FCDC: FEDERATED CONTEXTUAL DISCLOSURE CONTROL FOR HETEROGENEOUS LLM AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Contextual integrity (CI) requires large language model agents to disclose information according to the norms of each interaction. Learning this capability collaboratively is difficult when clients differ both in their disclosure requirements and in the main agents that generate candidate actions. We introduce Federated Contextual Disclosure Control (FCDC), integrating federated CI learning with nested local norm adaptation and shared representation learning. A policy-conditioned contextual encoder provides the common parameter space, while client-specific norm decision modules retain local decision states. A nested objective evaluates shared representations through locally adapted decisions; alternating updates couple local norm adaptation to encoder learning, and only encoder parameters are aggregated. This design supports collaboration across frozen, heterogeneous main agents while keeping interaction records and decision parameters local. Evaluation covers nine clients per fold, two held-out-backbone folds without target adaptation, and action-level and closed-loop privacy–utility outcomes. The tabulated held-out balanced-accuracy gains are approximately 11 percentage points over independent training and 2.6–4.7 points over the strongest federated comparator in each fold. In the MAGPIE-derived comparison with the held-out main agent, FCDC has 5.0 percentage points lower focal private-item leakage and 13.3 points higher constraint satisfaction than an adapted CI supervisor.

1 INTRODUCTION

An assistant arranging a medical appointment may share a detail with a clinic but omit it from a shared calendar. Disclosure can also require verified identity or explicit consent. Contextual integrity (CI) captures these contextual norms (Nissenbaum, 2004; Barth et al., 2006): protection must prevent inappropriate disclosure while preserving permitted sharing. Agents therefore need to judge proposed information flows in the context of their messages and tool calls.

Collaborative learning faces two distinct sources of heterogeneity. *Norm heterogeneity* changes the disclosure conditions, making recipient verification or consent requirements part of the prediction target on otherwise similar tasks. *Main-agent heterogeneity* changes candidate actions and histories through different language-model backbones. Independent CI training leaves other clients' examples unused when records must remain local. Existing CI checks and federated safeguards address parts of this problem (Lan et al., 2025; Wang et al., 2025; Shaaban & Elmahallawy, 2026; Yu et al., 2026). We ask: *can clients learn disclosure control collaboratively while retaining their applicable norms, and reuse that capability when their main agent changes?*

Federated CI learning must preserve the evidence on which different norms base their judgments. Verification can authorize a task-appropriate message under one norm, while another additionally requires scoped consent. Our insight is to assess shared representations through locally adapted disclosure decisions: each client's norm shapes the training signal for the representation being shared. *Federated Contextual Disclosure Control* (FCDC) couples a shared *Contextual Flow Encoder* (CFE) with client-specific *Norm Decision Modules* (NDMs). The CFE encodes original candidate content, task context, and policy evidence; each NDM learns a local decision rule from labeled local examples. Compatible CFE architectures permit aggregation across frozen, heterogeneous main agents (Figure 1). FCDC uses the interacting optimization levels of Nested Learning (Behrouz et al., 2025) to couple local norm adaptation with shared contextual representation learning. Local norms thereby

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

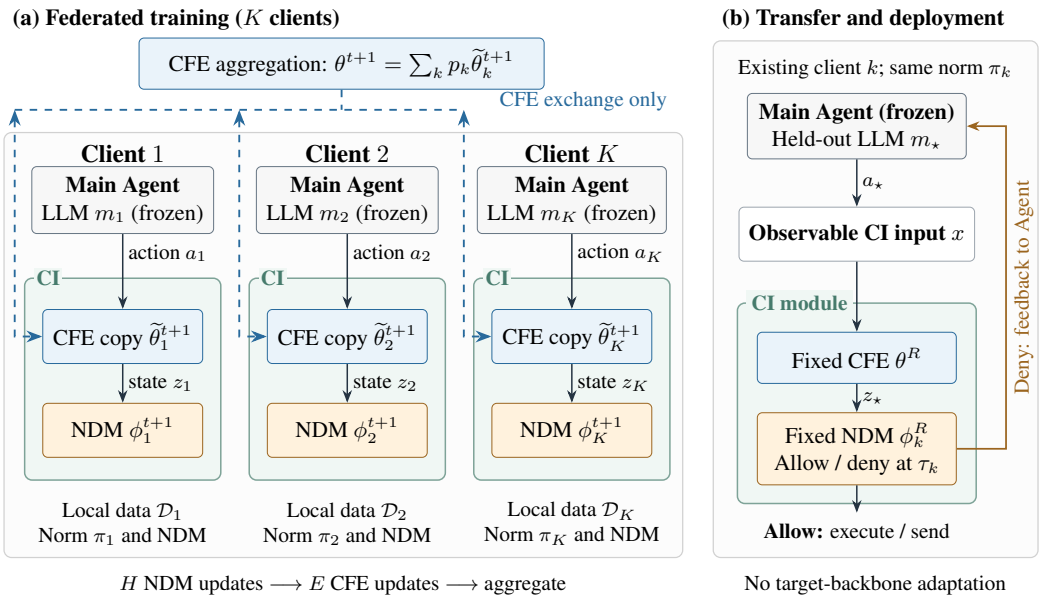

Figure 1: **FCDC framework.** (a) Federated training couples a shared CFE with local NDMs across heterogeneous frozen main agents. NDM adaptation precedes CFE updates and aggregation; records and NDMs remain local. (b) Transfer and deployment retain the existing client's norm, trained CI modules and threshold after main-agent replacement, without target-backbone adaptation.

influence encoder learning while decision parameters and records remain local. Main-agent replacement changes the candidate distribution while preserving this learned normative state.

Our contributions are threefold:

1. **Federated CI under normative and agent heterogeneity.** We formulate collaborative disclosure learning in which client norms and candidate-generating backbones vary independently. FCDC provides a common CI parameter space across heterogeneous frozen main agents while retaining client-specific decisions and keeping interaction records local.
2. **Nested learning for contextual disclosure decisions.** We make local norm adaptation part of shared contextual learning: disclosure risks evaluated through adapted NDMs guide CFE updates before aggregation. Persistent local states and separate update schedules connect client-specific normative supervision to cross-client representation learning.
3. **CI as a reusable capability across main-agent changes.** Across two held-out-backbone folds, FCDC improves balanced accuracy over independent training by approximately 11 percentage points and exceeds personalized federated baselines. Existing clients retain their norms, CI modules and thresholds, without target adaptation. Together with lower leakage and higher task utility than Local-CI, these findings support treating disclosure control as a collaboratively learned capability that persists across main-agent upgrades.

2 Related Work

Contextual integrity and agent privacy. CI evaluates information flows against contextual norms (Nissenbaum, 2004; Barth et al., 2006). ConfAIde tests contextual reasoning (Mireshghallah et al., 2024), PrivacyLens evaluates assistant actions (Shao et al., 2024), and MAGPIE examines privacy and task efficacy in multi-agent collaboration (Juneja et al., 2025). AirGapAgent minimizes trusted-context data before third-party interaction (Bagdasarian et al., 2024); CI supervisors assess information sharing (Ghalebikesabi et al., 2025); CI-RL develops CI reasoning through prompting and reinforcement learning (Lan et al., 2025); and PrivacyChecker provides model-agnostic CI checks (Wang et al., 2025). Together, these approaches establish contextual reasoning and modular protection. FCDC studies how to learn these judgments collaboratively across clients with different norms and main agents while retaining client-specific decisions.

Personalized federation and nested learning. FedAvg aggregates model parameters (McMahan et al., 2017); FedPer retains local personalization layers (Arivazhagan et al., 2019), Ditto regularizes personalized models toward a global solution (Li et al., 2021), and Per-FedAvg learns an initialization for local adaptation (Fallah et al., 2020). FedRep profiles local predictor risks and alternates predictor adaptation with shared representation updates (Collins et al., 2021), offering a structural comparison for FCDC's objective and update schedule. FLAP transfers access-control policies across parties with conflicting policies and data-sharing constraints (Abu Jabal et al., 2023). In CI, the same candidate can require different decisions under recipient-verification or consent requirements. Nested Learning characterizes interacting optimization levels through their states and update schedules (Behrouz et al., 2025). FCDC couples local norm adaptation to shared contextual learning (Equation 4): persistent NDMs retain client-specific disclosure decisions, providing an inductive bias beyond the policy conditioning available to all methods.

Heterogeneous models, federated safeguards, and transfer. FedDF addresses model heterogeneity by distilling heterogeneous predictions using unlabeled data (Lin et al., 2020). SecureGate combines shared sanitized and local revealing adapters with token-controlled routing for federated LLM disclosure (Shaaban & Elmahallawy, 2026); FGLGuard learns a shared graph-based risk detector over private multi-agent traces for runtime intervention (Yu et al., 2026). FCDC exchanges compatible CI encoders outside heterogeneous frozen main agents while retaining local norm adaptation. Transfer tests replacement of an existing client's main agent under an unchanged norm and fixed CI modules and thresholds, without target adaptation (Section 3), rather than adaptation to new clients or norms.

3 Problem Formulation

CI evaluates disclosure under contextual norms governing the sender, recipient, information subject, information attributes, and transmission conditions (Nissenbaum, 2004; Barth et al., 2006).

Clients and heterogeneity. Among K clients, client k holds norm π_k and a local dataset $\mathcal{D}_k = \{(x_i, y_i)\}_{i=1}^{n_k}$ of n_k input–label pairs collected with source backbone m_k. Each client is a participant, such as a user or agent deployment, that retains interaction data and learning state. Each client's main agent proposes actions; its CI module judges their permissibility. Norms and backbones vary independently: norms determine disclosure conditions, whereas backbones affect candidate actions and interaction histories. Main-agent parameters remain frozen throughout CI training.

Observable input and decisions. For task q, observable history h, candidate action a, and trusted context metadata u, the CI input x and predicted permission $\widehat{y}_k(x)$ are

$$x = (q, h, a, \pi_k, u), \qquad \widehat{y}_k(x) = \mathbb{I}[f_k(x) \geq \tau_k]\,. \tag{1}$$

For the input space $\mathcal{X}$, $f_k : \mathcal{X} \to [0, 1]$ scores permission, $\mathbb{I}$ is the indicator function, and $\tau_k \in [0, 1]$ is a threshold fixed using source-domain validation. Labels judge the complete candidate: $y = 1$ for an allowed action and $y = 0$ for a prohibited action under the supplied norm. The common interface serializes these observable fields while preserving the candidate's original content. Gold labels, reference answers, and annotation explanations remain outside x. Appendix A.2 specifies the input fields and their provenance.

Collaboration and transfer. Clients learn CI decisions through parameter exchange while retaining interaction records locally. Local record retention does not provide a formal privacy guarantee for shared parameter updates. To test transfer, we replace m_k with a backbone $m_\star \notin \mathcal{M}_{\mathrm{src}}$, where $\mathcal{M}_{\mathrm{src}}$ contains the fold's training backbones. Each existing client retains its norm, learned CI parameters, and threshold. Target-backbone data are excluded from training, adaptation, and model selection. Source and target evaluations use the same held-out source families; all normative and model variants inherit their source-family split (Appendix A.3).

4 Federated Contextual Disclosure Control

FCDC couples a shared Contextual Flow Encoder (CFE) with client-specific Norm Decision Modules (NDMs), separating cross-client representation learning from local norm adaptation (Figure 1).

4.1 Shared Representation and Client-Specific Decisions

The CFE E_θ, with shared parameters θ, uses ModernBERT-base (Warner et al., 2025). Averaging non-padding token states gives the d-dimensional representation $z = E_\theta(x) \in \mathbb{R}^d$. The local NDM computes

$$f_k(x) = f_{\theta,\phi_k}(x) = \sigma(w_k^\top z + b_k), \qquad \phi_k = (w_k, b_k), \tag{2}$$

where $w_k \in \mathbb{R}^d$ and $b_k \in \mathbb{R}$ are local weights and bias, and σ is sigmoid. Common CFE architecture, tokenizer, and initialization ensure parameter compatibility across heterogeneous main agents. Policy and candidate content enter the CFE, with additional local adaptation through the NDM. The full CFE and NDMs are trainable; main agents remain frozen.

4.2 Nested Learning Objective

Useful shared representations must support different client disclosure criteria. We place local adaptation inside the representation objective so that these criteria determine how the encoder is evaluated. For binary cross-entropy ℓ on client k's training set $\mathcal{D}_k$, the local loss and profiled representation objective are

$$\mathcal{L}_k(\theta, \phi_k) = \frac{1}{n_k} \sum_{(x,y)\in\mathcal{D}_k} \ell(f_{\theta,\phi_k}(x), y), \tag{3}$$

$$\min_\theta J(\theta), \qquad J(\theta) = \sum_{k=1}^{K} p_k \inf_{\phi_k} \mathcal{L}_k(\theta, \phi_k), \quad p_k = \frac{n_k}{\sum_j n_j}. \tag{4}$$

Weights p_k are proportional to training-set size. The inner infimum evaluates how well each client can use the representation after adapting its decision rule; the outer objective combines these locally adapted risks. Policy evidence remains in x, so local NDMs provide an inductive bias beyond policy conditioning, rather than additional policy information.

FCDC's Nested Learning formulation (Behrouz et al., 2025) assigns two learning roles: the local level adapts decisions at a fixed representation, and the shared level learns from losses evaluated through those adapted decisions. This dependency gives local norms a role in determining the encoder's training gradients before federation. FCDC realizes it with finite blockwise steps: adapt the NDM, hold it fixed while updating the CFE, then aggregate CFE parameters. The adapted state persists into the next round; no gradient is taken through its adaptation trajectory.

4.3 Nested Alternating Updates

FCDC alternates local norm adaptation and contextual encoder learning. Over R rounds indexed by $t = 0, \ldots, R-1$, clients take H NDM then E CFE stochastic gradient descent (SGD) steps. Let $\mathrm{SGD}_b^s(u; v, \mathcal{D}_k)$ return block b after s updates $b \leftarrow b - \eta\nabla_b\widehat{\mathcal{L}}_k$ from $b = u$, holding the other block at v. Here η is the learning rate and B the nominal batch size; each step draws $\min(B, n_k)$ records without replacement, with fresh draws across steps, and $\widehat{\mathcal{L}}_k$ is their mean loss. Algorithm 1 retains NDMs and aggregates only locally updated CFE parameters $\widetilde{\theta}_k^{t+1}$:

$$\theta^{t+1} = \sum_{k=1}^{K} p_k \widetilde{\theta}_k^{t+1}. \tag{5}$$

Deployment pairs the final aggregate θ^R with retained ϕ_k^R without further NDM fitting. Both modules and the source-selected threshold remain fixed during backbone replacement (Section 3); Appendix C details gradient paths and implementation.

5 Experimental Evaluation

5.1 Setup and Evaluation Protocol

Data and clients. Controlled disclosure decisions use synthetic scenarios released with CI-RL (Lan et al., 2025). Task families are split before policy or main-agent variants are generated (Appendix B). Each fold crosses three norms with three source backbones to form nine simulated clients.

Algorithm 1 FCDC training

Require: Local training sets $\{\mathcal{D}_k\}_{k=1}^K$; initial θ^0, ϕ^0; rounds R; steps H, E; batch size B; learning rate η

Initialize $\phi_k^0 \leftarrow \phi^0$ and $p_k \leftarrow n_k / \sum_j n_j$ for each client.
for $t = 0, \ldots, R-1$ **do**
  Server broadcasts θ^t.
  for each client $k = 1, \ldots, K$ **do**
    $\phi_k^{t+1} \leftarrow \mathrm{SGD}_\phi^H(\phi_k^t; \theta^t, \mathcal{D}_k)$.
    $\widetilde{\theta}_k^{t+1} \leftarrow \mathrm{SGD}_\theta^E(\theta^t; \phi_k^{t+1}, \mathcal{D}_k)$.
    Retain ϕ_k^{t+1} locally; upload $\widetilde{\theta}_k^{t+1}$.
  end for
  Server aggregates CFE parameters using Equation 5.
end for
return $\{(\theta^R, \phi_k^R)\}_{k=1}^K$.

The norms require contextual appropriateness alone, additional recipient verification, or additional scoped consent. Each client receives 96 training and 32 validation examples from distinct families within each split. Families can recur across clients for comparing norms on shared tasks. Inputs and labels follow Section 3.

Backbone transfer. Following Section 3's transfer protocol, backbone slots A–D form two folds: F_D trains on A/B/C and transfers to D; F_C trains on A/B/D and transfers to C. Each client's source and target tests share 124 scenario variants from 65 held-out families, with candidates generated by the corresponding backbone. Appendix C.6 details the training and evaluation configuration.

Training comparisons. Table 1 compares Local, FedAvg (McMahan et al., 2017), FedAvg with local fine-tuning (FT), FedPer (Arivazhagan et al., 2019), Ditto (Li et al., 2021), and FCDC. All share the ModernBERT-base CFE, linear NDM, within-repeat initialization, labeled data, and visible inputs; selection uses source validation. Local matches FCDC's alternating updates. FedAvg updates and aggregates both blocks; FedPer updates both but aggregates only the CFE. FT adapts the whole CI module on source training data; Ditto maintains a proximal personalized model alongside a global model. Block updates, sample visits, and communication are accounted for separately; equal rounds do not imply equal computation (Appendix C.5).

Agent-task comparisons. Table 2 compares no additional guard, CI-CoT (Lan et al., 2025), PrivacyChecker (Wang et al., 2025), an adapted CI supervisor (Ghalebikesabi et al., 2025), Local-CI, FedAvg+FT-CI, and FCDC. PrivacyLens-derived evaluation (Shao et al., 2024) scores final actions after fixed trajectory prefixes on 150 cases. MAGPIE-derived evaluation (Juneja et al., 2025) runs complete interactions on 80 cases, continuing after intervention; protection covers one focal agent, with peers using B. Both use CI modules trained in F_D, with three training checkpoints and three rollout repetitions per checkpoint, for nine evaluations with each of A and D. Within each evaluation, the checkpoint's CI parameters and source-selected threshold remain fixed. Systems share cases, actor backbones, interaction limits, and scoring rules; prompting and supervisory methods generate their own actions and trajectories. The 50/20 PrivacyLens/MAGPIE development cases use A for adapter checks, without CI fitting or threshold selection (Appendix E).

Metrics and uncertainty. We report equal-client means of balanced accuracy (BA), false allow rate (FAR) among prohibited candidates, and false block rate (FBR) among permitted candidates. BA weights both errors equally (Equation 7). Means and sample standard deviations (SDs) cover three paired allocation/training repeats per fold, reflecting both sources of variability. Pair is the percentage of counterfactual input pairs with both decisions correct after a context change reverses the labels. We retain the benchmarks' original privacy and utility metrics. PrivacyLens (Shao et al., 2024) reports case leakage rate (LR), mean task-fulfillment helpfulness (Help, 0–3), and LR among Help ≥ 2 actions (ALR; originally LR_h). For MAGPIE (Juneja et al., 2025), we report the fraction of focal private items partially or fully disclosed (Leak). Its task-completion score (TC) averages the fraction of task and participant constraints met by the last proposal, regardless of agreement; consensus rate (CR) is the fraction of episodes in which all participants accept the same proposal. We

additionally report thousands of episode tokens (kTok). Paired family-bootstrap intervals quantify scenario-sampling uncertainty (Appendix D).

5.2 Collaborative Learning

Value of cross-client learning. Table 1 gives FCDC the highest seen-backbone BA: 85.9%/85.2% in F_D/F_C, exceeding Local by 8.6/9.9 percentage points. With architecture, supervision, initialization, and local updates matched, CFE exchange supplies the additional cross-client learning signal. FCDC also exceeds the strongest personalized comparator in each fold: Ditto by 2.8 points and FedPer by 3.5 points. FedPer already retains local NDMs; the mechanism analysis in Section 5.5 examines the update procedure beyond this partition. These rankings compare complete procedures under their specified budgets (Appendix C.5).

Selective disclosure. Relative to Local, FAR falls by 14.3/16.3 points and FBR by 2.8/3.4 points: the gains combine fewer prohibited disclosures allowed with fewer permitted actions blocked. Differences throughout use unrounded values.

Table 1: Federated CI decisions before and after main-agent replacement.

	Seen main Agents			Held-out main Agent		
Method	BA ↑	FAR ↓	FBR ↓	BA ↑	FAR ↓	FBR ↓
(a) Fold F_D: train on A/B/C; test transfer to D						
Local	77.4±1.1	26.3±1.4	19.0±0.8	70.7±1.4	33.8±1.8	24.9±1.0
FedAvg	77.5±1.3	23.9±1.7	21.0±0.9	69.3±1.5	32.7±2.1	28.7±0.9
FedAvg + FT	81.9±1.6	18.5±2.0	17.6±1.2	74.9±1.9	25.8±2.4	24.3±1.5
FedPer	82.9±1.3	16.1±1.6	18.2±1.1	76.1±2.2	23.3±2.5	24.5±1.8
Ditto	83.1±2.3	17.4±2.6	16.3±2.0	76.9±1.7	24.1±2.0	22.1±1.3
FCDC (ours)	85.9±2.1	12.0±2.5	16.1±1.7	81.6±1.1	16.8±2.0	19.9±0.3
(b) Fold F_C: train on A/B/D; test transfer to C						
Local	75.4±0.8	29.1±1.1	20.1±0.6	68.3±0.9	36.2±1.4	27.2±0.5
FedAvg	75.7±1.1	24.4±1.3	24.2±0.9	68.2±1.3	32.3±1.6	31.4±1.0
FedAvg + FT	80.7±0.8	19.4±1.2	19.3±0.4	73.4±1.6	28.7±1.9	24.6±1.4
FedPer	81.7±0.7	12.4±0.9	24.2±0.7	76.8±1.0	17.8±1.2	28.7±0.8
Ditto	81.5±0.7	19.1±1.0	18.0±0.5	75.7±2.0	24.8±2.7	23.8±1.3
FCDC (ours)	85.2±0.9	12.8±0.9	16.8±0.9	79.4±1.2	20.1±1.5	21.1±0.9

Percentages; nine-client macro means ± sample SD over three paired repeats. Seen and held-out tests share families; transfer fixes the norm, CI modules and threshold without target adaptation. Methods share architecture, labels and visible inputs; selection uses source validation.

5.3 Transfer across Agent Backbones

Under fixed-module transfer within existing clients, without target adaptation (Section 5.1), FCDC reaches target BA of $81.6 \pm 1.1\%$/$79.4 \pm 1.2\%$ in F_D/F_C. Its gains over Local are 11.0/11.1 points; those over the strongest personalized comparators, Ditto/FedPer, are 4.7/2.6 points. A smaller source-to-target loss alone does not identify the better target evaluator: in F_C, FedPer loses only 4.9 points but reaches 76.8% BA, below FCDC's 79.4%.

Distribution of transfer gains. All nine client means are positive against both Local and FedAvg+FT in both folds (Figure 2), locating the aggregate advantage throughout the evaluated client set. These are gains in balanced accuracy; individual FAR reductions and benefits to unseen clients do not follow. The figure compares complete procedures, with Local's matched schedule and FT's different budget.

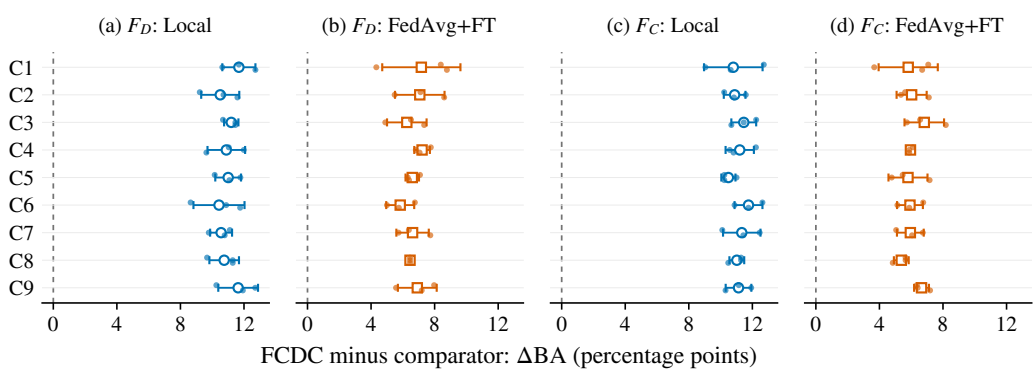

Figure 2: Held-out BA gains (ΔBA, percentage points): FCDC minus Local/FedAvg+FT in F_D (a,b) and F_C (c,d). Dots: three paired repeats; open markers/bars: mean/sample SD. Positive values favor FCDC. Shared client indices C1–C9 have no recorded norm/backbone mappings.

Both target error rates remain below Local's, but F_C exposes a tradeoff against FedPer: FCDC has higher FAR (20.1% versus 17.8%) and lower FBR (21.1% versus 28.7%). Its BA advantage therefore reflects a more balanced decision rule, with task consequences assessed next. Appendix F.2 examines threshold dependence.

5.4 Privacy and Utility in Agent Tasks

Protection with useful final actions. On PrivacyLens, FCDC records LR of 8.0%/12.0% and Help of 2.43/2.29 with A/D (Table 2). Relative to Local-CI, LR falls by 12.0/16.7 points while Help rises by 0.32/0.31; both metrics also improve over the evaluated existing defenses and FedAvg+FT-CI. This joint change connects selective disclosure to more useful final actions. ALR conditions on each system's own helpful outputs and is read alongside LR and Help, rather than as a comparison on a common subset.

Protection during continued interaction. On MAGPIE, FCDC reduces focal Leak against Local-CI by 15.0/21.0 points (95% intervals $[-20.5, -10.0]$/$[-27.4, -15.1]$) while increasing TC by 7.4/11.0 points. These differences combine lower disclosure with greater constraint fulfillment after intervention and continued interaction. Consensus remains distinct: CR intervals include zero against Local-CI with A and against FedAvg+FT-CI with both backbones (Figure 6), leaving those comparisons unresolved.

Utility and cost. The unguarded system retains higher Help, TC, and CR, alongside greater leakage. FCDC uses 9.53/10.14 kTok per episode, below the adapted supervisor's 13.34/14.64 but above the unguarded 8.76/9.28. These totals include actor, peer, and defense calls, excluding CI-encoder computation and offline evaluation; they measure token consumption, not wall-clock efficiency. MAGPIE protection covers the focal participant, and the paired family-bootstrap intervals describe scenario-sampling uncertainty.

5.5 Mechanism Analysis

The ablations examine the learning roles in Section 4.2: No federation tests CFE exchange, Shared NDM tests persistent local decision states, and Joint tests adaptation before representation updates. All retain policy input; counterfactual probes test whether decisions respond correctly to changes in normative context.

Shared learning and local norm adaptation. Full FCDC exceeds No federation and Shared NDM in Table 3. The former matches local updates; the latter also aggregates NDM parameters while retaining policy input, testing personalization beyond policy conditioning. Against Shared NDM, Full reduces FBR more than FAR (8.3 versus 3.4 points), locating most of the BA gain in recovering permitted flows. The contrasts connect CFE exchange to stronger classification and local

Table 2: Privacy and utility under the original PrivacyLens and MAGPIE metrics (Section 5.1).

	PrivacyLens			MAGPIE			
System	LR ↓	ALR ↓	Help ↑	Leak ↓	TC ↑	CR ↑	kTok ↓
(a) Seen main Agent A							
No additional guard	32.7	34.6	2.54	33.0	90.3	90.0	8.76
CI-CoT	22.0	24.0	2.35	23.0	85.9	81.3	10.07
PrivacyChecker	16.7	17.4	2.21	16.0	82.9	80.0	10.37
CI supervisor (adapted)	10.7	12.9	1.98	11.0	76.6	72.5	13.34
Local-CI	20.0	22.9	2.11	22.0	80.0	76.3	10.04
FedAvg + FT-CI	14.0	15.4	2.28	15.0	83.2	81.3	9.66
FCDC (ours)	8.0	7.8	2.43	7.0	87.5	86.3	9.53
(b) Held-out main Agent D							
No additional guard	39.3	42.1	2.36	40.0	85.6	82.5	9.28
CI-CoT	28.0	33.3	2.14	31.0	79.4	75.0	10.86
PrivacyChecker	22.0	25.7	2.07	24.0	76.9	71.3	11.05
CI supervisor (adapted)	14.0	18.3	1.83	16.0	69.6	65.0	14.64
Local-CI	28.7	34.7	1.98	32.0	71.9	67.5	10.91
FedAvg + FT-CI	21.3	24.5	2.13	22.0	76.6	73.8	10.43
FCDC (ours)	12.0	12.6	2.29	11.0	82.9	80.0	10.14

Rates are percentages; Help ranges from 0 to 3; kTok counts thousands of episode tokens. Point estimates use 150/80 PrivacyLens/MAGPIE cases; paired intervals appear in Appendix G.

NDMs to reduced overblocking. Frozen CFE trains only local NDMs, removing encoder adaptation and meaningful exchange together; its lower BA cannot separate their contributions. On held-out D, Full lowers Leak relative to Shared NDM, but TC and CR intervals include zero (Appendix H).

Update schedule sensitivity. Joint updates both blocks on each minibatch. Full exceeds Joint in BA and by 6.1 Pair points (Table 3), favoring adaptation of the local decision rule before shared representation updates under these budgets. At common rounds and batch size, Joint matches sample visits and CFE communication, but updates each block $R(H+E)$ times versus FCDC's RE/RH. The contrast therefore combines update order and per-block effort; equal sample visits do not establish equal computation. All four live difference intervals include zero (Appendix H.4), leaving interaction-level differences unresolved. The kTok values measure episode inference, not training cost.

Table 3: Mechanism diagnostics (%). CI aggregates both transfer folds; MAGPIE evaluates held-out Agent D.

	Held-out CI: mean ± SD				MAGPIE: point estimates	
Variant	BA ↑	FAR ↓	FBR ↓	Pair ↑	Leak ↓	TC ↑
No federation	69.5±1.1	35.0±1.6	26.0±0.7	45.4±1.4	32.0	71.9
Shared NDM	74.7±0.8	21.8±0.8	28.8±0.8	53.8±1.1	19.0	77.5
Joint ($H+E$)	76.4±0.5	22.4±0.6	24.7±0.6	62.3±1.7	16.0	80.7
Frozen CFE	66.8±0.5	34.5±0.5	31.9±0.5	41.0±1.6	35.0	68.8
FCDC (full)	80.5±1.1	18.5±1.7	20.5±0.6	68.4±0.9	11.0	82.9

CI: nine-client macro, then equal-fold mean within each of three repeats; sample SD across repeats. Pair uses 32 separate pairs/client/fold from 16 families. MAGPIE: 80 cases/38 families; CI modules trained in F_D, tested with D. Full/No federation reuse FCDC/Local BA/FAR/FBR and live records from Tables 1–2. Complete live metrics and intervals: Appendices H and H.4.

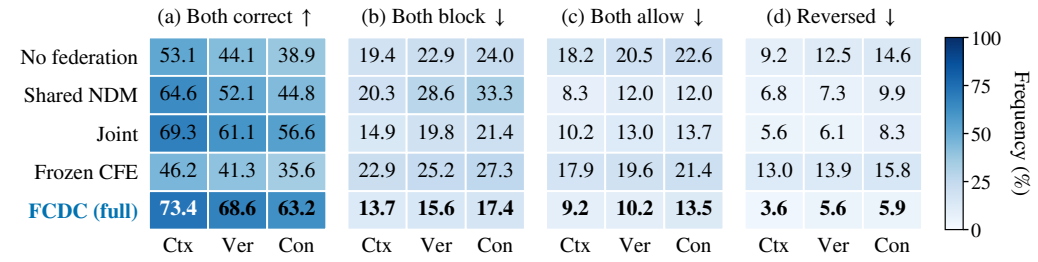

Figure 3: Paired outcomes (%): both correct, both block, both allow, and both wrong (a–d). Ctx/Ver/Con: contextual-minimum/verified-recipient/explicit-consent norms. Means over three repeats, two folds, three clients/norm and 32 pairs/client/fold. Full reduces both-block errors most for verification and consent relative to Shared NDM.

Norm sensitivity. Full's Pair advantage over Shared NDM mainly reflects fewer both-block outcomes within each norm (Figure 3). These errors reject both the permitted and prohibited member; their reduction identifies recovery of permitted flows. Both-allow errors fall for verification but rise for contextual minimum and consent, exposing a privacy tradeoff. The pairs fix task, history, policy, candidate, model state, and threshold. Recipient-role, verification, and consent-receipt interventions correspond to the three norms; comparisons remain within each known policy (Appendix H.3).

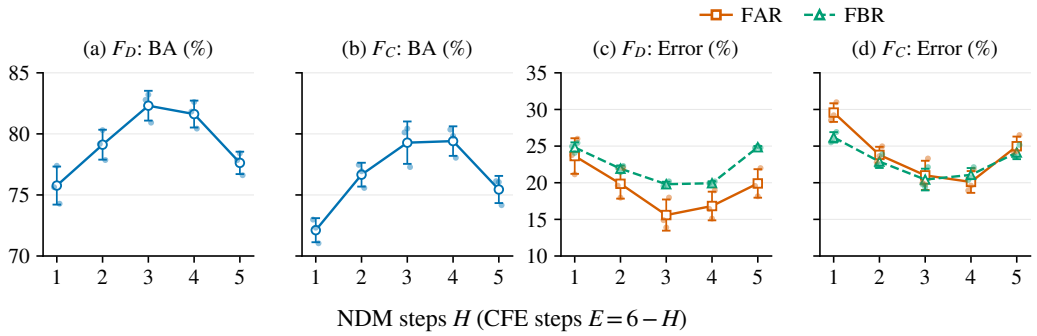

Figure 4: H/E allocation ($H + E = 6$): BA (a,b) and FAR/FBR (c,d) on held-out F_D/F_C. Dots: three nine-client macro repeats; bars: mean ± sample SD. Interior allocations lower mean FAR/FBR versus endpoints.

Allocation between local adaptation and shared learning. The $(H, E) = (3, 3)$ and $(4, 2)$ allocations have higher BA and lower FAR/FBR than endpoints (Figure 4). Within F_C, shifting one update from CFE to NDM between these allocations lowers FAR but raises FBR, despite similar BA. This decomposition exposes the error balance hidden by aggregate accuracy; it adds no independent evidence beyond BA. Holding $H + E$ fixed changes both blocks' update counts while controlling sample exposure and communication at fixed rounds and batch size (Appendix F.1). The sweep neither matches computation nor isolates local adaptation as the cause of the ranking.

6 Conclusion

FCDC integrates federated contextual disclosure control with nested learning for local norm adaptation and shared representations. Within the evaluated setting, it supports collaboration and reuse across main-agent backbones in existing clients. Nested training links a compatible shared CFE to persistent local NDMs, preserving local norm adaptation. Comparisons favor FCDC overall and connect CI decision quality to action-level and closed-loop privacy–utility outcomes, supporting disclosure control as a reusable capability across heterogeneous deployments.

AI USE STATEMENT

Generative AI tools, including OpenAI Codex, assisted with literature discovery and synthesis, refinement of the conceptual framework, mathematical formulation, experiment design, data-preparation and training code, result interpretation, manuscript drafting and editing, and table and figure preparation. AI-assisted cross-checks examined references, code–method consistency, and claim scope. Language-model roles within the experimental protocol are described separately in Section 5.1 and Appendix E. The authors retain responsibility for the accuracy, originality, and evidential support of the final text, code, analyses, and claims, including material developed with AI assistance.

ETHICS STATEMENT

This study uses benchmark scenarios derived from CI-RL, PrivacyLens, and MAGPIE with simulated client assignments (Appendix B). The specified policies define evaluation conditions rather than universal privacy preferences. Both inappropriate disclosure and unnecessary blocking can harm users, motivating joint evaluation of privacy and task utility. Data preparation retains source revisions and license metadata; redistribution of source or derived materials must respect the applicable source terms. The framework supports context-dependent disclosure decisions, and its use in a deployment requires policies and observable context appropriate to that deployment. Benchmark performance alone does not establish compliance with a deployment's full privacy obligations.

REPRODUCIBILITY STATEMENT

Section 4 and Algorithm 1 specify the learning objective and update procedure. Appendix A defines the input and transfer contract; Appendix B describes source-family splits and client construction; Appendix C details parameter states, gradient paths, and baseline update budgets. Appendix D specifies metric aggregation and paired uncertainty calculations, and Appendix E describes the derived agent-task protocols. The implementation separates data preparation, CI training, and benchmark execution, with configuration and result registers for recording run provenance. Appendix C.6 summarizes the execution configuration, including model assignments, training settings, source-only selection and evaluation procedures.

REFERENCES

Amani Abu Jabal, Elisa Bertino, Jorge Lobo, Dinesh Verma, Seraphin Calo, and Alessandra Russo. FLAP - a federated learning framework for attribute-based access control policies. In *Proceedings of the Thirteenth ACM Conference on Data and Application Security and Privacy*, pp. 263–272. ACM, April 2023. doi: 10.1145/3577923.3583641. URL https://doi.org/10.1145/3577923.3583641.

Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. Federated learning with personalization layers, 2019. URL https://arxiv.org/abs/1912.00818. arXiv:1912.00818.

Eugene Bagdasarian, Ren Yi, Sahra Ghalebikesabi, Peter Kairouz, Marco Gruteser, Sewoong Oh, Borja Balle, and Daniel Ramage. AirGapAgent: Protecting privacy-conscious conversational agents. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pp. 3868–3882. ACM, 2024. doi: 10.1145/3658644.3690350.

Adam Barth, Anupam Datta, John C. Mitchell, and Helen Nissenbaum. Privacy and contextual integrity: Framework and applications. In *Proceedings of the 27th IEEE Symposium on Security and Privacy*, May 2006. URL https://ptolemy.berkeley.edu/projects/truststc/pubs/414.html.

Ali Behrouz, Meisam Razaviyayn, Peilin Zhong, and Vahab Mirrokni. Nested learning: The illusion of deep learning architectures. In *Advances in Neural Information Processing Systems*, volume 38, pp. 46968–47002. Curran Associates, Inc., 2025. doi: 10.52202/085713-1565.

Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Exploiting shared representations for personalized federated learning. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 2089–2099. PMLR, 2021. URL https://proceedings.mlr.press/v139/collins21a.html.

Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. In *Advances in Neural Information Processing Systems*, volume 33, pp. 3557–3568. Curran Associates, Inc., 2020.

Sahra Ghalebikesabi, Eugene Bagdasaryan, Ren Yi, Itay Yona, Ilia Shumailov, Aneesh Pappu, Chongyang Shi, Laura Weidinger, Robert Stanforth, Leonard Berrada, Pushmeet Kohli, Po-Sen Huang, and Borja Balle. Privacy awareness for information-sharing assistants: A case-study on form-filling with contextual integrity. *Transactions on Machine Learning Research*, 2025. URL https://openreview.net/forum?id=l9rATNBB8Y.

Gurusha Juneja, Jayanth Naga Sai Pasupulati, Alon Albalak, Wenyue Hua, and William Yang Wang. MAGPIE: A benchmark for multi-agent contextual privacy evaluation, 2025. URL https://arxiv.org/abs/2510.15186. arXiv:2510.15186.

Guangchen (Eric) Lan, Huseyin A. Inan, Sahar Abdelnabi, Janardhan Kulkarni, Lukas Wutschitz, Reza Shokri, Christopher Brinton, and Robert Sim. Contextual integrity in LLMs via reasoning and reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 38, pp. 104355–104391. Curran Associates, Inc., 2025. doi: 10.52202/085713-3485.

Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 6357–6368. PMLR, 2021. URL https://proceedings.mlr.press/v139/li21h.html.

Tao Lin, Lingjing Kong, Sebastian U Stich, and Martin Jaggi. Ensemble distillation for robust model fusion in federated learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 2351–2363. Curran Associates, Inc., 2020.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pp. 1273–1282. PMLR, 2017. URL https://proceedings.mlr.press/v54/mcmahan17a.html.

Niloofar Mireshghallah, Hyunwoo Kim, Xuhui Zhou, Yulia Tsvetkov, Maarten Sap, Reza Shokri, and Yejin Choi. Can LLMs keep a secret? testing privacy implications of language models via contextual integrity theory. In *International Conference on Learning Representations*, pp. 1892–1915, 2024.

Helen Nissenbaum. Privacy as contextual integrity. *Washington Law Review*, 79(1):119–158, 2004. URL https://digitalcommons.law.uw.edu/wlr/vol79/iss1/10/.

Mohamed Shaaban and Mohamed Elmahallawy. SecureGate: Learning when to reveal PII safely via token-gated dual-adapters for federated LLMs. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 42593–42610, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1972. URL https://aclanthology.org/2026.acl-long.1972/.

Yijia Shao, Tianshi Li, Weiyan Shi, Yanchen Liu, and Diyi Yang. PrivacyLens: Evaluating privacy norm awareness of language models in action. In *Advances in Neural Information Processing Systems*, volume 37, pp. 89373–89407. Curran Associates, Inc., 2024. doi: 10.52202/079017-2837.

Shouju Wang, Fenglin Yu, Xirui Liu, Xiaoting Qin, Jue Zhang, Qingwei Lin, Dongmei Zhang, and Saravan Rajmohan. Privacy in action: Towards realistic privacy mitigation and evaluation for LLM-powered agents. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 17055–17074, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.925. URL https://aclanthology.org/2025.findings-emnlp.925/.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Nathan Cooper, Griffin Thomas Adams, Jeremy Howard, and Iacopo Poli. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2526–2547, Vienna, Austria, July 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-long.127. URL https://aclanthology.org/2025.acl-long.127/.

Jinxi Yu, Eric Hanchen Jiang, Levina Li, Dong Liu, Zhi Zhang, Wenxiao Zhao, Yanxuan Yu, Kai-Wei Chang, and Ying Nian Wu. Privacy-preserving topology-guided safety for llm-based multi-agent systems via federated graph learning, 2026. URL https://arxiv.org/abs/2609.02967. arXiv:2609.02967.

A Problem Definition and Evaluation Contract

A.1 Contextual Norms and the Decision Unit

Contextual integrity evaluates information flows against the norms of a social context (Nissenbaum, 2004). The sender, recipient, and information subject occupy roles that bear on whether a flow is appropriate; information attributes and transmission conditions also matter (Barth et al., 2006). In our operational task, the applicable norm is supplied as policy evidence π_k for client k. The label records whether a candidate action is permitted in its supplied context. This is a supervised decision about a specified norm; the norm itself is part of the task specification.

A candidate action a is the complete message or tool-action payload presented for a decision. A positive label denotes permission to execute that candidate; a negative label denotes prohibition. Task usefulness is evaluated separately: permission alone does not establish that an action accomplishes the task. The unit of classification remains the candidate action, even when its content contains several information attributes.

A.2 Observable Input and Record Boundaries

Table 4 maps the mathematical input to the serialized record. The renderer emits these five fields as JSON with sorted keys and preserves Unicode text. Policy text is available to all compared CI modules. The input retains the candidate's original content; the common interface is a shared record schema, with semantic interpretation performed by the learned model.

Table 4: Input contract for a CI decision. The policy is client-specific; the other fields describe the current candidate and its observable context.

Symbol	Serialized field	Meaning
q	`task`	Task request visible to the CI module.
h	`history`	Observable interaction or task context preceding the candidate.
π_k	`policy`	Applicable norm supplied as textual policy evidence.
a	`candidate_action`	Proposed message or tool-action payload before execution.
u	`trusted_metadata`	Context fields supplied by the task or runtime, such as sender, recipient, information subject, and authorization state.

Here, *trusted* specifies the source of a context field: it is supplied by the environment or task specification, rather than inferred from a candidate's claim about its own permissions. Controlled experiments may supply explicit recipient-verification or consent states as part of a normative variant. Such states are recorded as experimental overlays. They are distinct from a gold allow/deny annotation. Context evidence is treated as given for the decision task; authenticating that evidence is an environment assumption.

Dataset records separately store labels, annotation provenance, sample and family identifiers, client identifiers, and model identifiers. These fields support supervision, routing, and evaluation; the text encoder receives the five-field input in Table 4. In particular, routing a request to client k selects its local state without making its identifier an additional text feature. Gold decisions and annotation explanations are kept outside the observable input.

A.3 Source Families and Within-Client Transfer

For the controlled CI data, a source family is an operational task grouping. Records are connected when they share a normalized domain–intent key (after the declared intent aliases), an identical normalized task, or an identical canonical source item. Connected components define the families. These are inferred task families, rather than verified upstream generation lineages. Train, validation, and test assignments are made at this level before generating normative or agent-model variants. Every derived record inherits its family's split. Record counts and distinct family counts are reported separately; family counts are not a claim of statistical independence.

Each transfer fold specifies a source-model set $\mathcal{M}_{\mathrm{src}}$ and a target backbone $m_\star \notin \mathcal{M}_{\mathrm{src}}$. Client k retains its norm π_k, local decision state, and decision threshold while its source backbone m_k is replaced by $m_\star$. The learned CI parameters remain fixed. Source and target evaluations use the same held-out families, with each backbone generating its own candidates; pairing families does not require identical candidate text. Training, hyperparameter selection, and threshold selection use source-model data. Each fold follows its own source-only selection rule. Results from a fold in which a model is a source are not used to select the configuration of another fold in which that model is held out.

A.4 Decision Errors and Task Outcomes

Let $y = 1$ denote an allowed candidate and $y = 0$ a prohibited candidate, and let $\widehat{y}$ be the predicted decision. For a fixed client and evaluation distribution with both classes present, the two conditional errors are

$$\mathrm{FAR} = \mathbb{P}(\widehat{y} = 1 \mid y = 0), \qquad \mathrm{FBR} = \mathbb{P}(\widehat{y} = 0 \mid y = 1), \tag{6}$$

$$\mathrm{BA} = 1 - \tfrac{1}{2}\big(\mathrm{FAR} + \mathrm{FBR}\big). \tag{7}$$

FAR measures erroneous permission and FBR erroneous blocking. Empirical rates use their corresponding class counts as denominators. An empty denominator produces an undefined rate, not zero error. Client-macro comparisons use the same eligible client set for all three measures; percentage-valued BA equals $100 - (\mathrm{FAR} + \mathrm{FBR})/2$ when FAR and FBR are also percentages.

These errors characterize candidate classification. Interactive task evaluation instead measures information actually disclosed along the executed trajectory, together with task utility. A blocked candidate can change subsequent actions, so closed-loop evaluation continues the interaction after intervention. A fixed-prefix action evaluation measures the resulting action at that prefix. The two evaluation modes answer different questions and are reported separately.

B Data Construction and Annotation

B.1 Source Families and Disclosure Variants

The controlled data use the ContextualIntegritySyntheticDataset released with CI-RL (Lan et al., 2025). Family construction follows Appendix A.3: normalized task intent, declared aliases, and exact duplicate checks are applied before assigning a split. A source item carries a task, available information, and normative reference annotations. The annotations identify contextually appropriate or restricted information; they do not by themselves label an agent's generated candidate.

The prepared corpus contains 729 source records in 312 families. The split seed is 1729. The preparation checks quarantine 28 records with overlapping allowed/disallowed annotation keywords, leaving 701 eligible records. Table 5 distinguishes source records from families and generated decisions. All descendants of a source family inherit its split. The same underlying source can support multiple clients within a split; client datasets are not disjoint collections of independent source scenarios.

Each fold crosses three source backbone slots with three fixed policy texts: *contextual minimum* permits only task-appropriate information flows; *verified recipient* additionally requires confirmation of the intended recipient; and *explicit consent* additionally requires a current receipt scoped to the task and recipient. Neither verification nor consent overrides the underlying contextual restrictions. Two Boolean context fields represent recipient verification and consent confirmation, yielding four context states. The allocation cycles through these states across selected sources; a client assignment uses one state per selected source, rather than all four.

Within a source slot, the three policy clients receive the same source selection. Training and validation each select at most one record per family, using allocation seeds 101, 202, and 303 for the three repeats. Each client has 96 training and 32 validation decisions. Test allocations use all 124 eligible source records from 65 families (the table counts all 66 test families before eligibility filtering), once with the client's source backbone and once with the fold's target backbone.

Table 5: Prepared CI source allocation, before candidate generation. Family counts describe operational task groups, not independent generated decisions.

Split	Source families	Source records	Eligible records
Train	203	501	484
Validation	43	99	93
Test	66	129	124
Total	312	729	701

B.2 Candidate Labels and External Data

The learning target is whether the complete generated candidate complies with the visible policy and trusted context. Evaluation-only reference annotations, including restricted facts and necessity judgments, are kept outside the serialized model input. A reviewed candidate label is attached to its stable source, variant, and generation identifiers. The materialization interface requires a reviewed-label flag before admitting a candidate to training or test.

External benchmarks are reserved for evaluation. The PrivacyLens main set contains 493 cases; the prepared partition assigns 50 to development, 150 to test, and 293 to reserve, stratifying by crowdsourcing, literature, or regulation provenance. Its 150 test cases belong to 150 operational families. The MAGPIE set contains 200 cases in 93 operational families; 20 cases from nine families form development and 80 cases from 38 families form test. The remaining 100 cases are reserved. Family grouping precedes case allocation, and no family is divided between development and test. Source revisions and family definitions are retained with the experiment manifests.

C Model and Optimization Configurations

C.1 Encoding and Local Parameterization

For an input of L token positions, let $v_j \in \mathbb{R}^d$ be ModernBERT-base's final hidden state at position j, and let the attention mask r_j equal 1 for non-padding tokens and 0 otherwise. The CFE output is the masked mean

$$E_\theta(x) = \frac{\sum_{j=1}^{L} r_j v_j}{\sum_{j=1}^{L} r_j}. \tag{8}$$

The encoder parameters θ are shared. Client k retains a linear NDM with parameters $\phi_k = (w_k, b_k)$ and sigmoid output. The loss is binary cross-entropy, computed from logits for numerical stability. The policy is included in the input for every client, including variants that share NDM parameters. Thus, the local NDM provides a personalized parameterization in addition to explicit policy conditioning.

All clients start from the same pretrained CFE and a common randomly initialized NDM. The training routine maintains a separate copy of the NDM for each client. The full encoder is trainable, and the main-agent models are outside the optimization graph. Padding is masked in pooling. Inputs exceeding the configured maximum sequence length raise an error; the encoder interface does not silently truncate the policy or candidate action.

C.2 Finite Adaptation and Gradient Paths

The profiled objective in Equation 4 describes representation quality after local adaptation. The infimum permits an unattained optimum, as can occur for unregularized logistic loss on separable data. Actual training uses a finite number of minibatch steps and retains the resulting local parameters. At round t, let $\psi_{k,s}$ and $\vartheta_{k,r}$ be local NDM and CFE iterates after s and r steps, initialized as $\psi_{k,0} = \phi_k^t$ and $\vartheta_{k,0} = \theta^t$. With sampled minibatches $\mathcal{B}_{k,s}^{\phi}$ and $\mathcal{B}_{k,r}^{\theta}$ for the respective blocks, the updates are

$$\psi_{k,s+1} = \psi_{k,s} - \eta \nabla_\phi \widehat{\mathcal{L}}_k(\theta^t, \psi_{k,s}; \mathcal{B}_{k,s}^{\phi}), \qquad s = 0, \dots, H-1, \tag{9}$$

$$\vartheta_{k,r+1} = \vartheta_{k,r} - \eta \nabla_\theta \widehat{\mathcal{L}}_k(\vartheta_{k,r}, \psi_{k,H}; \mathcal{B}_{k,r}^{\theta}), \qquad r = 0, \dots, E-1. \tag{10}$$

The client retains $\phi_k^{t+1} = \psi_{k,H}$ and uploads $\widetilde{\theta}_k^{t+1} = \vartheta_{k,E}$. Representation gradients treat $\psi_{k,H}$ as fixed; they are partial derivatives of the minibatch loss, with no differentiation through the preceding adaptation trajectory. Thus, finite-step training implements an alternating procedure motivated by the profiled objective, rather than an exact hypergradient method for J.

During NDM updates, the encoder is in evaluation mode and its parameters are fixed. During CFE updates, the encoder is in training mode and the NDM's parameters are fixed, but differentiation passes through the NDM to its input. Both modules are in evaluation mode when scoring candidates. The optimizer is SGD without momentum or weight decay; the same configured learning rate is used for both blocks. Each step draws a batch without replacement within that batch, with fresh draws across steps. No optimizer history is transferred between clients.

C.3 State, Communication, and Deployment

Training uses full participation. Each round starts all local encoder copies from the server's current θ^t, while each NDM starts from its own ϕ_k^t. The server aggregates CFE parameters using training-set sizes; validation and test records do not enter these weights. The local execution can be simulated sequentially while preserving separate client datasets, random sampling streams, and NDM states.

Each client performs RH NDM and RE CFE gradient steps over R rounds. For a per-client batch size $B_k = \min(B, n_k)$, its total training sample visits are $R(H + E)B_k$, counting repeated visits. Communication exchanges the CFE; the NDM and interaction records remain local. The final returned model at each client contains θ^R and its retained ϕ_k^R. There is no post-aggregation NDM fitting stage. Decision thresholds are fixed from source validation, with a common threshold across clients as a permitted special case of the notation τ_k. Held-out-backbone evaluation uses these same states and thresholds.

C.4 Learning Levels and Relation to Prior Work

Nested learning describes models through interacting optimization processes with their own state, learning signals, and update frequencies (Behrouz et al., 2025). Here, the local level updates the NDM from a client's labeled decisions at a fixed representation. The representation level uses gradients conditional on these adapted modules and combines them through federation. Their update counts are H and E per round; the shared CFE is consolidated once per round. With $H > E$, the NDM receives more gradient updates than each local CFE copy. These counts refer to the training schedule, while deployed modules remain fixed.

FCDC assigns these optimization roles to the NDM and CFE for policy-conditioned disclosure judgments from heterogeneous main agents. Its nested CI training retains the norm in the encoder input and the adapted decision module at the client. The profiled objective and alternating schedule are structurally related to FedRep (Collins et al., 2021); here the learning levels operate on disclosure decisions with explicit policy and context evidence. Transfer replaces the main backbone within that existing client while preserving both learned modules and the decision threshold.

C.5 Baseline Updates and Budget Accounting

The comparison methods use a common text-classification backend. A joint step updates the CFE and NDM on the same minibatch. An alternating cycle first takes H NDM steps with the CFE fixed, then E CFE steps with the NDM fixed. Table 6 states the schedules in the training routine. Here R is the number of rounds, $B_k = \min(B, n_k)$, and a sample visit counts an example each time it enters an update, including repeated visits.

FedAvg (McMahan et al., 2017) trains and aggregates both blocks. FedPer (Arivazhagan et al., 2019) retains a client-specific NDM while jointly updating the two blocks locally. Ditto (Li et al., 2021) maintains persistent personal parameters v_k for both blocks and uses the local objective $\mathcal{L}_k(v_k) + \frac{\mu}{2}\|v_k - w^t\|_2^2$, where w^t is the broadcast global model and $\mu \geq 0$ the proximal coefficient. Each round also trains a separate global branch for aggregation. Inference uses the personal branch. The proximal coefficient and other selection choices use source validation within each fold. Local and FCDC have identical block schedules; their difference is CFE communication. FCDC's alternating rule is described in Section 4.

Table 6: Per-client optimization counts for the implemented comparisons. Ditto counts both global and personalized branches. All counts exclude validation inference; configuration-specific values and costs come from run logs.

Method	CFE steps	NDM steps	Sample visits	Aggregated blocks
Local	RE	RH	$R(H+E)B_k$	None
FedAvg	RE	RE	REB_k	Both
FedAvg+FT	RE	RE	REB_k	Both†
FedPer	RE	RE	REB_k	CFE
Ditto	$2RE$	$2RE$	$2REB_k$	Global model
FCDC	RE	RH	$R(H+E)B_k$	CFE

†FedAvg+FT uses $R-1$ aggregation rounds followed by E local joint steps from the final global model; this final phase is included in the budget.

The common backend uses the same encoder, linear NDM, serialization, and loss, with the full encoder trainable. SGD is stateless, and clients participate in every round. These controls fix information access and model capacity. Table 6 makes the different computational budgets explicit: equal R does not equalize sample visits or parameter-block updates. Run-level accounting retains executed updates, sample visits, elapsed run time, and the parameter bytes assigned to each upload/download. These are parameter-payload counts in the sequential simulator, excluding network-protocol overhead and transport timing.

C.6 Training and Evaluation Configuration

Fixed design and model assignments. The backbone identifiers A–D consistently denote model slots across folds and external evaluations. They are not input features. Folds are selected independently using their own source data: for example, a result using C as a source in F_D cannot select a configuration for F_C, where C is the target. The three paired allocation/training seeds are $(101, 7)$, $(202, 17)$, and $(303, 29)$. Table 7 specifies the model families assigned to these slots. Model-card links identify the repositories.

Table 7: Model assignments.

Role	Repository identifier
CFE	`answerdotai/ModernBERT-base`
Actor A	`Qwen/Qwen2.5-7B-Instruct`
Actor B / peers	`meta-llama/Llama-3.1-8B-Instruct`
Actor C	`mistralai/Mistral-7B-Instruct-v0.3`
Actor D	`google/gemma-2-9b-it`
Supervisor / judge	`Qwen/Qwen2.5-32B-Instruct`

Training defaults. Table 8 instantiates the stateless SGD algorithm without changing its gradient paths. The reference allocation $(H, E) = (4, 2)$ lies in the interior of the specified $H+E = 6$ sweep. Methods share the initialized CFE and NDM within each repeat. Inputs use dynamic padding; those longer than 8,192 tokens raise an explicit failure rather than being silently truncated or removed from evaluation.

Each effective batch is drawn once without replacement, then partitioned into eight microbatches. Their losses are weighted by microbatch size before one SGD update; new draws are made for the next update. The encoder is in evaluation mode for NDM updates and training mode for CFE updates. Frozen NDM parameters still permit gradients to its input. The returned model uses the final aggregate and retained local heads, without final head fitting.

At the reference defaults, FCDC performs 120 NDM and 60 CFE updates and 2,880 sample visits per client. FedAvg/FedPer perform 60 joint updates and 960 visits; Ditto performs 60 updates per branch, totaling 120 updates to each block and 1,920 visits. Joint$(H+E)$ performs 180 updates to each block and 2,880 visits. FedAvg+FT uses 29 aggregation rounds followed by two local joint steps, totaling 60 joint updates. These counts instantiate Table 6; they do not imply equal computational cost.

Table 8: Training defaults.

Setting	Value
Clients / participation	Nine per fold; full participation every round
CFE / pooling	Full ModernBERT-base; masked mean, $d = 768$
NDM initialization	$w_j \sim \mathcal{N}(0, 0.02^2)$; $b = 0$; common within repeat
Rounds / block steps	$R = 30$; $H = 4$ NDM steps then $E = 2$ CFE steps
Optimizer	SGD; $\eta = 5 \times 10^{-4}$ for both blocks
Optimizer options	No momentum, weight decay, clipping, warmup or scheduler
Effective batch	$B = 16$; eight microbatches of two records per update
Sequence limit	8,192 tokens including the complete five-field CI input
Precision	BF16 forward/backward; FP32 parameters, accumulated gradients and aggregation
Encoder dropout	MLP dropout 0.1 in training; other dropout at checkpoint defaults
Loss	Unweighted binary cross-entropy from logits
Ditto reference	$\mu = 10^{-2}$, with $\frac{\mu}{2}\|v_k - w^t\|_2^2$
Reference threshold	Common $\tau = 0.50$ for the fixed-configuration reference

Source-only selection. The FCDC search space comprises $R \in \{20, 30, 40\}$, $\eta \in \{10^{-4}, 5 \times 10^{-4}, 10^{-3}\}$, and $(H, E) \in \{(2, 4), (3, 3), (4, 2)\}$: 27 configurations per fold. Evaluating each on all three paired repeats requires 81 complete training runs. The complete five-allocation H/E sweep is a separate sensitivity analysis. FedAvg, FedAvg+FT and FedPer use the same three round and learning-rate choices and $E \in \{2, 3, 4\}$ (27 configurations each). Ditto additionally varies $\mu \in \{10^{-3}, 10^{-2}, 10^{-1}\}$, giving 81 configurations or 243 training runs per fold. Local inherits FCDC's selected block schedule and learning rate to preserve the matched no-federation comparison; structural ablations inherit them as well. These grids allocate different search budgets to the methods.

For each candidate, only the terminal checkpoint at its specified R is eligible. Selection uses equal-client macro BA on the 32 source-validation records per client, averaged over the three repeats. Ties are resolved by lower FAR, fewer CFE updates and finally the smaller learning rate. Client validation summaries can be aggregated without exporting interaction records. Target-backbone records and external benchmark outcomes never select settings.

Threshold selection uses cached source-validation scores and a common threshold per method, fold and repeat from $\tau \in \{0.10, 0.15, \ldots, 0.90\}$ (17 values, no retraining). The criterion is maximum macro BA, with ties resolved by lower FAR, smaller distance to 0.50 and then the higher threshold. Configuration selection uses these validation scores, and the selected threshold remains frozen during testing. The default $\tau = 0.50$ above defines the fixed-configuration reference; selected configurations use their validation-selected thresholds. The dense test-domain threshold curves in Appendix F.2 remain descriptive.

Generation, annotation and intervention. Actors A–D and MAGPIE peers use temperature 0.7, top-p 0.9, 1,024 maximum new tokens per call and a 6,144-token input limit under each actor's tokenizer, with its published chat template. Rollout seeds are 11, 22 and 33 for each checkpoint. Systems share case and seed indices, although diverging trajectories need not consume identical random draws. Non-trained defenses are evaluated in the corresponding nine run slots with distinct random streams; those slots are not nine trained models.

The annotation workflow obtains candidate labels from the 32B model using the complete candidate, explicit policy, trusted context and evaluation-only source references. The reviewed-label flag requires human confirmation of the label and adjudication of ambiguities. Reference annotations and judge explanations remain outside encoder inputs.

The adapted CI supervisor uses the 32B model with greedy decoding and a 512-token output cap, seeing only the allowed five-field input. CI-CoT and PrivacyChecker use the evaluated actor with their respective prompt adaptations. Systems share task information and execution limits. A denied action receives the fixed feedback: "The proposed disclosure does not satisfy the supplied policy. Produce a compliant alternative." At most two regeneration attempts follow the initial proposal. If none is permitted, the outbound action is withheld. For MAGPIE, a rejected focal action yields the turn and the episode continues until consensus or 12 turns per participant; attempts within a turn do

not create extra turns. PrivacyLens scores the final permitted action, or an empty withheld action with helpfulness zero if all three candidates are denied. Parsing or execution errors remain failures, with at most one same-seed technical retry, rather than automatically becoming safe outputs.

Judging and repeated-run summaries. The 32B judge uses greedy decoding and a 1,024-token output cap. It receives the benchmark rubric, case references and observable output or dialogue, with system identities hidden. PrivacyLens judging returns the leakage indicator and 0–3 helpfulness; MAGPIE judging returns none/partial/full for each focal private item and satisfaction for each constraint. Consensus is computed from proposal identifiers and participant acceptance events. The quality-check protocol specifies independent double review of 100 cases per benchmark, sampled across systems and actors, with adjudication of disagreements and reporting of agreement and error rates.

Evaluation comprises three checkpoints and three rollouts per checkpoint. For metric q_{rs} on the complete test set for checkpoint r and rollout s, the summaries are

$$\bar{q}_r = \frac{1}{3}\sum_{s=1}^{3} q_{rs}, \qquad \bar{q} = \frac{1}{3}\sum_{r=1}^{3} \bar{q}_r, \qquad s_{\mathrm{cp}} = \sqrt{\frac{1}{2}\sum_{r=1}^{3}(\bar{q}_r - \bar{q})^2}. \tag{11}$$

Within-checkpoint rollout SDs are separate summaries; s_{cp} does not isolate pure training variance from finite-rollout noise. In particular, ALR aggregation averages per-run ratios and retains each run's helpful-case denominator, rather than replacing them by a pooled ratio. Scenario-only intervals use identical complete-family resamples across methods and all nine run slots, with each run's metric recomputed before averaging paired differences. The bootstrap uses 2,000 resamples, seed 44120260925 and percentile endpoints with linear interpolation. Such intervals remain conditional on the realized run set; they are separate from repeat SDs.

Environment and measurements. The execution environment consists of one NVIDIA A100 80 GB GPU for sequential client training and separate sequential actor/judge inference, with Python 3.11, PyTorch 2.6.0, CUDA 12.4 and Transformers 4.48.3. The encoder excludes its masked-language-model prediction head. Computation uses BF16 autocast and FP32 SGD/aggregation, with separate client datasets, NDMs and sampling streams. Execution records contain model revisions, package versions, prompts, seeds, selected configurations, thresholds, failure counts and per-case predictions. For P_θ shared parameters sent as FP32, each FCDC client uploads and downloads $30 \times 4P_\theta$ bytes in each direction at the default $R = 30$, excluding protocol overhead. Runtime accounting includes GPU hours, peak memory, wall time and communication totals.

D Detailed Results and Statistical Procedures

D.1 Classification Aggregation and Paired Repeats

A classification result first computes BA, FAR, and FBR for each client. The common eligible set contains clients with both label classes, so the three metrics use identical macro weights and obey Equation 7. Per-client class counts, excluded clients, and pooled confusion counts accompany the aggregate. Table 1 keeps the two transfer folds separate. For each metric q, let q_r be the client-macro result in repeat r; the reported mean and sample standard deviation are

$$\bar{q} = \frac{1}{3}\sum_{r=1}^{3} q_r, \qquad s_q = \sqrt{\frac{1}{2}\sum_{r=1}^{3}(q_r - \bar{q})^2}. \tag{12}$$

The allocation and training seeds change together. Accordingly, s_q measures variation across those paired repeats, rather than initialization variance at fixed client data. Methods share the same allocation within a repeat; folds are not counted as six independent repeats. For the offline ablation table, the two fold values are averaged within each repeat before computing the mean and sample standard deviation across repeats.

D.2 Client-Level Paired Gains

For client k, fold f, repeat r, and comparator b, the plotted difference is $\delta^{(b)}_{kfr} = \mathrm{BA}^{\mathrm{FCDC}}_{kfr} - \mathrm{BA}^{(b)}_{kfr}$, in percentage points. Figure 2 retains the three repeat differences for each client and fold, with their mean and sample SD. It does not combine the marginal SDs of the two methods. The client mean of these paired means equals the corresponding difference between the Table 1 macro means under the same support and weights. Clients are displayed in fixed index order, without selection by observed gain. Local and FedAvg+FT are fixed protocol comparisons, not target-selected winners. The figure's 108 differences comprise two folds, two comparisons, nine indices and three repeats; they are not 108 independent experimental repeats.

D.3 External Paired Intervals and Denominators

The external protocol uses three training checkpoints from F_D, with three rollout repetitions per checkpoint (Section 5.1). Point estimates summarize the fixed test sets. To quantify a difference between two systems, the paired bootstrap resamples source families with replacement and includes all cases from each sampled family for both systems. Each resample recomputes the entire metric, including its denominator. This preserves the dependence among variants within a family and pairs the two systems on the same scenarios. Family-bootstrap intervals characterize scenario-sampling uncertainty; training and rollout variation are distinct sources of uncertainty.

Following PrivacyLens (Shao et al., 2024), LR is the mean leakage indicator and Help is the mean 0–3 task-fulfillment score. ALR, the original LR_h, is

$$\mathrm{ALR}_{\geq 2} = \frac{\sum_i l_i \, \mathbf{1}[h_i \geq 2]}{\sum_i \mathbf{1}[h_i \geq 2]}, \tag{13}$$

where i indexes evaluated cases, l_i indicates leakage and h_i is helpfulness. A zero denominator is undefined. Helpful-case counts accompany ALR, and LR and Help remain visible alongside it. The $h_i \geq 2$ threshold follows the original benchmark and is fixed across all systems. For our MAGPIE-derived evaluation (Juneja et al., 2025), Leak pools focal private-item counts across cases. TC averages the fraction of task and participant constraints met by the last proposal, independently of acceptance; CR averages the indicator that all participants accept the same proposal. The additional cost statistic kTok is the mean episode-token count in thousands. All percentage metrics are multiplied by 100 for display. Differences in such metrics are reported in percentage points.

An evaluator or execution failure is retained in the case register. The reporting interface leaves privacy and cost results missing when any case is unresolved; missing records cannot be converted into zero leakage. A completed interaction with no proposal has TC zero, whereas an unscored interaction is a failure. Final comparisons require consistent evaluated supports and explicit failure/retry counts.

E Agent Evaluation Protocols

E.1 Deployment and Method Adaptations

External evaluation uses CI modules trained in F_D, with three training checkpoints and three rollout repetitions per checkpoint. Each checkpoint is evaluated with A (seen) and D (held out), without fitting on external cases; its learned CI parameters and source-selected threshold remain fixed during the rollouts. This protocol is used for Local-CI, FedAvg+FT-CI, and FCDC. The policy remains explicit in the model input. Development cases use A to check adapters and task validity; CI parameter, threshold, and model selection remain confined to source CI data. Compatibility between a benchmark task and the routed policy is evaluated before test comparisons.

No additional guard retains the task's native privacy instructions. CI-CoT prompts the actor to reason about contextual integrity before producing an action (Lan et al., 2025). PrivacyChecker uses its system-prompt deployment for information-flow analysis (Wang et al., 2025); task and recipient placeholders are adapted to the benchmark. The CI supervisor adapts the information-flow supervision of Ghalebikesabi et al. (2025) to proposed agent disclosures and returns a decision to the actor. It uses visible task, policy, and candidate information, without gold information-flow tuples.

Each method produces its own actions; prompting methods are not evaluated by relabeling cached unguarded candidates. Task budgets are shared across systems, and additional model calls enter cost accounting.

E.2 PrivacyLens-Derived Action Evaluation

PrivacyLens (Shao et al., 2024) supplies a fixed interaction prefix and a task that elicits an outbound action. Each system receives the same case prefix and available tools, then produces its final action under its own protection mechanism. Leakage and helpfulness are evaluated on that final output using the benchmark's scoring criteria. Help follows its ToolEmu-based rubric: 0 (Poor), 1 (Unsatisfactory), 2 (Good), or 3 (Excellent), according to whether the final action fulfills the user instruction. Wrong-recipient actions and nonexistent attachments score 0. ALR is the original LR_h, restricted to Help ≥ 2. An intervention cannot inherit the helpfulness of a candidate that was withheld. Required-action parsing failures are recorded as failures, not automatically scored as safe. The evaluation assesses the action following the supplied prefix; it does not measure a complete task restarted from its initial state or undo information already disclosed in that prefix.

E.3 MAGPIE-Derived Interactive Evaluation

MAGPIE (Juneja et al., 2025) provides collaborative tasks with separate agent views and private information. The focal participant is role 0 and uses A or D; peers use B. Protection applies to focal outbound messages, proposals, and acceptance/rejection reasons. An intervention withholds the candidate and returns policy feedback, after which the actor continues the interaction subject to the shared retry and episode limits. Private memories remain separate. This evaluates the consequences of intervention on subsequent behavior, not retrospective deletion from an already generated dialogue.

Leak counts each focal private information item once as none, partial, or full disclosure over the dialogue; the numerator is the number with partial or full disclosure and the denominator is the number of focal private items. TC retains MAGPIE's task-completion score: the fraction of task and participant constraints satisfied by the last proposal, averaged over cases, and is zero if there is no proposal. These constraints specify task requirements and participant needs; they are distinct from disclosure norms. TC does not require consensus. CR is the proportion of cases in which every participant accepts the same proposal identifier. These participants belong to the evaluation task and are distinct from federated training clients; negotiation is a property of this test environment, not a requirement of FCDC. The additional cost statistic kTok counts actor, peer, and defense input/output tokens per episode, divided by 1,000. It excludes offline evaluation and CI-encoder computation, with call categories fixed across methods. The protected scope is the focal participant, and all-participant leakage is a separate diagnostic.

Protocol provenance. These are derived protocols on fixed subsets; their results belong to the common evaluation described here rather than to the original papers' complete benchmark settings. The source snapshots are retained in the data manifests.

F Sensitivity Analyses

F.1 H/E Allocation Sensitivity

Figure 4 varies (H, E) over $(1, 5)$ through $(5, 1)$ at fixed $H + E = 6$. Every setting alternates NDM and CFE updates; this differs from the Full–Joint comparison, which changes whether the blocks are updated separately. The figure reports each fold's nine-client macro mean and sample SD over three paired repeats.

Budget and interpretation. For fixed $S = H + E$, rounds R and $B_k = \min(B, n_k)$, a client processes RSB_k sample visits and exchanges R CFE payloads in each direction. The NDM and CFE receive RH and RE updates, respectively. Reallocation therefore holds sample exposure and communication fixed, but changes block-update counts and potentially runtime. It does not isolate update order or establish a convergence advantage. Paired comparisons must hold client data, initialization,

batch sampling, learning rate and threshold fixed. Configuration selection uses source validation; target curves are descriptive, and similar BA alone does not establish equivalent FAR/FBR.

Figure 4 connects only the five evaluated settings and makes no interpolation-based optimization claim.

F.2 Decision-Threshold Tradeoffs

Figure 5 varies the disclosure threshold while holding candidate scores, labels and model state fixed. Let s_i be the allow score of candidate i, and let $\mathcal{P}_k$ and $\mathcal{N}_k$ contain the allowed and prohibited candidates for client k. Then

$$\mathrm{FAR}_k(\tau) = \frac{\sum_{i\in\mathcal{N}_k} \mathbf{1}[s_i \geq \tau]}{|\mathcal{N}_k|}, \qquad \mathrm{FBR}_k(\tau) = \frac{\sum_{i\in\mathcal{P}_k} \mathbf{1}[s_i < \tau]}{|\mathcal{P}_k|}. \tag{14}$$

Rates are displayed as percentages. Each fold averages nine clients within each repeat, then averages three repeats. Increasing τ cannot increase FAR or decrease FBR on these fixed candidates.

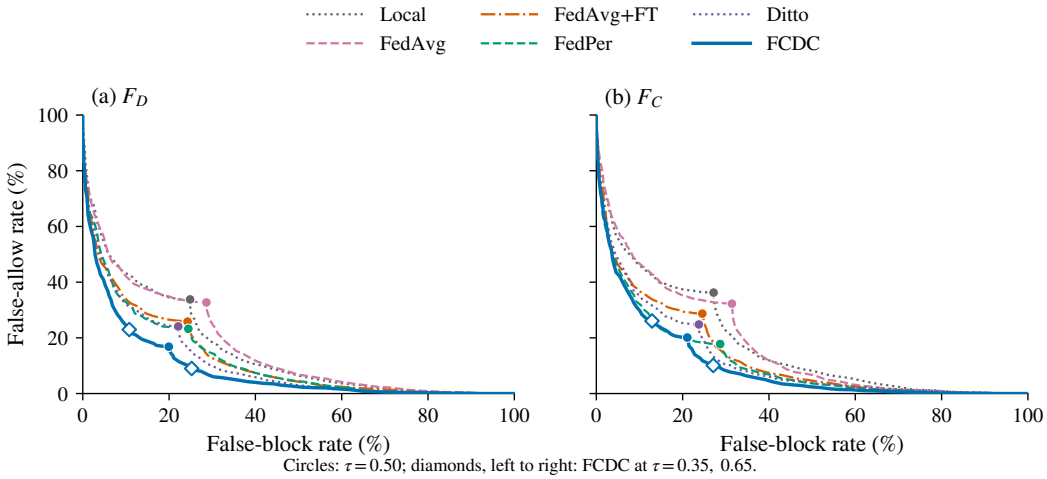

Figure 5: Frozen-score threshold tradeoffs on held-out candidates: six methods, nine-client macro means over three repeats, $\tau = 0, 0.01, \ldots, 1$. Lower-left is preferable. Circles ($\tau = 0.50$) recover the reference counts; FCDC diamonds mark $0.35/0.65$ from left to right. These are sensitivity-analysis thresholds, not source-selected operating thresholds; no confidence intervals are shown.

For FCDC, moving τ from 0.50 to 0.65 lowers FAR from 16.8% to 9.0% in F_D and from 20.1% to 10.1% in F_C, while raising FBR from 19.9% to 25.2% and from 21.1% to 27.1%, respectively. This trades prohibited disclosures against blocked authorized actions; it does not change the underlying score function. Lower macro FAR at a matched macro FBR describes a better aggregate error tradeoff under the common-threshold policy. Its position depends on both within-client ranking and cross-client score scales, so curve separation cannot be attributed solely to better CI representations.

Scope. The figure shows no confidence intervals. Thresholds must be selected on source validation and frozen before testing. Client-specific thresholds define a different policy. These candidate-level curves assess neither calibration nor end-to-end leakage: intervention changes subsequent interactions, requiring fresh matched rollouts for a task-level threshold comparison.

G System Result Accounting

PrivacyLens ALR uses each system's own helpful-action subset (Help $\geq$ 2). Table 9 retains the exact denominators needed to interpret this conditional rate alongside LR and Help. Changes in subset membership mean that ALR alone is not a comparison on identical cases.

Table 9: Helpful-action denominators. Every system has 150 cases per actor; ALR is 100 times the leaking-eligible count divided by eligible count.

	Actor A		Actor D	
System	Eligible	Leaking eligible	Eligible	Leaking eligible
No additional guard	130	45	121	51
CI-CoT	121	29	108	36
PrivacyChecker	115	20	105	27
CI supervisor (adapted)	101	13	93	17
Local-CI	109	25	101	35
FedAvg + FT-CI	117	18	106	26
FCDC (ours)	128	10	119	15

Paired intervals use 2,000 common family resamples within each benchmark/actor block (seed 44120260925), recomputing every metric and its denominator. Endpoints are the 2.5th and 97.5th percentiles with linear interpolation. A resample with undefined ALR is excluded from that paired ALR contrast and counted. Figure 6 shows the MAGPIE intervals for all six comparators, under both actor conditions.

G.1 Paired End-to-End Stability Analysis

Figure 6 shows FCDC-minus-system differences for all six comparators under A and D. Each bootstrap draw samples complete families and uses the same cases for every system, recomputing pooled focal-item Leak, mean per-case constraint satisfaction (TC) and consensus frequency (CR). Thus, bars describe paired differences rather than overlap of marginal system intervals. Sampling and metric-accounting details are in Appendix G.

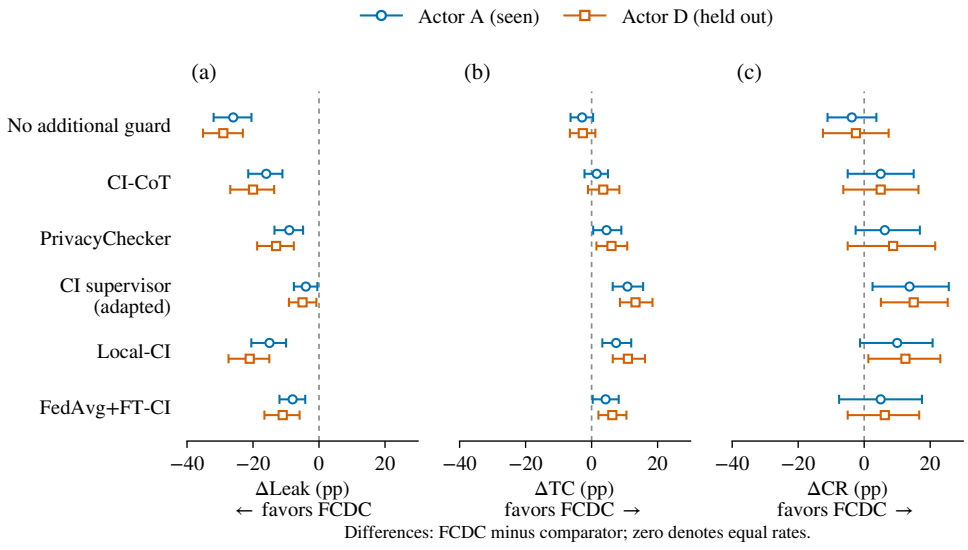

Figure 6: FCDC-minus-system differences (percentage points) for A/D, using 80 cases in 38 families per condition. Bars: 95% pointwise paired family-bootstrap intervals, 2,000 common resamples. Negative Leak and positive TC/CR favor FCDC. Privacy and utility comparisons have different sign patterns.

Among the ten added-defense comparisons (five systems, two actors), 10 Leak intervals lie below zero, 8 TC intervals lie above zero, and 3 CR intervals lie above zero. The different sign patterns distinguish disclosure protection, satisfying proposal constraints and securing participant agreement. Higher TC need not imply higher CR; aggregate intervals do not identify the trajectory mechanism. TC against CI-CoT and CR against FedAvg+FT-CI remain unresolved in both actor conditions. The

unguarded reference has higher TC/CR point estimates, but all four corresponding intervals include zero.

These are pointwise, unadjusted intervals. They do not establish utility equivalence or a nonzero A-to-D change in a method gain. More resamples reduce bootstrap Monte Carlo error without supplying new training runs or interactions. PrivacyLens evaluates a final action after a fixed prefix and is outside this interactive stability analysis.

H Mechanism Diagnostics

H.1 Fold-Resolved Ablations

Figure 7 separates the two transfer folds that Table 3 averages. It combines the structural and schedule ablations with the auxiliary Pair diagnostic, retaining all five variants. Each classification repeat averages nine clients with 124 candidates each; Pair uses a separate probe set (Appendix H.3). The mean and sample SD use three paired repeats within each fold. In the main table, folds are averaged within each repeat before this final aggregation.

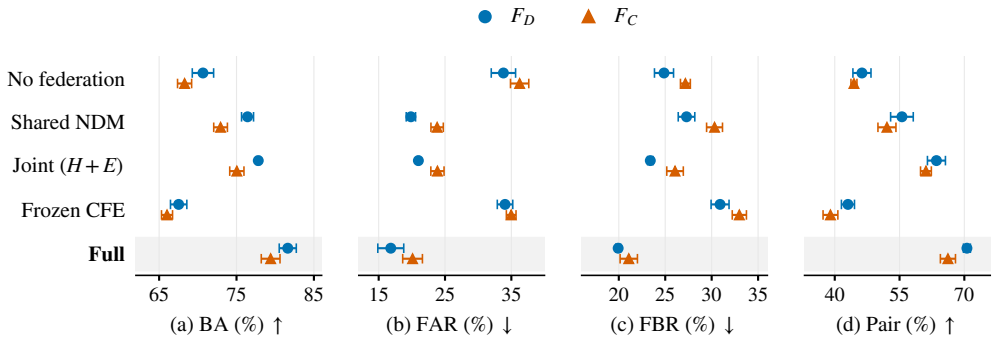

Figure 7: Fold-resolved ablations: nine-client macro means ± sample SD over three paired repeats in F_D/F_C. BA/FAR/FBR use 124 candidates per client; Pair uses 32 separate probes per client and fold. Joint uses more block updates at equal sample visits and CFE payload (Appendix H.4). Full's mean advantage occurs in both folds.

Full has higher mean BA and Pair and lower mean FAR and FBR than each ablation in both folds, so averaging folds does not create the ordering. This view checks directional consistency across two specified transfer conditions; it is neither a new experiment nor a test of generalization to arbitrary backbones. Error bars describe repeat variation, not uncertainty in a paired method difference.

H.2 Paired Interactive Contrasts

Figure 8 complements the Leak/TC point estimates in Table 3 with paired intervals, consensus and token cost. CI modules trained in F_D are deployed with D. All variants use the same 80 cases in 38 families, 400 focal private items and 480 constraints. The bootstrap follows Appendix G.1; token cost includes actor, peer and defense input/output tokens but excludes CI-encoder computation and offline evaluation.

Lower leakage is separated from utility and cost: the Shared NDM TC, CR and kTok intervals include zero, and the Frozen CFE CR interval touches zero. All four Joint intervals cross zero, leaving those interaction-level differences unresolved despite the offline mean ordering. These are pointwise, unadjusted paired family-bootstrap intervals.

Record consistency. The figures reuse the same evaluation records. Full is identical across the structural, norm and schedule sources; No federation reuses Local's alternating-update records. The five-variant classification display contains 270 client confusion matrices, and the probe display contains 8,640 repeated pair predictions. These counts are record totals, not independent scenarios.

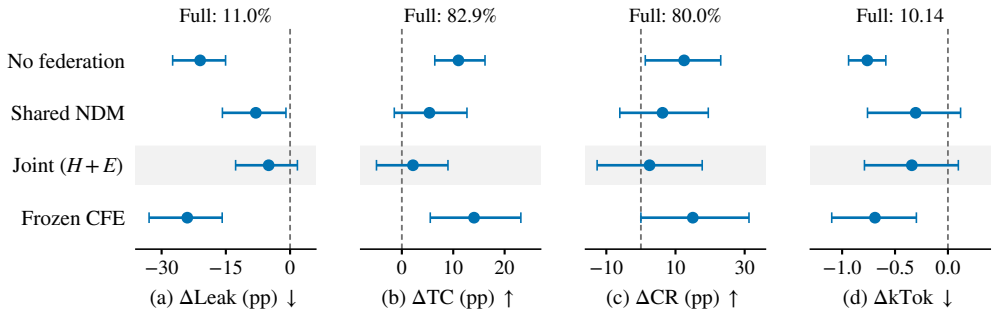

Figure 8: Full-minus-ablation contrasts on Agent D: 80 cases in 38 families; bars are 95% pointwise paired family-bootstrap intervals (2,000 resamples). Joint uses more block updates at equal sample visits and CFE payload (Appendix H.4). Arrows indicate the favorable direction; pp denotes percentage points. Full reference values appear above each panel. All four Joint intervals cross zero.

H.3 Paired Norm Probes

The preparation protocol assigns one verification/consent state per source; it does not supply the paired inputs required here. Pair therefore uses a separate diagnostic set and cannot be recovered from classification confusion matrices.

For client k, fold f and repeat r, let P_{kfr} contain input pairs (x_0, x_1) with different stipulated gold labels. With y_j and $\widehat{y}_j$ denoting the gold and predicted labels of member $j \in \{0, 1\}$, and $n_{kfr} = |P_{kfr}|$, define

$$\text{Pair}_{kfr} = \frac{100}{n_{kfr}} \sum_{(x_0,x_1)\in P_{kfr}} \mathbf{1}[\widehat{y}_0 = y_0 \ \wedge \ \widehat{y}_1 = y_1]. \tag{15}$$

Both members use the same model state and threshold. Task, history, policy and candidate text remain fixed; only the trusted-context field in Table 10 changes. Selection uses the stipulated labels and task contract, not method predictions. Gold labels and pair identifiers never enter the five-field model input.

Table 10: Norm probes: three clients per norm and fold, with 32 pairs per client from the same 16 families.

Client norm	Changed context field	Fixed prerequisites
Contextual minimum	Recipient role: designated collaborator / unrelated vendor	Same task-relevant message
Verified recipient	Recipient verification: true / false	Same designated recipient and message
Explicit consent	Current scoped receipt: present / absent	Same verified recipient and message

Each client/fold has 16 allow-first and 16 deny-first pairs. Across nine clients and two folds there are 576 pairs (1,152 members), reused across three repeats and five variants for 8,640 prediction records. The 16 family identifiers recur across clients and folds. A group without eligible pairs is undefined, and comparisons must use a common method-independent eligibility rule.

Clients are averaged within each fold/repeat; folds are then averaged within the repeat. Norm summaries use the corresponding three clients. Different norms use different interventions, so their levels do not measure intrinsic norm difficulty. Shared NDM uses one prediction per identical serialized input within a fold/repeat; local-NDM variants retain distinct client states. Figure 3 partitions outcomes into both correct, both block, both allow and reversed. Their fractions sum to 100% before rounding; only both correct contributes to Pair. Always allowing, always blocking or reversing the required decisions receives zero Pair. Thus, a raw decision flip alone is insufficient. Empirical probes must inherit their source families' held-out split and use fixed checkpoints and thresholds.

H.4 UPDATE-SCHEDULE CONTROLS

At common R, H, E, B, n_k, FCDC and Joint retain local NDMs and exchange only the CFE. Let $B_k = \min(B, n_k)$ and let S_θ be the CFE parameter payload in bytes. With identical data order and sampling seeds, both schedules consume $H + E$ batches per round. A Joint step updates both blocks on one batch and counts as one sample visit per example.

Table 11: Per-client schedule budgets for $H, E > 0$; payload counts exclude network overhead. Neither method performs final NDM fitting.

Quantity	FCDC	Joint ($H + E$ steps)
CFE updates	RE	$R(H + E)$
NDM updates	RH	$R(H + E)$
Sample visits	$R(H + E)B_k$	$R(H + E)B_k$
CFE upload / download, each	RS_θ	RS_θ

Joint therefore performs more updates to each parameter block despite equal sample visits and parameter payloads. Both use stateless SGD and the same configured learning rate. FCDC freezes the encoder in evaluation mode during NDM updates; CFE and Joint updates use training mode. The contrast also changes exposure to encoder stochasticity and does not isolate update order under equal computation. Figure 7 reports the separate-fold offline results, while Figure 8 retains all four live contrasts, including their zero-crossing intervals. Inference kTok does not measure training FLOPs, wall time or communication latency.