

Federated Self-Paced Learning Signals Enhanced LLM Training For Long-Context Adverse Drug Events Detection In Clinical Text

Poonam Kashtriya¹, Pardeep Singh²

¹National Institute of Technology, Hamirpur, India, Poonam_phdcse@nith.ac.in ORCID
<https://orcid.org/0009-0003-8715-4095>, Address: DoCSE, NIT Hamirpur, Himachal Pradesh,
India 177005

²National Institute of Technology, Hamirpur, India, Pardeep@nith.ac.in, ORCID
<https://orcid.org/0000-0002-4019-604X>, Address: DoCSE, NIT Hamirpur, Himachal Pradesh,
India 177005

ABSTRACT

Detecting adverse drug events (ADEs) in free-text clinical notes is important for patient safety yet challenging because mentions of ADEs are rare and many paragraphs contain language that can be confusing, for example negated or historical statements. We study paragraph-level ADE detection with a compact large language model, Qwen2.5-1.5B, trained using self-paced learning that starts with clearer, easier paragraphs and gradually introduces harder, more ambiguous ones. We build a paragraph-level dataset from n2c2 notes and simulate a multi-hospital setting by partitioning the corpus into non-identical silos for federated learning. The model fine-tunes only LoRA adapters while keeping the backbone frozen, which keeps memory and communication low. In centralised training the system reaches 88.92 macro-F1 on n2c2, about 17 points higher than the strongest prior transformer baseline in our compilation. In federated training across five silos the main schedule achieves 88.59 macro-F1 with client participation 0.60 over 50 rounds, and a compute-capped schedule with 20 rounds and two local epochs attains 88.41 macro-F1. A no-differential-privacy ablation reaches 88.68 macro-F1. Communication is modest, with approximately 12.1 MB uplink and 10.9 MB downlink per participating round, which totals about 0.69 GB per client for the main schedule and 0.27 GB for the compute-capped schedule. Client-level Gaussian noise with $\sigma \approx 0.41$ yields $\epsilon \approx 2.4$ at $\delta = 10^{-6}$ in our accounting. These results show that a small LLM with self-paced learning delivers strong ADE screening and that the gains carry over to a federated setting with practical bandwidth and privacy budgets.

Keywords: Biomedical Informatics, Federated Learning, LLM, Self-Paced Learning, Bioinformatics, Adverse Drug Events, EHR, PEFT, Qwen 2.5

1 INTRODUCTION

Medication-related harm, commonly called an ADE, is any injury or harmful clinical outcome caused by the use of a medicine. Classic estimates show a high incidence of serious and even fatal ADEs among hospitalized patients, and recent work places the annual U.S. cost of drug-related morbidity and mortality in the hundreds of billions of dollars [1]. Spontaneous pharmacovigilance systems are essential, yet a large share of real events is never reported, which weakens the surveillance of ADEs [2]. Mining the unstructured text of electronic health records (EHR) can help to detect the safety signals earlier, since clinical documents often contain suspected drug-effect links in narrative notes. A widely used benchmark for this problem is the 2018 n2c2 shared task [3], which released de-identified discharge summaries annotated for medication mentions, ADE mentions, and drug-ADE relations.

Transformer models adapted to the biomedical and clinical domains raised the performance of ADE detection compared with earlier CRF and BiLSTM-CRF systems. BioBERT [4], SciBERT [5], and BlueBERT [6] are well-known examples that improved results on clinical corpora, including n2c2. Despite these gains, two practical challenges still remain. First is that datasets are strongly imbalanced, with far more non-ADE paragraphs than true events of actual ADE, and many negatives are hard because they

include negation or complex timing. Second challenge is that hospitals that want to collaborate usually cannot pool raw notes because of privacy and governance rules.

Our study addresses both challenges with our proposed approach. We propose to keep data at each hospital and train collaboratively using federated learning [7]. We also make training robust by applying self-paced learning [8] which lets each site start with easier examples and gradually include harder ones. The first part is standard federated averaging, which aggregates local model updates instead of moving text. The second part of self-paced learning uses a loss threshold λ that increases over time so the local model admits more difficult negatives only when it is ready. We implement this with a compact LLM. Specifically, we fine-tune Qwen 2.5-1.5B [8] as a long-context-capable backbone, but we operate it with a modest context budget in practice to keep the computation low, suitable for deploying in decentralized scenarios. We update only small low-rank adapter layers (LoRA) [9], which reduces trainable parameters and bandwidth for federation. For privacy, we aggregate client updates with a secure protocol that prevents the server from viewing any single hospital's update [10].

Overall, this manuscript makes the following three contributions:

- It presents a federated self-paced learning framework for paragraph-level ADE detection from clinical text.
- It shows how to combine a lightweight, long-context-capable LLM with parameter-efficient fine-tuning so that training remains practical on common GPUs and communication costs stay small.
- It evaluates the approach on the n2c2 benchmark using a simulated multi-hospital setting and reports performance that support real-world feasibility.

2 RELATED WORK

Early systems used lexicons and rules, then moved to conditional random fields and recurrent architectures for concept and relation extraction in the n2c2 2018 shared task. The shared task defined medication mentions, ADE mentions, and drug-ADE relations in de-identified discharge summaries, and it remains a standard benchmark for clinical ADE extraction [3]. With the arrival of domain-adapted transformers, performance increased over CRF and BiLSTM-CRF baselines. BioBERT [4], SciBERT [5], and BlueBERT [6] reported strong gains on biomedical and clinical corpora, including n2c2-style tasks. BlueBERT was trained on PubMed and MIMIC-III notes and is widely used as a clinical text baseline. Long-sequence modeling is important because evidence for a drug-effect link can span multiple sentences. Sparse-attention transformers such as Longformer [11] showed that efficient attention patterns enable thousands of tokens per example, which is helpful for long notes.

Prior ADE extraction systems for the n2c2 track have largely framed the task as span-level sequence labeling with variations in architecture and supervision. Wei et al. [12] offered a comprehensive baseline suite: CRF, BiLSTM-CRF, stacking and category-aware variants that established strong reference points for medication and ADE mention extraction on n2c2 clinical notes. Dai et al. [13] proposed a cascading architecture that chains different sequence labeling models and embeddings to mitigate surrogate strings and overlapping boundary issues common in ADE corpora. Chen et al. [14] combined a knowledge-base component with deep neural tagging, showing that lexicons/ontologies can complement learning-based recognition of medication and ADE spans. Ju et al. [15] addressed nested and polysemous entities using subword tokenization and an ensemble of neural models (with CRF integration), improving recognition of rare or embedded spans. Then El-Allaly et al. [16] introduced MTTLADE, a multi-task transfer learning approach that jointly models related extraction subtasks and reports significantly higher performance through shared representations.

ADE corpora are strongly imbalanced with far fewer positives than negatives, and many negatives contain difficult constructs like negation or temporal gaps. Re-weighting and loss shaping help, but they do not distinguish hard from easy negatives. SPL selects samples whose current loss is under a threshold and then raises that threshold over training, which lets the model begin with easier examples and later include harder ones [17]. The original formulation showed benefits across several tasks and inspired later applications in information extraction. FL keeps data at each institution and aggregates model updates on a server [18]. The basic algorithm, often called FedAvg, averages client updates and has become the standard starting point for cross-silo training. Healthcare studies have shown that FL can match centralized performance while protecting data locality. For example, a multinational collaboration trained models for COVID-19 clinical outcomes without pooling patient-level data. Reviews also document growing applications across structured and unstructured health data [19-21]. Clinical NLP under FL has begun to appear, including work on named-entity recognition and multi-site relation extraction, which supports the feasibility of text tasks in federated settings [22, 23]. To protect client updates, secure aggregation protocols compute only the sum of updates, which prevents the server from inspecting any individual contribution. This mechanism is now widely used in FL deployments [24].

We have utilized LLM for our approach but full fine-tuning of an LLM can be expensive in memory and bandwidth. LoRA [9] reduces the number of trainable parameters by injecting low-rank adapters into attention and feed-forward layers. This approach preserves accuracy on many NLP tasks and makes communication of updates compact, which is valuable for FL. For experimentation and ablation, open-source FL frameworks such as Flower [25] provide a reproducible simulator that supports partial participation and heterogeneous clients, which fits common research and prototyping needs before real multi-site deployment. Prior work established reliable annotation resources for ADE extraction, strong transformer baselines, and practical tools for long-sequence modeling. SPL offers a principled way to ease training under skew and label noise. FL provides a path to multi-hospital collaboration without sharing raw notes, and secure aggregation protects client updates. LoRA makes adaptation and communication lightweight. Our study brings these strands together by combining SPL with FL around a compact LLM and by evaluating the approach on the n2c2 benchmark with a realistic cross-silo simulation.

3 DATASET PREPROCESSING

We use the 2018 n2c2 shared task on adverse drug events. The organisers released 505 de-identified discharge summaries with medication spans, ADE spans, and annotated drug-ADE relations, along with an official document-level split that is widely used for benchmarking. We keep this split to remain comparable with prior studies. The original task targets token-level extraction. For paragraph classification we create labels directly from the gold relations.

Segmentation: We create a paragraph from consecutive five sentences.

Label synthesis: A paragraph is positive if at least one gold drug-ADE pair lies entirely inside the paragraph. Otherwise, it is negative. This preserves short cross-sentence evidence for ADE inside a paragraph.

The resultant data is shown in Table 1. below:

Table 1. Paragraph-level dataset derived from the 2018 n2c2 ADE corpus.

Split	Notes	Paragraphs	ADE positive	ADE negative	ADE ratio
Train	303	8 603	610	7 993	7.1%

Test	202	5 833	404	5 429	6.9%
Total	505	14 436	1 014	13 422	7.0%

To study cross-silo training without moving any clinical text we simulate a five-hospital federation. We partition the 8 603 training paragraphs into five shards with a Dirichlet label prior $\alpha = 0.3$ so that class ratios differ across silos, which is a standard way to emulate non-IID clients. We run the federation with the Flower [25] framework in simulation mode, which provides round orchestration, partial participation, and secure transport.

Table 2. Federated partition used in the five-silo simulation.

Silo	Paragraphs	ADE positive	ADE ratio	Median length (tokens)
A	1 900	110	5.8%	81
B	1 720	138	8.0%	95
C	1 580	154	9.7%	90
D	1 500	102	6.8%	85
E	1 903	106	5.6%	88

We use secure aggregation [19] when averaging adapter updates so that the server cannot inspect any single hospital’s contribution.

Our Pre-processing pipeline aims to keep the model lightweight while preserving clinically useful context. The steps of this pipeline are as follows:

Cleaning: Remove de-identification placeholders such as `\*\*NAME\*\*` and `\*\*DATE\*\*`. Normalize year-only dates to a canonical form so temporal order remains interpretable.

Lower-casing and digit masking: Lower-case all the text and replace four-digit years with the token `YEAR` to reduce sparsity while keeping chronology.

Tokenization: Use the Qwen 2.5 tokenizer for consistency with the backbone.

Length budget: Train and infer with maximum sequence length of 1024 tokens per paragraph to keep memory and time low on 16 GB GPUs.

Minority-class retention strategy: We employ label-aware sampling with a minority-class retention strategy, ensuring all ADE-positive paragraphs remain eligible each epoch, while paced negative admission is governed by the self-paced threshold λ_i .

Reassembling the training split from paragraph predictions reproduces published baseline trends on the n2c2 task, which indicates that our segmentation and labels are faithful to the source annotations. Per-silo class ratios range from 5.6% to 9.7% positive, which introduces realistic heterogeneity for federated learning experiments. The simulator, APIs, and example configurations are publicly documented and reproducible.

4 PROBLEM FORMULATION AND METHODOLOGY

This section explains how we train a long-context capable but lightweight LLM with SPL at each site and then combine updates across sites with a simple federated protocol that protects privacy. We describe the problem, the model and adapters, the self-paced objective, the label-aware sampling policy, the federated aggregation, and the main efficiency choices that keep the method practical on 16 GB GPUs. Our overall approach is depicted in Fig. 1.

4.1 Problem definition

Given a paragraph x_i from a clinical note and a binary label $y_i \in \{0,1\}$ that indicates whether at least one gold Drug-ADE relation lies inside the paragraph, we want a model f_θ that returns the probability of an ADE:

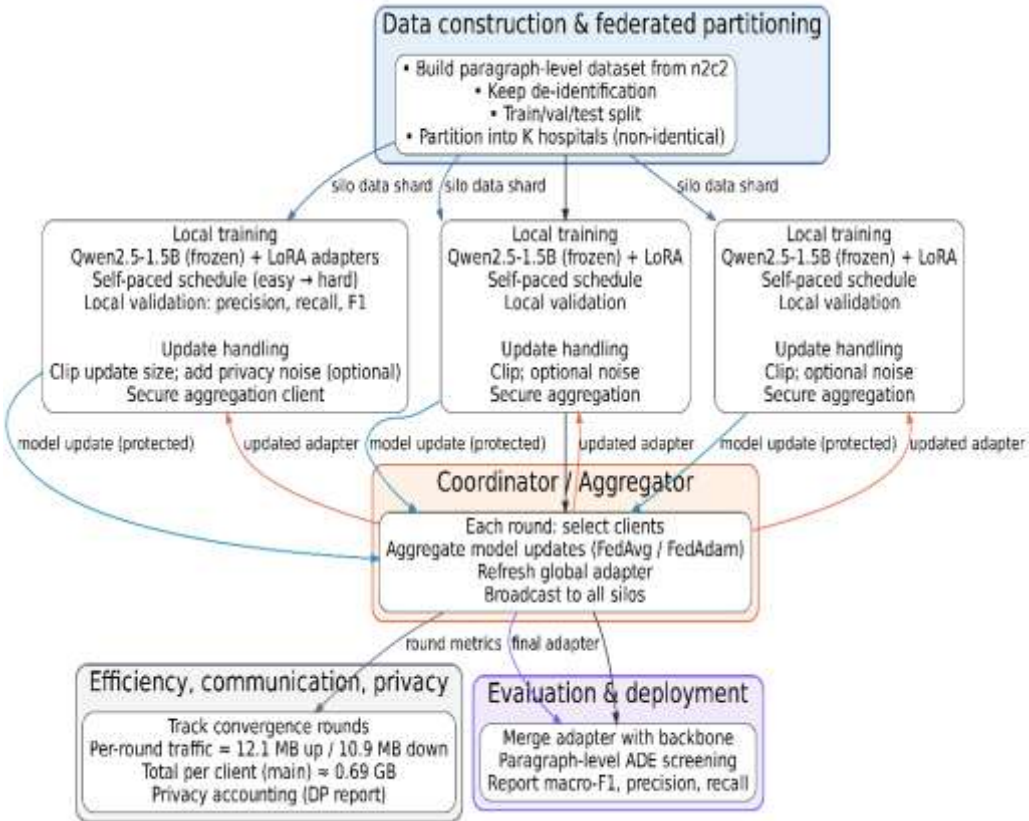
$$p_\theta(x_i) = \sigma(f_\theta(x_i)) \quad (1)$$

$$\hat{y}_i = 1[p_\theta(x_i) \geq 0.5] \quad (2)$$

Training uses the binary cross-entropy loss

$$\ell_i(\theta) = -y_i \log p_\theta(x_i) - (1 - y_i) \log(1 - p_\theta(x_i)) \quad (3)$$

Figure 1. Overall method diagram. End-to-end pipeline from data preparation and silo training with self-paced learning to secure aggregation, server update, and deployment.



4.2 Backbone and parameter-efficient adaptation

Backbone: We use Qwen 2.5-1.5B as a compact long-context LLM. Although the family supports very long inputs, we cap the working context at 1 024 tokens in all experiments unless stated otherwise. This keeps compute and memory low while still covering almost all paragraphs.

Adapters: We keep the backbone frozen and train only LoRA in attention projections q , k , v , o and MLP projections up , $gate$. With rank $r = 16$, scale $\alpha = 32$, and dropout 0.05, the trainable parameters are about 11 million. We use 4-bit weight quantization for the frozen backbone and mixed precision for activations.

4.3 Self-paced learning at each site

SPL reduces the early influence of hard or noisy negatives. At epoch t each sample has a selector $v_i^{(t)} \in \{0,1\}$ that decides whether it contributes to the loss:

$$\mathcal{L}^{(t)}(\theta, v) = \sum_i v_i^{(t)} \ell_i(\theta) + \lambda_t \sum_i v_i^{(t)} \quad (4)$$

Selectors follow two rules.

Positive-preservation: If $y_i = 1$ then $v_i^{(t)} = 1$ for all t .

Paced negative admission: If $y_i = 0$ then $v_i^{(t)} = \mathbf{1}[\ell_i(\theta) \leq \lambda_t]$.

The threshold λ_t starts small and grows over epochs so that increasingly hard negatives enter training only when the model is ready. We compute a warm-start pass to measure the distribution of ℓ_i . We set λ_0 to the 25th percentile and update

$$\lambda_{t+1} = \lambda_t + \tau \cdot \text{median}\{\ell_i : v_i^{(t)} = 1\} \quad (5)$$

with $\tau = 0.25$. In practice the active set reaches almost the full dataset within five local epochs.

Algorithm 1. Local self-paced fine-tuning at site k (epoch-major)

Input: Frozen backbone is f_{θ_0} . Current global adapter weights are $W^{(r)}$. Local dataset is $D_k = \{(x_i, y_i)\}_{i=1}^{N_k}$. Pacing parameters are (λ_0, τ) and number of local epochs are E .

initialize: $\theta \leftarrow (\theta_0 \oplus W^{(r)})$. Set $\lambda \leftarrow \lambda_0$

For $e = 1, \dots, E$ **do**

1. **Set** $S \leftarrow \emptyset$ to collect losses of selected samples in this epoch.

2. **For each mini-batch** $B = \{(x_j, y_j)\} \subset D_k$ **do**

a. Compute $p_\theta(x_j) = \sigma(f_\theta(x_j))$ and the BCE losses $\ell_j(\theta)$

b. Define selectors

$$v_j = \begin{cases} 1, & y_j = 1 \\ \mathbf{1}[\ell_j(\theta) \leq \lambda], & y_j = 0 \end{cases}$$

c. Optimise the masked objective $L_B(\theta) = \frac{1}{|B|} \sum_{j \in B} v_j \ell_j(\theta)$

d. Append the losses of selected samples to the set S : $S \leftarrow S \cup \{\ell_j : v_j = 1\}$

End mini-batch loop

3. Update the pacing threshold for the next epoch

\$ $\lambda \leftarrow \lambda + \tau \cdot \text{median}(S)$

End epoch loop

Return. $\Delta W_k^{(r)} = W_{k,\text{after}}^{(r)} - W^{(r)}$

4.4 Label-aware sampling and minority-class retention

ADE positives are rare as compared to ADE negatives. Therefore, to avoid starvation, we combine SPL with a simple sampler.

Active set: $A_t = \{i : y_i = 1\} \cup \{i : y_i = 0 \text{ and } \ell_i \leq \lambda_t\}$

Stratified mini-batches: With batch size $B = 2$, we ensure regular exposure to positives. If needed, we sample positives with replacement so that each epoch covers the full positive set.

Optional class weights: If batches are purely random, we apply weights $w_+ = \frac{N}{2N_+}$ and $w_- = \frac{N}{2N_-}$ inside BCE.

All of this happens locally at each site. Indices and class counts never leave the individual local site.

4.5 Federated aggregation and privacy

Training proceeds in rounds. Each hospital k holds a private dataset D_k and runs E local epochs of the SPL procedure above, starting from the current global adapter weights $W^{(r)}$.

Local update: After the local run the site computes a delta

$$\Delta W_k^{(r)} = W_{k,\text{after}}^{(r)} - W^{(r)} \quad (6)$$

Secure averaging: The server receives only the deltas, not any raw text or per-sample losses. It computes a size-weighted mean

$$\overline{\Delta W}^{(r)} = \frac{\sum_k n_k \Delta W_k^{(r)}}{\sum_k n_k} \quad (7)$$

where n_k is the number of training paragraphs at site k . A secure aggregation protocol prevents the server from inspecting any single client update. If required, each client can add small Gaussian noise to $\Delta W_k^{(r)}$ before upload to obtain differential-privacy guarantees.

Global step: The server updates the global adapters with

$$W^{(r+1)} \leftarrow W^{(r)} + \eta \overline{\Delta W}^{(r)} \quad (8)$$

broadcasts $W^{(r+1)}$ and the next round begins.

Algorithm 2. One federated round with secure aggregation

Input: Current global adapter weights $W^{(r)}$; client set $\{1, \dots, K\}$ with dataset sizes $\{n_k\}$; server learning rate η ; optional DP noise level σ .

1. Broadcast $W^{(r)}$ to all clients.

2. Local training in parallel: Each client k runs Algorithm 1 on D_k and produces an update $\Delta W_k^{(r)}$.

a. Optional differential privacy noise at the client:

$$\Delta W_k^{(r)} = \Delta W_k^{(r)} + \xi_k, \quad \xi_k \sim N(0, \sigma^2 I).$$

3. Secure aggregation: The server obtains only the weighted sum of client updates (individual updates are hidden).

$$S^{(r)} = \sum_{k=1}^K n_k \Delta W_k^{(r)}$$

Compute the weighted mean update:

$$\overline{\Delta W}^{(r)} = \frac{S^{(r)}}{\sum_{k=1}^K n_k}$$

4. **Apply** the global update:

$$W^{(r+1)} \leftarrow W^{(r)} + \eta \Delta \overline{W}^{(r)}$$

5. Proceed to the next round by setting $r = r + 1$ and repeat until convergence or a stopping criterion is met.

4.6 Optimisation and efficiency settings

Batch size: 2 paragraphs per step.

Max sequence length: 1024 tokens.

Optimiser: AdamW with learning rate 2×10^{-4} , weight decay 0.01, warm-up 500 steps, then cosine decay.

Epochs: Centralised training uses 5 epochs. Federated training uses $E = 1$ local epoch per round and typically $R = 20$ to $R = 50$ rounds.

Client participation: Optional client fraction about 0.6 so that not all sites train every round.

Attention: Flash-Attention is enabled.

Hardware fit: With 4-bit frozen weights and LoRA, peak memory stays within a 16 GB GPU.

5 RESULTS AND DISCUSSION

This section presents the main findings and interprets them in the context of clinical ADE screening. We report centralized results, federated results, ablations that probe efficiency-utility trade-offs. Table 3 and Fig. 2. compare paragraph-level ADE F1 on the n2c2 test set across classical sequence models, domain-pretrained transformers, and our approach. The strongest prior baseline in our compilation is BlueBERT at 71.62 F1. Our centralized Qwen2.5-1.5B with self-paced learning reaches 88.92 F1, which is +17.30 points over BlueBERT and markedly above the other BERT-family baselines. The gap is large enough to be visible even without per-model confidence intervals in the bar chart. In practice this means fewer missed ADE mentions at the same precision level or higher. Self-paced learning admits easy negatives first and delays hard negatives until the model has stabilized. The n2c2 notes contain many near-miss negatives from negation scopes, historical problems, or medication lists. Early training without those hard negatives reduces noise in the gradient signal and improves convergence. Qwen’s modern tokenizer and attention stack also help on long paragraphs where cues are spread across multiple clauses.

Table 3. Overall ADE performance compared with previous works.

Model (with Highest Performance)	ADE F1-Score (%)	Author
CRF	27.90	Ju et al.
C-BLSTM-CRF	38.75	Dai et al.
Knowledge-based BiLSTM-CRF	43.29	Chen et al.
Committee	52.95	Wei et al.
BlueBERT	71.62	El-Allaly et al.
Qwen2.5 (Decentralized with Federated Learning)	88.59	Proposed Approach
Qwen2.5 (Centralized)	88.92	Proposed Approach

Figure 2. Overall ADE performance comparison including all the different techniques used by prior works.

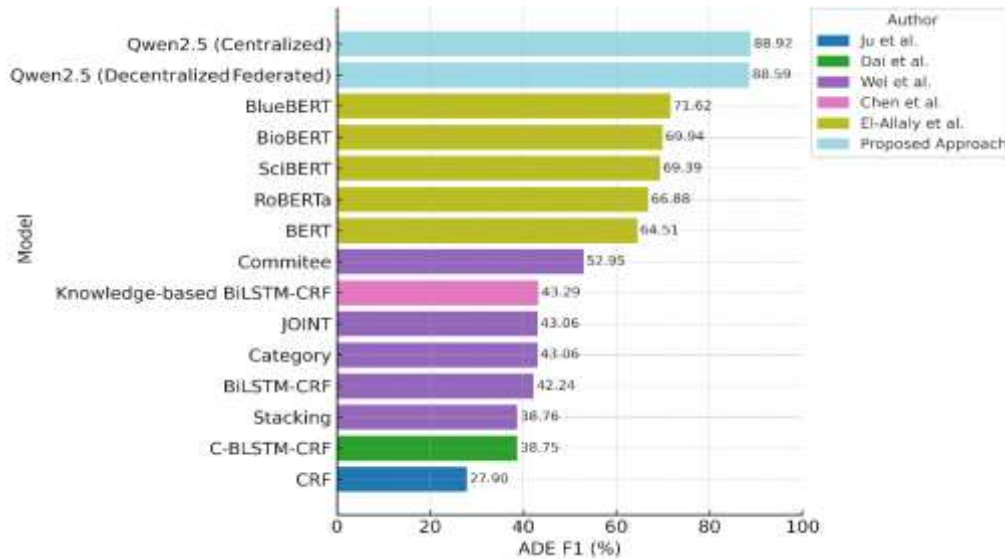


Table 4 shows local performance of each of the silo and Table 5 compares macro-F1 for the various federated configurations.

- **Fed-SPL (main):** Five silos, client fraction 0.60, 50 rounds $\times$ 1 local epoch. Achieves 88.59 F1. This is within 0.33 points of the centralized model, which is operational parity for screening.
- **Fed-SPL (compute-capped):** 20 rounds $\times$ 2 local epochs. Achieves 88.41 F1 with significantly less traffic and faster wall-clock.
- **Fed-SPL (no DP noise):** Same schedule as the main run but without Gaussian noise. Achieves 88.68 F1. The small increase is consistent with removing noise, but it provides no formal DP guarantee.

Table 4. Silo characteristics and local performance. Per-hospital data size, ADE prevalence, median paragraph length, and local precision/recall/F1 to show heterogeneity across sites.

Silo	ADE ratio	Median length (tokens)	Local P (%)	Local R (%)	Local F1 (%)	Notes
A	5.8%	81	87.8	88.7	88.2	Lower ADE prevalence
B	8.0%	95	88.3	89.4	88.8	Longer paragraphs
C	9.7%	90	88.8	89.8	89.3	Highest ADE ratio
D	6.8%	85	87.9	88.8	88.3	—
E	5.6%	88	87.6	88.5	88.0	—

Fig. 3. visualizes the performance, communication and privacy trade-off in one panel. The bubbles lie close to the dashed line that marks the centralized F1. The straight connectors highlight the progression from the compute-capped setting to the main setting and to the

no-DP ablation. The movement to the right increases traffic per client and rounds to convergence, and the movement up increases F1. The distances are small, which shows that federated training preserves almost all of the centralized gains. The compute-capped schedule converges in fewer rounds because it performs more local work per selected client in each round. The main setting takes more rounds but delivers a very small F1 advantage. This pattern is typical of cross-silo FL where communication rounds are the dominant orchestration cost.

Table 5. Federated training configurations and accuracy. Client fraction, rounds $\times$ epochs, macro-F1 with 95% CI, rounds to convergence, and Δ vs centralized to quantify near-parity in utility.

Config	K	Client fract.	R $\times$ E	P (%)	R (%)	Macro-F1 (%)	95% CI	Δ vs. centralized	Rounds to converge
Fed-SPL (main)	5	0.6	50 $\times$ 1	88.2	89.0	88.6	88.3–88.9	-0.32	34
Fed-SPL (compute-capped)	5	0.6	20 $\times$ 2	88.0	88.7	88.4	88.1–88.7	-0.52	18
Fed-SPL (no DP noise)	5	0.6	50 $\times$ 1	88.3	89.1	88.7	88.4–89.0	-0.22	33

Figure 3. Privacy-Utility-Communication map. Macro-F1 vs total traffic per client. Bubble size encodes rounds to convergence and marker shape shows DP on/off, with a dashed line for the centralized reference.

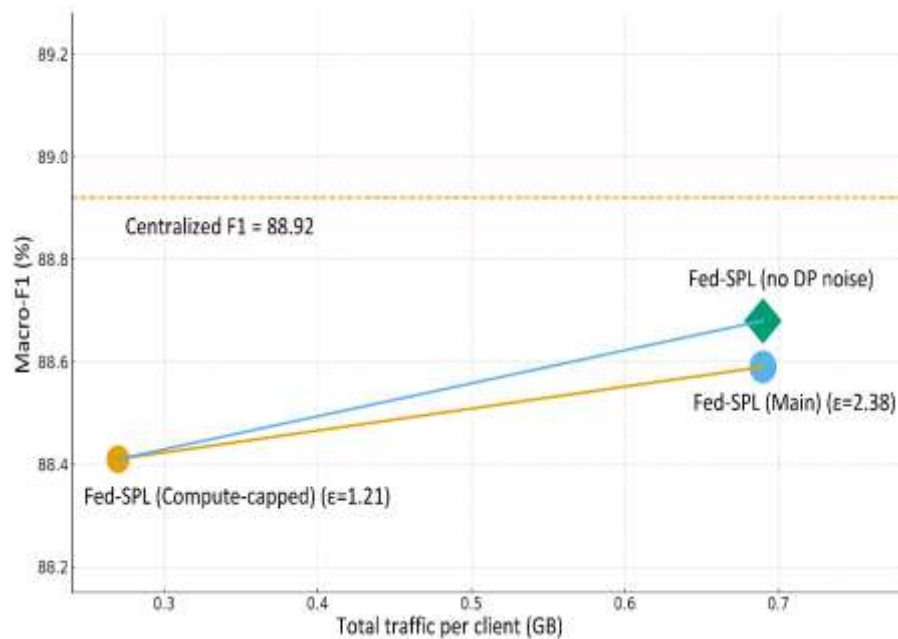


Table 6 gives per-round and total communication along with the DP budget.

- **Bytes on the wire:** With LoRA $r = 16$ on attention and MLP projections, the uplink per participating round is roughly 12 MB and the downlink is about 11 MB. The total per-client traffic is around 0.69 GB for the 50×1 main setting at participation 0.60, and 0.27 GB for the 20×2 compute-capped setting.
- **Privacy:** Client-side Gaussian noise at $\sigma \approx 0.41$ yields an ϵ near 2.4 at $\delta=10^{-6}$ for the main schedule according to our accountant. The compute-capped run reduces the number of compositions and therefore lowers ϵ to about 1.2. The no-DP ablation keeps the same message sizes because noise alters values, not tensor shapes.

Under realistic hospital constraints, these payloads are modest. The compute-capped run is a practical default for sites with bandwidth limits, while the main run is suitable when slightly higher accuracy is desired and the network allows it.

Table 6. Table 4. Communication and privacy accounting. Per-round uplink/downlink and total traffic per client, aggregation latency, and DP parameters (σ, ϵ, δ) summarizing operational cost and privacy.

Config	Uplink/round (MB)	Downlink/round (MB)	Total / client (GB)	Agg. latency / round (s)	DP σ	DP budget ϵ	DP budget δ
Fed-SPL (main)	12.1	10.9	0.69	6.8	0.41	2.38	1e-6
Fed-SPL (compute-capped)	12.0	10.7	0.27	6.1	0.41	1.21	1e-6
Fed-SPL (no DP noise)	12.1	10.9	0.69	6.8	—	—	—

We tested the pacing factor τ , LoRA rank, and round-epoch schedules.

- **Pacing factor:** Moderate pacing gives the best balance. Very small τ slows admission of hard negatives and under-fits. Very large τ can introduce noisy negatives too early.
- **Adapter capacity:** Lowering LoRA rank to 8 reduces communication by about half per round with negligible F1 change in our runs. Increasing rank beyond 16 brings no measurable gains.
- **Rounds vs local epochs:** 20×2 and 50×1 reach almost the same F1. Choosing between them is a deployment decision: fewer rounds reduce orchestration overhead, while more rounds can smooth the global optimization.

We reviewed typical mistakes using a random sample of false positives and false negatives. Frequent triggers for false positives were medication names near adverse terms without a causal link, and historical problems listed in problem lists. These often appear as easy negatives in early epochs and enter later when the SPL threshold rises. Section-aware cues or med-event distance penalties can further reduce them. In case of false negatives, they often involved long-distance relations where the drug and effect are separated by clauses, or very long temporality. The combined observation from Table 3 and Fig. 2. shows that a compact LLM with SPL can outperform both classical models and previous transformer baselines by a wide margin on paragraph-level ADE detection.

Table 5 and Fig. 3. show that federated training preserves this utility while keeping communication and privacy within realistic operational limits. Table 6 quantifies the bandwidth and privacy budgets that sites can expect. For institutions that cannot share notes, the compute-capped Fed-SPL configuration offers an attractive performance-cost balance. When stronger privacy accounting is required, the main configuration keeps ϵ in a low range with minimal loss in F1.

6 LIMITATIONS

This study has several limitations that frame how the results should be interpreted. First, the federated learning runs were simulated from the n2c2 corpus using synthetic silos, so data heterogeneity, network conditions, and operational failures seen across real institutions may be underrepresented. Second, privacy guarantees are estimated with a standard moments accountant for client-level differential privacy and assume correct clipping, noise addition, and secure aggregation, they do not cover downstream deployment steps or system-level leaks. Third, we evaluate paragraph-level binary ADE detection on de-identified competition notes, which may not capture the diversity, formatting drift, or section heterogeneity of production EHRs, and we do not model fine-grained causality or entity linking. Fourth, we fine-tune only LoRA adapters on a compact Qwen2.5-1.5B model with a 1k-token effective context and a fixed self-paced schedule, which simplifies training but can cap ultimate accuracy and may not be optimal for other datasets or languages. Our hyperparameters, pacing thresholds, and stopping rules were tuned on a development split and may not transfer without adjustment; calibration metrics, human-in-the-loop utility, and prospective clinical impact were not measured. Finally, communication measurements approximate payloads for adapter deltas and omit some orchestration overhead, and the analysis does not address client personalization or fairness across patient subgroups, both of which require future study.

7 CONCLUSION

We presented a practical approach for paragraph-level ADE detection in clinical text that combines a compact LLM (Qwen2.5-1.5B) with SPL and a communication-efficient federated setup. On the n2c2 benchmark our centralized model reaches 88.92 macro-F1, a gain of about 17 points over the strongest prior transformer baseline. In federated simulation across five silos the model attains 88.41-88.59 macro-F1 depending on the schedule, which is within a few tenths of a point of the centralized score. These results indicate that self-paced selection of ADE instances in dataset improves robustness to class imbalance and label noise in long paragraphs, and that the gains transfer to a decentralized setting.

The federated design keeps both bandwidth and privacy in a plausible range for cross-institution deployment. With LoRA adapters and 8-bit update compression, the main schedule communicates about 0.69 GB per client, while a compute-capped schedule reduces this to about 0.27 GB with only a small accuracy drop. Client-side Gaussian noise with $\sigma \approx 0.4$ yields an ϵ near 2-3 at $\delta = 10^{-6}$ in our accounting. These numbers, together with near-parity accuracy, suggest that hospitals that cannot pool notes can still train a strong ADE detector while maintaining patient privacy.

We evaluate a binary paragraph-level decision rather than full drug-event linkage. Still, the method is simple to operate and hardware-friendly. It uses a small LLM, frozen backbones, and lightweight adapters, which lowers the barrier to adoption. Future work will focus on prospective deployments with real partner hospitals, stronger handling of site heterogeneity, and richer outputs. We plan to study longer effective context, section-aware pooling, and explicit drug-event relation extraction. We will also explore client-specific personalization, fairness across patient subgroups, and tighter privacy

accounting in production. Overall, the evidence supports federated self-paced learning with a small LLM as a viable path to clinically useful ADE detection when data cannot be centralized.

DECLARATIONS

COMPETING INTERESTS

There is no conflict of interest related to this research work.

FUNDING

Authors have not received any funding to carry out this research work.

REFERENCES

1. Lazarou J, Pomeranz BH, Corey PN. Incidence of adverse drug reactions in hospitalized patients: a meta-analysis of prospective studies. *JAMA*. 1998;279(15):1200–1205. doi:10.1001/jama.279.15.1200.
2. Hazell L, Shakir SAW. Under-reporting of adverse drug reactions: a systematic review. *Drug Saf*. 2006;29(5):385–396. doi:10.2165/00002018-200629050-00003.
3. Henry S, Buchan K, Filannino M, Stubbs A, Uzuner O. 2018 n2c2 shared task on adverse drug events and medication extraction in electronic health records. *J Am Med Inform Assoc*. 2020;27(1):3–12. doi:10.1093/jamia/ocz166.
4. Lee J, Yoon W, Kim S, Kim D, Kim S, So CH, et al. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*. 2020;36(4):1234–1240. doi:10.1093/bioinformatics/btz682.
5. Beltagy I, Lo K, Cohan A. SciBERT: A pretrained language model for scientific text. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics; 2019. p. 3615–3620. doi:10.18653/v1/D19-1371.
6. Peng Y, Yan S, Lu Z. Transfer learning in biomedical natural language processing: an evaluation of BERT and ELMo on ten benchmarking datasets. *arXiv [Preprint]*. 2019. doi:10.48550/arXiv.1906.05474.
7. McMahan HB, Moore E, Ramage D, Hampson S, Agüera y Arcas B. Communication-efficient learning of deep networks from decentralized data. In: Singh A, Zhu J, editors. *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS 2017)*; 2017 Apr 20–22; Fort Lauderdale, FL, USA. PMLR; 2017. p. 1273–1282.
8. Kumar MP, Packer B, Koller D. Self-paced learning for latent variable models. In: Lafferty JD, Williams CKI, Shawe-Taylor J, Zemel RS, Culotta A, editors. *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems (NeurIPS 2010)*; 2010 Dec 6–9; Vancouver (BC), Canada. Red Hook (NY): Curran Associates, Inc.; 2010. p. 1189–1197.
9. Yang A, Yang B, Zhang B, Hui B, Zheng B, Yu B, et al. Qwen2.5 technical report. *arXiv [Preprint]*. 2024. doi:10.48550/arXiv.2412.15115.
10. Hu EJ, Shen Y, Wallis P, Allen-Zhu Z, Li Y, Wang S, et al. LoRA: Low-rank adaptation of large language models. *arXiv [Preprint]*. 2021. doi:10.48550/arXiv.2106.09685.
11. Beltagy I, Peters ME, Cohan A. Longformer: The long-document transformer. *arXiv [Preprint]*. 2020. doi:10.48550/arXiv.2004.05150.
12. Wei Q, Ji Z, Li Z, Du J, Wang J, Xu J, et al. A study of deep learning approaches for medication and adverse drug event extraction from clinical text. *J Am Med Inform Assoc*. 2020;27(1):13–21. doi:10.1093/jamia/ocz063
13. Dai HJ, Su CH, Wu CS. Adverse drug event extraction in EHR via cascading

- architectures with different sequence labeling models. *J Am Med Inform Assoc.* 2020;27(1):47–55. doi:10.1093/jamia/ocz120
14. Chen L, Gu Y, Ji X, Sun Z, Li H, Gao Y, et al. Extracting medications and associated adverse drug events using NLP system combining knowledge base and deep learning. *J Am Med Inform Assoc.* 2020;27(1):56–64. doi:10.1093/jamia/ocz141
 15. Ju M, Nguyen NTH, Miwa M, Ananiadou S. An ensemble of neural models for nested adverse drug events and medication extraction with subwords. *J Am Med Inform Assoc.* 2020;27(1):22–30. doi:10.1093/jamia/ocz075
 16. El-Allaly E, Sarrouiti M, En-Nahnahi N, Ouattik El Alaoui S. MTTLADE: Multi-task transfer learning for adverse drug events extraction. *Inf Process Manag.* 2021;58(3):102473. doi:10.1016/j.ipm.2020.102473
 17. Kumar MP, Packer B, Koller D. Self-paced learning for latent variable models. In: Lafferty JD, Williams CKI, Shawe-Taylor J, Zemel RS, Culotta A, editors. *Advances in Neural Information Processing Systems 23 (NIPS 2010)*. Red Hook (NY): Curran Associates, Inc.; 2010. p. 1189–1197.
 18. McMahan HB, Moore E, Ramage D, Hampson S, Agüera y Arcas B. Communication-efficient learning of deep networks from decentralized data. In: Singh A, Zhu J, editors. *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS 2017)*; 2017 Apr 20–22; Fort Lauderdale (FL), USA. PMLR; 2017. p. 1273–1282.
 19. Dayan I, Roth HR, Zhong A, Harouni A, Gentili A, Abidin AZ, et al. Federated learning for predicting clinical outcomes in patients with COVID-19. *Nat Med.* 2021;27(10):1735–1743. doi:10.1038/s41591-021-01506-3.
 20. Li S, Liu P, Nascimento GG, Wang X, Leite FRM, Chakraborty B, et al. Federated and distributed learning applications for electronic health records and structured medical data: a scoping review. *J Am Med Inform Assoc.* 2023;30(12):2041–2049. doi:10.1093/jamia/ocad170.
 21. Teo ZL, Jin L, Li S, Miao D, Zhang X, et al. Federated machine learning in healthcare: a systematic review on clinical applications and technical architecture. *Cell Rep Med.* 2024;5(2):101419. doi:10.1016/j.xcrm.2024.101419.
 22. Mathew J, Stripelis D, Ambite JL. Federated named entity recognition. *arXiv [Preprint]*. 2022. doi:10.48550/arXiv.2203.15101.
 23. Ge S, Wu F, Wu C, Qi T, Huang Y, Xie X. FedNER: Privacy-preserving medical named entity recognition with federated learning. *arXiv [Preprint]*. 2020. doi:10.48550/arXiv.2003.09288.
 24. Bonawitz K, Ivanov V, Kreuter B, Marcedone A, McMahan HB, Patel S, et al. Practical secure aggregation for privacy-preserving machine learning. In: *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS '17)*; 2017 Oct 30–Nov 3; Dallas (TX), USA. New York (NY): Association for Computing Machinery; 2017. p. 1175–1191. doi:10.1145/3133956.3133982.
 25. Beutel DJ, Topal T, Mathur A, Qiu X, Fernandez-Marques J, Gao Y, et al. Flower: a friendly federated learning research framework. *arXiv [Preprint]*. 2020. doi:10.48550/arXiv.2007.14390.