation (EU) 2025/327 on the European Health Data Space," *Official Journal of the EU*, L 2025/327, Mar. 2025.
[2] A. Ganna, E. Ingelsson, and D. Posthuma, "The European Health Data Space can be a boost for research beyond borders," *Nature Medicine*, vol. 30, pp. 3053–3056, 2024.
[3] C. Staunton *et al.*, "Ethical and social reflections on the proposed European Health Data Space," *Eur. J. Human Genetics*, vol. 32, no. 5, pp. 498–505, 2024.
[4] P. Quinn, E. Ellyne, and C. Yao, "Will the GDPR restrain health data access bodies under the EHDS?" *Computer Law & Security Review*, vol. 54, art. 105993, 2024.
[5] TEHDAS Joint Action, "Are EU member states ready for the European Health Data Space?" *Eur. J. Public Health*, vol. 34, no. 6, pp. 1102–1108, 2024.
[6] H. Fröhlich *et al.*, "Reality check: The aspirations of the EHDS amidst challenges in decentralized data analysis," *J. Med. Internet Res.*, vol. 27, art. e76491, 2025.
[7] S. van Drumpt *et al.*, "Secondary use under the European Health Data Space: Setting the scene and towards a research agenda on privacy-enhancing technologies," *Frontiers in Digital Health*, vol. 7, art. 1602101, 2025.
[8] R. Hussein *et al.*, "Interoperability framework of the EHDS for secondary use: Interactive EIF-based standards compliance toolkit," *J. Med. Internet Res.*, vol. 27, art. e69813, 2025.
[9] R. Forster *et al.*, "User journeys in cross-European secondary use of health data," *Eur. J. Public Health*, vol. 35, Suppl. 3, pp. iii18–iii24, 2025.
[10] L. Svingel *et al.*, "Shaping the future EHDS: Recommendations for implementation of Health Data Access Bodies," *Eur. J. Public Health*, vol. 35, Suppl. 3, pp. iii32–iii38, 2025.
[11] C. Christiansen *et al.*, "Piloting an infrastructure for secondary use of health data: Learnings from the HealthData@EU Pilot," *Eur. J. Public Health*, vol. 35, Suppl. 3, pp. iii3–iii4, 2025.
[12] M. Shabani and P. Borry, "The European Health Data Space: Challenges and opportunities for health data governance," *Eur. J. Human Genetics*, vol. 32, no. 8, pp. 891–897, 2024.
[13] OHDSI, "The Book of OHDSI: Observational Health Data Sciences and Informatics," 2019.
[14] B. McMahan *et al.*, "Communication-efficient learning of deep networks from decentralized data," in *Proc. AISTATS*, pp. 1273–1282, 2017.
[15] P. Kairouz *et al.*, "Advances and open problems in federated learning," *Found. Trends Mach. Learn.*, vol. 14, no. 1–2, pp. 1–210, 2021.
[16] T. Li *et al.*, "Federated optimization in heterogeneous networks," in *Proc. MLSys*, vol. 2, pp. 429–450, 2020.
[17] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," in *Proc. NeurIPS*, vol. 32, pp. 14774–14784, 2019.

[18] I. Mironov, "Rényi differential privacy," in *Proc. IEEE CSF*, pp. 263–275, 2017.
[19] Z. L. Teo *et al.*, "Federated machine learning in healthcare: A systematic review," *Cell Reports Medicine*, vol. 5, no. 2, art. 101419, 2024.
[20] Z. Qu *et al.*, "FedLESAM: Federated learning with locally estimated sharpness-aware minimization," in *Proc. ICML*, PMLR 235, 2024.
[21] Y. Chen, X. Cao, and L. Sun, "HPFL: Hot-pluggable federated learning with shared backbone and personalized classifiers," in *Proc. ICLR*, 2025.
[22] D. J. Beutel *et al.*, "Flower: A friendly federated learning research framework," *arXiv:2007.14390*, 2023.
[23] NVIDIA, "NVIDIA FLARE: An open-source federated learning platform," *GitHub Repository*, 2023.