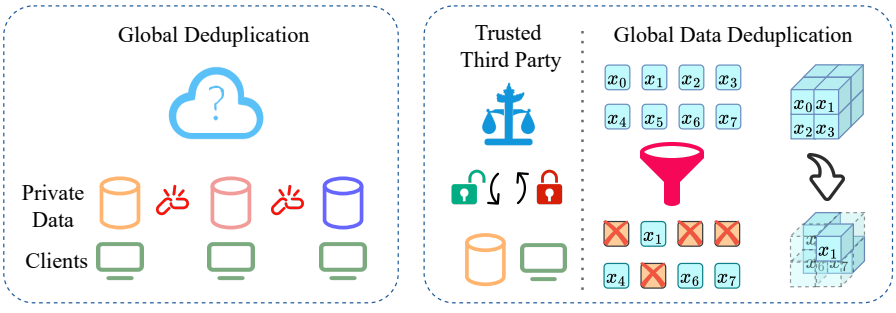
FedRW: Efficient Privacy-Preserving Data Reweighting for Enhancing Federated Learning of Language Models

Pukang Ye[*◇] **Junwei Luo**[♣]
Jiachen Shen[◇] **Saipan Zhou**[◇] **Shangmin Dou**[△] **Zhenfu Cao**[◇] **Hanzhe Yao**[◇]
Xiaolei Dong[†◇] **Yunbo Yang**[†◇▼■]
[◇]East China Normal University [♣]Wuhan University [▼]Zhejiang University [△]ZStack
[■]Hangzhou High-Tech Zone (Binjiang) Institute of Blockchain and Data Security

Abstract

Data duplication within large-scale corpora often impedes large language models' (LLMs) performance and privacy. In privacy-concerned federated learning scenarios, conventional deduplication methods typically rely on trusted third parties to perform uniform deletion, risking loss of informative samples while introducing privacy vulnerabilities. To address these gaps, we propose Federated ReWeighting (FedRW), the first privacy-preserving framework, to the best of our knowledge, that performs soft deduplication via sample reweighting instead of deletion in federated LLM training, without assuming a trusted third party. At its core, FedRW proposes a secure, frequency-aware reweighting protocol through secure multi-party computation, coupled with a parallel orchestration strategy to ensure efficiency and scalability. During training, FedRW utilizes an adaptive reweighting mechanism with global sample frequencies to adjust individual loss contributions, effectively improving generalization and robustness. Empirical results demonstrate that FedRW outperforms the state-of-the-art method by achieving up to 28.78× speedup in preprocessing and approximately 11.42% improvement in perplexity, while offering enhanced security guarantees. FedRW thus establishes a new paradigm for managing duplication in federated LLM training.



(a) Deduplication challenges in FL **(b) Hard deduplication solution**

Figure 1: Deduplication in Federated Learning (FL). (a) Challenges of global deduplication in decentralized settings: privacy constraints prohibit direct data sharing. (b) State-of-the-art solution utilizing hard deduplication over encrypted data, requiring a trusted third party.

[*]✉ Main Contact: 51275902028@stu.ecnu.edu.cn
[†]Correspondence authors.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

1 Introduction

Large language models (LLMs) [1–5] have driven remarkable progress across a wide range of applications [6–9]. However, their performance fundamentally depends on data quality, yet real-world corpora often suffer from noise, bias, and especially redundancy. Among these issues, duplicated sequences are particularly widespread in large text datasets [10, 11], weakening generalization and encouraging memorization. This not only hinders downstream performance but also increases vulnerability to privacy attacks such as model inversion, prompt injection, and membership inference [12–15]. As a result, data deduplication has become a standard preprocessing step in training pipelines. Existing techniques fall into two categories: hard deduplication, which removes duplicates via exact or fuzzy matching (e.g., suffix arrays, MinHash) [16, 17]; and soft deduplication, which reweights samples to preserve dataset integrity and avoid brittle thresholding [18–22].

Meanwhile, the growing scarcity of high-quality public data and rising concerns over data privacy [23] have brought federated learning (FL) [24] to the forefront as a compelling alternative for LLM training. By enabling collaborative learning across decentralized clients without local data sharing, FL naturally supports privacy preservation and improved utilization of high-value private data. Yet, FL introduces unique challenges for deduplication, presented in Figure 1(a). Unlike centralized settings, global redundancy across clients cannot be directly resolved due to privacy constraints. A fundamental dilemma emerges: local deduplication fails to detect inter-client duplicates, while global mechanisms cannot bypass privacy silos, leaving redundancy unresolved in federated settings.

Abadi et al. [25]'s EP-MPD represents the most state-of-the-art work for federated hard deduplication, a robust cryptographic framework built on group private set intersection [26], as illustrated in Figure 1(b). Nonetheless, key challenges remain unresolved: (1) strict removal of samples may discard informative or domain-specific content beneficial to model training; (2) multi-round key agreement and encryption introduce significant computational and communication overhead; and (3) reliance on a trusted third party for both encryption and duplicate counting reduces feasibility in stricter privacy settings.

To address the issues mentioned above, we propose Federated ReWeighting (FedRW), to the best of our knowledge, the first framework that enables privacy-preserving soft deduplication in federated LLM training without relying on any trusted third party. Unlike state-of-the-art method that discards duplicated samples, FedRW pioneers a new paradigm of secure, frequency-aware sample reweighting, enabling fine-grained control over sample redundancy while ensuring strict privacy guarantees. At the core of FedRW lies a novel protocol, Privacy-Preserving Multi-Party Reweighting (PPMPR), which securely identifies global duplication patterns across clients through a series of lightweight, third-party-free two-party interactions. To ensure scalability, we further introduce a parallel orchestration strategy that organizes the pairwise interactions into a hierarchical schedule, significantly reducing protocol complexity. Comprehensive experiments demonstrate that FedRW improves both preprocessing efficiency and model generalization, particularly in data-scarce and resource-constrained federated settings. In summary, the key contributions are:

- **FedRW Framework.** Duplicate or overly frequent samples in federated LLM training lead to inefficiency and privacy leakage, especially when deletion-based solutions are impractical. To the best of our knowledge, we propose FedRW, the first framework to achieve privacy-preserving soft deduplication in federated LLM training. Unlike hard deletion methods, FedRW introduces secure, frequency-aware sample reweighting, establishing a new paradigm that bridges privacy protection and data-centric optimization.
- **PPMPR Protocol.** We design PPMPR, a secure protocol for global frequency estimation without relying on a trusted third party. To scale to practical settings, we further introduce a parallel orchestration strategy that reduces the total protocol complexity from $O(n^2)$ to $O(2^{\lceil \log_2 n \rceil})$, achieving 17.61-$28.78\times$ acceleration on large datasets and 4.09-$28.78\times$ speedup in preprocessing when scaled to 50 parties.
- **Experimental Evaluation.** We conduct extensive empirical studies across diverse datasets and model configurations. By adaptive reweighting, FedRW yields approximately 11.42% perplexity reduction over the baseline, with particularly enhanced robustness under data-scarce and resource-constrained federated settings, where hard deduplication methods often exhibit apparent limitations.

2 Related Work

This section reviews data deduplication, categorizing centralized and distributed approaches. We emphasize the limitations in distributed settings, which motivates our proposed FedRW framework.

Centralized Deduplication. Centralized deduplication is crucial for large text corpora, which often contain substantial exact or near-exact samples [10, 11] that degrade model performance and compromise privacy [11, 13–15, 25]. Techniques for exact matching commonly include suffix arrays [16, 27], while fuzzy matching typically employs MinHash for syntactic similarity [11, 16, 17]. Semantic duplication can be identified using pretrained reference models [20, 22, 28].

Instead of removing duplicates, soft deduplication methods reweight training data to mitigate redundancy while preserving the integrity and valuable diversity of datasets. For instance, RedPajama-Data-v2 [29] leverages over 40 quality metrics for systematic filtering and reweighting. DoReMi [20] derives domain-specific weights estimated by a proxy model. Methods like SoftDedup [22] and DSIR [21] quantify sample commonness or importance via n-grams. DrICL [30] uses differentiated learning and cumulative advantages for dynamic reweighting. RHO-1 [31] employs token-level scoring with Selective Language Modeling. However, these centralized strategies are not directly applicable to privacy-concerned FL environments, which effectively leverage high-quality private data.

Distributed Deduplication. Deduplication in FL faces unique challenges due to privacy constraints and data silos. Existing work DupLESS [32] proposes encrypted deduplication using a dual-server architecture, one for encryption key derivation and one for ciphertext deduplication. The state-of-the-art, EP-MPD [25], introduces a group private set intersection framework built on symmetric-key encryption [26] and oblivious pseudorandom functions [33], but still relies on a trusted third party. Critically, these methods focus solely on hard deduplication, neglecting the benefits of reweighting strategies that better preserve data utility and potentially enhance model performance.

These limitations highlight the need for a decentralized soft deduplication solution that ensures privacy without relying on trusted third parties. To this end, we propose an efficient, secure, and third-party-free reweighting framework for federated LLM training, delivering enhanced scalability, performance, and robustness while also ensuring stronger privacy guarantees.

3 Preliminaries

Causal Language Models. Causal language models are autoregressive architectures that estimate the joint probability of a token sequence by expressing it into a chain of conditional probabilities:

$$P(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} P(x_i \mid x_{<i}), \tag{1}$$

where $P(x_i \mid x_{<i})$ is probability of token x_i given its historical context $x_{<i}$. Model training minimizes the cross-entropy loss to maximize the likelihood of contextually consistent sequences:

$$\mathcal{L} = -\frac{1}{n} \sum_{i=1}^{n} \log P\left(x_i \mid x_{<i}; \theta\right), \tag{2}$$

where n is the sequence length and θ the model parameters. Perplexity is the standard evaluation metric, calculated as the exponentiated average negative log-likelihood over the sequence:

$$\text{Perplexity} = \exp\left(-\frac{1}{n} \sum_{i=1}^{n} \log P(x_i \mid x_{<i})\right). \tag{3}$$

Lower perplexity signifies reduced prediction uncertainty and better data distribution alignment.

Security Definition. In cryptographic protocol design, the ideal functionality f models the desired behavior of a protocol in an idealized setting. It serves as a trusted third party that collects inputs from all parties, performs the computation securely, and returns the outputs. A protocol is considered secure if its real-world execution is computationally indistinguishable from the ideal execution with f. Due to space constraints, formal definitions are deferred to Appendix A.

4 Framework

This section details the design and implementation of FedRW. We start by formalizing the PPMPR protocol, followed by a practical construction using cryptographic primitives and a parallel orchestration acceleration strategy for efficiency and scalability. Finally, we describe the integration of the derived weights into the FL training pipeline. An overview of key stages of the FedRW framework is illustrated in Figure 2.

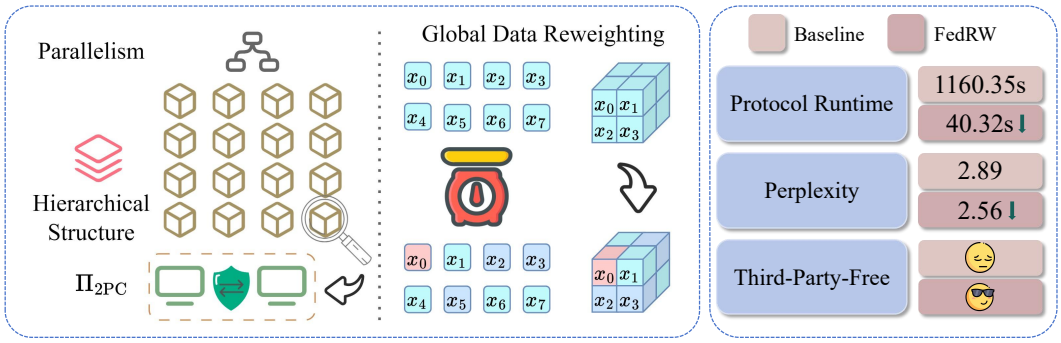


Figure 2: FedRW Framework: Parallel Π_{2PC}-based Reweighting for Efficient FL. The overview is divided into three parts: **(Left)** The parallel orchestration of the third-party-free Π_{2PC} protocol. **(Center)** The frequency-aware reweighting scheme that dynamically assigns weights (reflected by color) to samples while preserving data integrity. **(Right)** A comparison between FedRW and the baseline approach.

4.1 Formal definition of PPMPR

Consider a federated setting with n clients $P_1, \ldots, P_n$, where each client P_i holds a local dataset $X_i = [x_1^i, \ldots, x_{m_i}^i]$ consisting of m_i text samples. The objective of our proposed PPMPR protocol is to assign a weight to each sample in a privacy-preserving manner, based on how often it appears across all datasets. This functionality f_{PPMPR} can be formally defined as:

$$f_{\text{PPMPR}}(X_1, ..., X_n) \rightarrow (W_1, ..., W_n), \tag{4}$$

where $W_i = [w_1^i, \ldots, w_{m_i}^i]$ refers to the weight vector for the samples in X_i. Specifically, each sample x_j^i is associated with an individualized weight w_j^i reflecting its global frequency.

Subsequently, the derived weights are applied to enhance the federated training of LLMs, providing a fine-grained pattern to handle duplicated data. To quantify the relative informativeness of each sample x, we employ an intuitive yet effective heuristic: the weight $w(x)$ is inversely proportional to its global frequency:

$$w(x) \propto \frac{1}{freq_{global}(x)}. \tag{5}$$

Here, $freq_{global}(x)$ denotes the occurrence frequency of the sample x within the entire dataset, specifically, the concatenation of all clients' local datasets. This formulation naturally turns the reweighting task into a challenge of securely deriving global frequencies without revealing local data. To solve this problem, we leverage a secure multi-party computation (MPC) approach.

To avoid reliance on a trusted third party, the procedure is decomposed into multiple rounds of secure two-party computation (2PC), a sub-issue of MPC. In a 2PC protocol, two clients, P_i and P_j, jointly compute a specific function based on their private inputs, X_i and X_j, without directly disclosing the inputs to each other. The functionality f_{2PC} defined for this situation is:

$$f_{\text{2PC}}(X_i, X_j) \rightarrow (\vec{C}_i, \vec{C}_j), \tag{6}$$

where $\vec{C}_i$ is vector of length m_i, containing the counts of samples in X_j that are identical to each sample x in X_i. Since there are no privacy concerns client-side, each unique sample x will be maintained only once in X_i, along with its local frequency, $freq_{X_i}(x)$, which can be easily collected and securely shared with other clients that hold the same sample. Through this iterative pairwise 2PC protocol, each client computes and obtains the global frequency for its local samples, allowing them to adjust the sample weights without exposing private data.

4.2 Efficient construction of PPMPR

To realize the defined functionalities, we utilize two-party private set intersection (PSI) as the cryptographic foundation of our 2PC protocol. PSI enables two parties to compute the intersecting elements of their datasets without revealing any additional information beyond the agreed-upon rules. The protocol involves only the two participating parties as sender and receiver. In the semi-honest setting, the protocol reveals solely the shared samples and how often they appear in each local dataset, as specified by f_{2PC}. The detailed procedure is outlined in Protocol 1.

Table 1: Protocol Π_{2PC} in the semi-honest setting model

Protocol 1 Two-Party Computation (2PC)

Input: Client P_1 holds input $X_1 = \{x_1^1, ..., x_{m_1}^1\}$, and client P_2 holds input $X_2 = \{x_1^2, ..., x_{m_2}^2\}$. Both input sets are preprocessed local data samples.

Output: P_1 outputs $\vec{C_1}$, and P_2 outputs $\vec{C_2}$, as defined in Eq. (6).

Protocol:

1. P_1 and P_2 initiate a two-party Private Set Intersection (PSI) protocol, where:
   - P_1 acts as sender, and receives nothing.
   - P_2 acts as receiver, and receives the intersection set $\mathcal{I}$ of P_1's data.
2. For each sample x in $\mathcal{I}$, P_2 extracts the local frequency $freq_{X_2}(x)$, and creates the frequency set $\mathcal{F}_2$. P_2 then sends $\mathcal{I}$ and $\mathcal{F}_2$ to P_1.
3. Upon receiving $\mathcal{I}$ and $\mathcal{F}_2$, P_1 extracts the local frequency $freq_{X_1}(x)$, and creates the frequency set $\mathcal{F}_1$. P_1 then sends $\mathcal{F}_1$ to P_2.
4. P_1 outputs $\vec{C_1}$ = $[freq_{X_2}(x_1^1), \ldots, freq_{X_2}(x_{m_1}^1)]$, and P_2 outputs $\vec{C_2}$ = $[freq_{X_1}(x_1^2), \ldots, freq_{X_1}(x_{m_2}^2)]$.

The 2PC protocol provides an efficient and secure method for pairwise exchange of sample frequencies between clients. We now extend this building block to construct the full PPMPR protocol.

Table 2: Protocol Π_{PPMPR} in the semi-honest setting model

Protocol 2 Full Protocol (PPMPR)

Input: Each client P_i holds a local dataset $X_i = \{x_1^i, \ldots, x_{m_i}^i\}$, where $i \in \{1, \ldots, n\}$. All datasets are preprocessed.

Output: Each P_i outputs a frequency vector $\vec{\mathcal{C}_i}$ containing $freq_{global}(x)$ for every x in X_i, as defined in Eq. (5).

Protocol:

1. Each P_i initialize $\vec{\mathcal{C}_i}$ using its local frequencies $freq_{X_i}(x)$ for all x in X_i.
2. Each P_i performs Π_{2PC} once with every other client P_j (all $n-1$ of them).
   - After each run, P_i outputs $\vec{C_i}$ and updates its global vector: $\vec{\mathcal{C}_i} \leftarrow \vec{\mathcal{C}_i} + \vec{C_i}$.
3. After $n-1$ rounds, P_i outputs the final $\vec{\mathcal{C}_i}$.

As presented in Protocol 2, Each client P_i starts by initializing its frequency vector with local counts, then iteratively executes Π_{2PC} with every other clients to progressively build $\vec{\mathcal{C}_i}$, the vector of global frequencies. The formal security definitions and proofs are available in Appendix A.

4.3 Parallel Acceleration

The formula "N choose 2" represents that the full protocol involves each pair of the n clients performing Π_{2PC}, which results in $\binom{n}{2}$ executions, leading to an overall time complexity of $O(n^2)$ when run sequentially. This quickly becomes inefficient as a growing number of clients. To address this scalability bottleneck, we introduce a parallel orchestration strategy that reorganizes the execution schedule to minimize overall runtime. We start with a toy example where $n = 8$ in Figure 3, and the detailed procedure is provided in Appendix B.

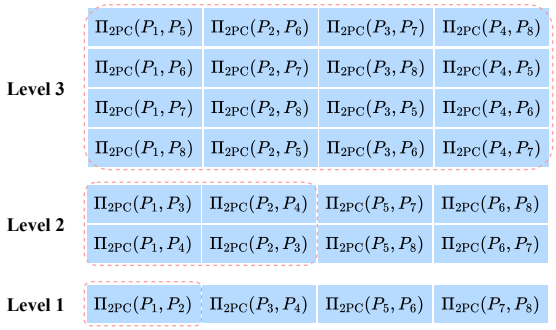


Figure 3: A toy example for the parallel orchestration when $n = 8$.

The key insight is that multiple Π_{2PC} instances can be performed concurrently, provided their participating sets do not overlap. As shown in Figure 3, from the left-hand side of level 1, adjacent pairs of clients perform Π_{2PC} independently. At the next level, these client pairs are grouped into disjoint blocks (e.g., $\{P_1, P_2\}$ with $\{P_3, P_4\}$), and inter-block protocols are executed in parallel. This hierarchical process forms progressively larger blocks, such as $\{P_1, P_2, P_3, P_4\}$ and $\{P_5, P_6, P_7, P_8\}$ at level 3. The structure resembles a binary tree and can be viewed as a recursive two-way merge that manages all $\binom{n}{2}$ sub-protocols efficiently.

To organize this orchestration, the client pairings at each level are structured into pairing matrices, with partial examples highlighted in the dashed-line areas of Figure 3. When n is a power of two, these matrices perfectly arrange all Π_{2PC} executions, maximizing parallelism. Each matrix is constructed by element-wise pairing of two client blocks. For instance, at level 3, matrix $\mathcal{M}_3$ is formed as follows:

$$\begin{aligned}
&\vec{a} := (1,2,3,4), \quad \vec{b} := (5,6,7,8) \\
&\vec{b'} \leftarrow \texttt{RotL}(\vec{b}, 0), \quad row_1 \leftarrow \{(\vec{a_i}, \vec{b'_i}) | i = 1,2,3,4\} \\
&\vec{b'} \leftarrow \texttt{RotL}(\vec{b}, 1), \quad row_2 \leftarrow \{(\vec{a_i}, \vec{b'_i}) | i = 1,2,3,4\} \\
&\vec{b'} \leftarrow \texttt{RotL}(\vec{b}, 2), \quad row_3 \leftarrow \{(\vec{a_i}, \vec{b'_i}) | i = 1,2,3,4\} \\
&\vec{b'} \leftarrow \texttt{RotL}(\vec{b}, 3), \quad row_4 \leftarrow \{(\vec{a_i}, \vec{b'_i}) | i = 1,2,3,4\}
\end{aligned} \tag{7}$$

Here, $\vec{a}$ and $\vec{b}$ contain the indices of clients from interacting blocks. In each step, $\vec{b'}$ is generated by cyclically left-shifting $\vec{b}$ by k positions using $\texttt{RotL}(\vec{b}, k)$. Client pairs are then formed by matching elements from $\vec{a_i}$ and $\vec{b'_i}$, allowing Π_{2PC} to run concurrently across each row. For $2^{m-1} < n \leq 2^m$, the hierarchical structure remains valid by simply ignoring the unused blocks, thus maintaining optimality and full coverage of client interactions. This parallel approach reduces the total runtime complexity of the full protocol from $O(n^2)$ to $O(2^{\lceil \log_2 n \rceil} - 1)$.

4.4 Enhanced Training

To integrate duplication awareness into model optimization, FedRW employs a frequency-based sample reweighting strategy. Given the global frequency vector $\vec{\mathcal{C}}$, where each element represents the occurrence count of a local sample across all clients, the corresponding weight vector $\vec{\mathcal{W}}$ is defined as:

$$\vec{\mathcal{W}} = \frac{1}{\ln(\vec{\mathcal{C}} + \vec{1}) + \vec{\varepsilon}} \tag{8}$$

Here, ε is a small constant for numerical stability. This formula penalizes frequent samples using a logarithmic function, reducing their impact on optimization without complete exclusion. The logarithm, shifted by 1, ensures that the weights decrease moderately and prevents extreme weights for infrequent samples (e.g., when $\vec{\mathcal{C}_i} = 1$). Compared to linear or hard-threshold formulas, this scheme offers a smoother and adaptive adjustment across varying duplication levels, leveraging the observation that moderate redundancy can promote better model generalization.

These derived weights, $\vec{\mathcal{W}}$, are then applied during training via a sample-wise reweighted loss. Instead of modifying the model architecture, each sample's loss contribution is rescaled by its assigned weight. For a batch of B samples, with $\vec{\mathcal{W}}_i$ as the weight and $\ell_i^{(t)}$ as the token-level average loss of the i-th sample, the aggregated batch loss is calculated as:

$$\mathcal{L}_{\text{batch}} = \frac{\sum_{i=1}^{B} \vec{\mathcal{W}}_i \cdot \ell_i^{(t)}}{\sum_{i=1}^{B} \vec{\mathcal{W}}_i} \tag{9}$$

This method diminishes the impact of frequent samples while balancing the influence of less frequent or underrepresented ones. By adapting to statistical redundancy across clients, it preserves informative samples and mitigates overfitting to specific patterns. This provides a lightweight yet effective sample-level reweighting mechanism, particularly advantageous in federated settings with skewed or redundant data. Model updates are then aggregated using the standard FedAvg [24] algorithm.

5 Experiments

5.1 Experimental Settings

Environments. For protocol evaluation, we implement the Π_{2PC} prototype based on [33] and benchmark its runtime under varying configurations. For FL experiments, we use eight public datasets: *Haiku* [34], *Rotten Tomatoes* [35], *Short Jokes* [36], *Poetry* [37], *IMDB* [38], *Sonnets* [39], *Plays*[40], and *Twitter Sentiment Analysis*[41]. To simulate redundancy, duplicates are synthetically added into the training set at different rates and distributed uniformly across 10 clients. The final performance of models is evaluated using perplexity on the test sets. More details can be found in Appendix C.

Baseline Setting. We choose EP-MPD [25] as the primary baseline, the most state-of-the-art hard deduplication solution for federated LLM training via a trusted third party. We follow their original experimental settings and directly use their reported runtime and perplexity results for comparison.

5.2 Main Results

Preprocessing. This part evaluates the efficiency and scalability of our proposed PPMPR protocol against the baseline across three key factors: **dataset size**, **client number**, and **duplication percentage**, with the results shown in tables 3 and 4, and figure 4.

Table 3: Effect of dataset size with 30% duplication percentage on Π_{2PC} running time.

Method	**Protocol Running Time (ms)**					
Dataset Size	2^{10}	2^{12}	2^{14}	2^{16}	2^{18}	2^{20}
Setup	$47.0_{\pm 0.002}$	$48.6_{\pm 0.003}$	$54.6_{\pm 0.078}$	$76.0_{\pm 0.178}$	$167.8_{\pm 0.478}$	$715.8_{\pm 1.841}$
Execution	$0.4_{\pm 0.006}$	$1.0_{\pm 0.006}$	$5.9_{\pm 0.019}$	$23.5_{\pm 0.325}$	$118.7_{\pm 1.738}$	$713.3_{\pm 7.600}$
Π_{2PC}-total	$47.4_{\pm 0.055}$	$49.7_{\pm 0.118}$	$60.9_{\pm 0.423}$	$100.8_{\pm 1.500}$	$291.8_{\pm 6.250}$	$1451.8_{\pm 27.141}$

The runtime of the basic 2PC protocol increases with dataset size due to the underlying frequency counting mechanism. For small datasets (e.g., 2^{10}-2^{14}), runtime differences are minimal, mainly because the cryptographic setup overhead of two-party PSI is significant compared with the actual execution time, which grows linearly with the dataset size. Noticeably, the execution time scales rapidly beyond a certain dataset size, and begins to dominate the total runtime. For instance, processing 2^{20} samples per client takes approximately 1.45 seconds, as illustrated in table 3.

Table 4 examines how the duplication percentage affects Π_{2PC} runtime when each client holds 2^{19} samples. The results show a negligible effect, with the protocol maintaining near-constant performance even at extreme duplication levels. For instance, with 90% duplication, the runtime remains stable at 0.666 seconds, differing by only 6.9% from the 0.620-second runtime with 10% duplication. These small variations are attributed to the increased amount of *frequency information exchange* ($\mathcal{F}_1$, $\mathcal{F}_2$ in 3) as the intersection ($\mathcal{I}$ in 3) cardinality grows.

Table 4: Effect of duplication percentage with 2^{19} data size in each client on Π_{2PC} running time.

Method	**Protocol Running Time (ms)**				
Duplication Percentage	10%	30%	50%	70%	90%
Setup	$342.2_{\pm 1.092}$	$322.4_{\pm 0.966}$	$337.4_{\pm 0.988}$	$339.8_{\pm 1.015}$	$343.4_{\pm 0.974}$
Execution	$265.9_{\pm 0.259}$	$293.3_{\pm 3.456}$	$284.9_{\pm 5.905}$	$304.2_{\pm 9.283}$	$310.6_{\pm 11.622}$
Π_{2PC}-total	$620.1_{\pm 12.213}$	$626.9_{\pm 13.091}$	$633.6_{\pm 14.766}$	$656.5_{\pm 18.170}$	$665.9_{\pm 18.913}$

Figure 4 analyzes the effect of client number on the runtime of the baseline and PPMPR under consistent experimental settings. The baseline proposes two variants that trade off performance and leakage, and we utilize EP-MPD$^{(\text{I})}$, which prioritizes efficiency while introducing more privacy leakage. PPMPR demonstrates superior efficiency and scalability, achieving a $17.61\times$ to 28.78 speedup over the baseline with 10 to 50 clients. This advantage is primarily due to the efficient Π_{2PC} protocol as its basic component, with the parallel orchestration strategy, which reduces the overall computational complexity to $O(m-1)$ for $n \in (2^{m-1}, 2^m]$ clients.

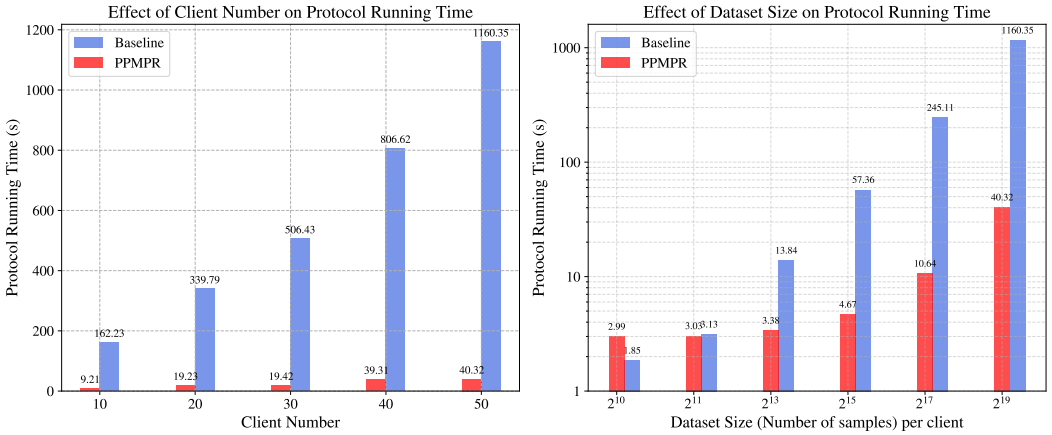


Figure 4: We evaluate the effect of client number and dataset size on protocol running time. For clients $(10\text{-}50)$ with 2^{19} data per client and 30% duplication, PPMPR exhibits $17.61\text{-}28.78\times$ speedup. For 50 clients, PPMPR outperforms the baseline by $4.09\text{-}28.78\times$ with increasing dataset size.

Furthermore, we evaluate the impact of dataset size per client with 50 clients. While PPMPR initially lags EP-MPD$^{(\text{I})}$ on smaller datasets, its parallel strategy quickly becomes dominant as data size scales. With 2^{17} samples per client, PPMPR achieves a $23.04\times$ speedup. This dual advantage in scaling across both client counts and data volumes positions PPMPR as a highly efficient and scalable solution for real-world federated environments.

Model Performance. This section evaluates model performance across eight text datasets with diverse linguistic structures. To simulate realistic data redundancy in FL, we introduce different levels of artificial duplication (10%, 20%, and 30%) into the training data. Initially, we assess the robustness of each method under two foundational models utilized in the baseline, GPT-2 Large [42] and DistilGPT2 [43], with perplexity as the evaluation metric.

Table 5: Model perplexity (↓) on test set under various duplication settings with GPT-2 Large

Method	**Dataset**											
	Haiku			Rotten Tomatoes			Short Jokes			Sonnets		
Duplication Percentage	30%	20%	10%	30%	20%	10%	30%	20%	10%	30%	20%	10%
Raw Data	3.26	3.25	2.98	2.65	2.61	2.53	4.11	4.03	3.94	4.39	4.34	4.31
Baseline	2.89	-	-	2.21	-	-	3.79	-	-	4.35	-	-
FedRW (Ours)	**2.56**	**2.67**	**2.69**	**1.61**	**1.63**	**1.64**	**3.15**	**3.17**	**3.17**	**4.07**	**4.26**	**4.26**

As detailed in Table 5, FedRW consistently outperforms the baseline across all datasets and duplication levels with GPT-2 Large. The improvement is evident on the highly structured *Sonnets* and *Haiku* datasets, where FedRW achieves relative perplexity reductions of up to 6.44% and 11.42% at 30% duplication, respectively. The strict structures of these datasets likely worsen the negative impact of redundancy, highlighting FedRW's ability to preserve content diversity and reduce overfitting through adaptive reweighting.

Similar trends are observed on less structured datasets. For *Short Jokes*, FedRW reduces perplexity from 3.79 to 3.15 under 30% duplication, despite its high lexical diversity. Likewise, on *Rotten Tomatoes*, which is composed of short, opinion-based reviews often prone to duplication, perplexity decreases from 2.21 to 1.61. These results indicate FedRW's effectiveness even when redundancy arises from stylistic repetition.

Furthermore, FedRW exhibits robustness to varying duplication rates. While the baseline's hard filtering yields fixed perplexity (10% to 30% duplication), FedRW maintains stable or slightly improved performance. For instance, perplexity on *Short Jokes* remains constant at 3.17, and on *Haiku*, it decreases from 2.69 to 2.56. These observations align with prior research suggesting that controlled repetition can enhance generalization by reinforcing key training patterns [44]. Instead of discarding duplicates, FedRW adaptively reweights updates to retain informative redundancy, as seen in datasets where increased duplication slightly improves performance. This suggests that effectively managed redundancy can amplify useful linguistic or semantic signals, underscoring FedRW's ability to adapt to varying levels of data noise.

Table 6: Model perplexity (↓) on test set under 30% duplication percentage with DistilGPT2

Method	**Dataset**						
	Haiku	Short Jokes	Rotten Tomatoes	IMDB	Poetry	Sonnets	Plays
Raw Data	3.70	**2.07**	1.78	7.17	2.84	5.87	15.07
Baseline	3.67	**2.07**	1.77	7.25	3.01	6.08	16.09
FedRW (Ours)	**3.65**	2.08	**1.75**	**7.00**	**2.66**	**5.75**	**14.50**

To evaluate FedRW's generalizability in resource-limited scenarios, we evaluate it with DistilGPT2, a smaller version of GPT-2 suitable for FL with limited computational resources. Despite its reduced size, which makes it more vulnerable to the negative effects of data duplication, Table 6 shows that FedRW consistently maintains or slightly improves performance across various datasets.

On datasets like *Haiku* and *Short Jokes*, perplexity remains similar across the three methods. However, more noticeable variances emerge on *Sonnets*, *Poetry*, and *Plays*, where the baseline sometimes underperforms even the undeduplicated data. This could be due to the literary structure and the sparse samples of these datasets. As noted in the baseline, hard deduplication considerably reduces the training samples (e.g., *Poetry*: 526 to 405; *Plays*: 542 to 417), potentially increasing training variance, especially for distilled models. By contrast, FedRW's flexible and adaptive approach aims to retain useful instances when handling excessive redundancy. This reweighting strategy provides a more stable training signal to preserve the integrity of sparse datasets, leading to improved generalization.

Table 7: Model perplexity (↓) on test set under 30% duplication percentage on mainsteam models

Model	**Method**	**Dataset**					
		Haiku	Jokes	Rotten	Poetry	Sonnets	Plays
Qwen3-0.6B	Baseline	2.47	2.61	1.71	2.54	4.07	8.21
	FedRW (Ours)	**2.36**	**2.44**	**1.59**	**2.21**	**3.62**	**7.23**
Qwen2.5-0.5B-Instruct	Baseline	2.21	2.48	1.58	2.28	4.11	11.77
	FedRW (Ours)	**2.12**	**2.36**	**1.55**	**2.03**	**3.84**	**9.92**
Llama-3.2-1B-Instruct	Baseline	2.14	2.34	1.65	2.39	4.11	18.35
	FedRW (Ours)	**2.09**	**2.21**	**1.54**	**1.99**	**4.00**	**16.03**

To further validate FedRW's applicability beyond the GPT-2 family, we evaluate the performance on three representative modern models with diverse architectures: Qwen3-0.6B [45], Qwen2.5-0.5B-Instruct [46], and Llama-3.2-1B-Instruct [47]. The results in Table 7 demonstrate that data redundancy remains a substantial challenge even for these contemporary architectures. FedRW robustly maintains its advantage in mitigating the impact of redundancy on model performance, particularly under challenging conditions such as data complexity or sparsity. For instance, FedRW achieves an average relative perplexity reduction of approximately 13.43% on the *Plays* dataset across the three models.

Table 8: Model perplexity (↓) on test set under 30% duplication percentage on larger models

Model	**Method**	**Dataset**						
		Haiku	Jokes	Rotten	Poetry	Sonnets	Plays	Twitter
Qwen2.5-3B-Instruct	Baseline	1.69	2.09	2.20	2.33	4.14	9.17	3.35
	FedRW (Ours)	**1.55**	**1.94**	**2.01**	**1.85**	**3.29**	**7.53**	**2.46**
Qwen2.5-7B-Instruct	Baseline	1.68	2.07	1.74	2.09	4.52	8.82	2.24
	FedRW (Ours)	**1.49**	**1.95**	**1.61**	**1.81**	**3.43**	**6.54**	**1.35**

With increasing model capacity, memorization of specific patterns due to duplication becomes more pronounced and critical, leading to overfitting, degraded generalization, and increased privacy risks [12]. To assess the issue, we conduct experiments on two large-scale models from the Qwen family: Qwen2.5-3B-Instruct and Qwen2.5-7B-Instruct [46]. While larger models may exhibit lower perplexity on certain datasets, the results in Table 8 show that FedRW sustains its performance advantage over the hard deduplication method. Under the extensive near-duplicate contents in *Twitter*, FedRW achieves a relative reduction of approximately 26.57% in perplexity compared to the baseline.

Table 9: Model Perplexity (↓) on test set on the Non-IID settings

Method	**IID**	**Quantity Skew**	**Label Skew**	**Feature Skew**
Baseline	1.71	2.02	2.44	3.43
FedRW (Ours)	**1.59**	**1.96**	**1.66**	**2.70**

To evaluate the efficacy of FedRW under Non-IID data distributions, a major challenge in FL, we conduct experiments on Qwen3-0.6B under three scenarios: *Quantity Skew*, *Label Skew*, and *Feature Skew*. For *Quantity* and *Label Skew*, we categorized the *Rotten Tomatoes* dataset by the binary (0/1) labels across 5 clients, with proportions set to [40%, 20%, 20%, 10%, 10%] and label distributions as [(0.5, 0.5), (0.6, 0.4), (0.4, 0.6), (0.9, 0.1), (0.1, 0.9)], respectively. To simulate *Feature Skew*, we allocate *Poetry*, *Sonnets*, and *Plays* to separate clients, as these datasets differ distinctly in terms of text structure, sentence length, and lexical and syntactic complexity. The results in Table 9 confirm FedRW's robustness to provide a stable training process across heterogeneous data distributions.

6 Conclusion

In this work, we introduce FedRW, a novel and principled framework designed to tackle the widespread challenge of data duplication in federated language model training. At its core is PPMPR, a secure and efficient protocol for data reweighting. PPMPR enables soft deduplication methods without compromising data privacy or introducing substantial computational and communication costs. Crucially, our protocol works without a trusted third party, enhancing security and achieving notable improvements in efficiency and scalability.

Our comprehensive experiments across diverse text datasets show that FedRW consistently improves model generalization under redundancy, outperforming the state-of-the-art method across varying duplication levels, dataset settings, and model configurations. Beyond simply discarding duplication, FedRW effectively harnesses redundancy to foster more robust representation learning. These compelling results establish FedRW as a practical, privacy-preserving solution for robust federated training in noisy data scenarios. Moreover, its lightweight modular design allows for seamless integration into broader applications, including multimodal learning pipelines and flexible reweighting strategies, highlighting its potential as a fundamental building block for future federated LLM systems.

Acknowledgements

This work is supported in part by the National Key Research and Development Program of China (Grant No. 2020YFA0712300), in part by the National Natural Science Foundation of China (Grant No. 62132005, 62172162), in part by Shanghai Trusted Industry Internet Software Collaborative Innovation Center, and in part by Fundamental Research Funds for the Central Universities. This work is also supported by the Postdoctoral Fellowship Program of CPSF under Grant Number GZB20250407.

References

[1] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, *et al.*, "Training language models to follow instructions with human feedback," *Advances in neural information processing systems*, vol. 35, pp. 27730–27744, 2022.

[2] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

[3] M. Hanna, O. Liu, and A. Variengien, "How does gpt-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained language model," *Advances in Neural Information Processing Systems*, vol. 36, pp. 76033–76060, 2023.

[4] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[5] A. Liu, B. Feng, B. Xue, B. Wang, B. Wu, C. Lu, C. Zhao, C. Deng, C. Zhang, C. Ruan, *et al.*, "Deepseek-v3 technical report," *arXiv preprint arXiv:2412.19437*, 2024.

[6] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, "Large language models are zero-shot reasoners," *Advances in neural information processing systems*, vol. 35, pp. 22199–22213, 2022.

[7] S. Wu, O. Irsoy, S. Lu, V. Dabravolski, M. Dredze, S. Gehrmann, P. Kambadur, D. Rosenberg, and G. Mann, "Bloomberggpt: A large language model for finance," *arXiv preprint arXiv:2303.17564*, 2023.

[8] G. Wang, Y. Xie, Y. Jiang, A. Mandlekar, C. Xiao, Y. Zhu, L. Fan, and A. Anandkumar, "Voyager: An open-ended embodied agent with large language models," *arXiv preprint arXiv:2305.16291*, 2023.

[9] J. Luo, Z. Pang, Y. Zhang, T. Wang, L. Wang, B. Dang, J. Lao, J. Wang, J. Chen, Y. Tan, *et al.*, "Skysensegpt: A fine-grained instruction tuning dataset and model for remote sensing vision-language understanding," *arXiv preprint arXiv:2406.10100*, 2024.

[10] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of machine learning research*, vol. 21, no. 140, pp. 1–67, 2020.

[11] K. Lee, D. Ippolito, A. Nystrom, C. Zhang, D. Eck, C. Callison-Burch, and N. Carlini, "Deduplicating training data makes language models better," *arXiv preprint arXiv:2107.06499*, 2021.

[12] N. Carlini, D. Ippolito, M. Jagielski, K. Lee, F. Tramer, and C. Zhang, "Quantifying memorization across neural language models," in *The Eleventh International Conference on Learning Representations*, 2022.

[13] E. Shayegani, M. A. A. Mamun, Y. Fu, P. Zaree, Y. Dong, and N. Abu-Ghazaleh, "Survey of vulnerabilities in large language models revealed by adversarial attacks," *arXiv preprint arXiv:2310.10844*, 2023.

[14] G. Qi, Y. Chen, X. Mao, B. Hui, X. Li, R. Zhang, and H. Xue, "Model inversion attack via dynamic memory learning," in *Proceedings of the 31st ACM International Conference on Multimedia*, pp. 5614–5622, 2023.

[15] J. Mattern, F. Mireshghallah, Z. Jin, B. Schölkopf, M. Sachan, and T. Berg-Kirkpatrick, "Membership inference attacks against language models via neighbourhood comparison," *arXiv preprint arXiv:2305.18462*, 2023.

[16] G. Penedo, Q. Malartic, D. Hesslow, R. Cojocaru, A. Cappelli, H. Alobeidli, B. Pannier, E. Almazrouei, and J. Launay, "The refinedweb dataset for falcon llm: outperforming curated corpora with web data, and web data only," *arXiv preprint arXiv:2306.01116*, 2023.

[17] A. Z. Broder, "On the resemblance and containment of documents," in *Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No. 97TB100171)*, pp. 21–29, IEEE, 1997.

[18] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.

[19] M. Ren, W. Zeng, B. Yang, and R. Urtasun, "Learning to reweight examples for robust deep learning," in *International conference on machine learning*, pp. 4334–4343, PMLR, 2018.

[20] S. M. Xie, H. Pham, X. Dong, N. Du, H. Liu, Y. Lu, P. S. Liang, Q. V. Le, T. Ma, and A. W. Yu, "Doremi: Optimizing data mixtures speeds up language model pretraining," *Advances in Neural Information Processing Systems*, vol. 36, pp. 69798–69818, 2023.

[21] S. M. Xie, S. Santurkar, T. Ma, and P. S. Liang, "Data selection for language models via importance resampling," *Advances in Neural Information Processing Systems*, vol. 36, pp. 34201–34227, 2023.

[22] N. He, W. Xiong, H. Liu, Y. Liao, L. Ding, K. Zhang, G. Tang, X. Han, and W. Yang, "Softdedup: an efficient data reweighting method for speeding up language model pre-training," *arXiv preprint arXiv:2407.06654*, 2024.

[23] C. Hou, A. Shrivastava, H. Zhan, R. Conway, T. Le, A. Sagar, G. Fanti, and D. Lazar, "Pre-text: Training language models on private federated data in the age of llms," *arXiv preprint arXiv:2406.02958*, 2024.

[24] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*, pp. 1273–1282, PMLR, 2017.

[25] A. Abadi, V. A. Dasu, and S. Sarkar, "Privacy-preserving data deduplication for enhancing federated learning of language models," *arXiv preprint arXiv:2407.08152*, 2024.

[26] S. Kamara, P. Mohassel, M. Raykova, and S. Sadeghian, "Scaling private set intersection to billion-element sets," in *Financial Cryptography and Data Security: 18th International Conference, FC 2014, Christ Church, Barbados, March 3-7, 2014, Revised Selected Papers 18*, pp. 195–215, Springer, 2014.

[27] U. Manber and G. Myers, "Suffix arrays: a new method for on-line string searches," *siam Journal on Computing*, vol. 22, no. 5, pp. 935–948, 1993.

[28] C. Coleman, C. Yeh, S. Mussmann, B. Mirzasoleiman, P. Bailis, P. Liang, J. Leskovec, and M. Zaharia, "Selection via proxy: Efficient data selection for deep learning," *arXiv preprint arXiv:1906.11829*, 2019.

[29] M. Weber, D. Fu, Q. Anthony, Y. Oren, S. Adams, A. Alexandrov, X. Lyu, H. Nguyen, X. Yao, V. Adams, *et al.*, "Redpajama: an open dataset for training large language models," *Advances in neural information processing systems*, vol. 37, pp. 116462–116492, 2024.

[30] X. Zhang, A. Lv, Y. Liu, F. Sung, W. Liu, S. Shang, X. Chen, and R. Yan, "More is not always better? enhancing many-shot in-context learning with differentiated and reweighting objectives," *arXiv preprint arXiv:2501.04070*, 2025.

[31] Z. Lin, Z. Gou, Y. Gong, X. Liu, R. Xu, C. Lin, Y. Yang, J. Jiao, N. Duan, W. Chen, *et al.*, "Not all tokens are what you need for pretraining," *Advances in Neural Information Processing Systems*, vol. 37, pp. 29029–29063, 2024.

[32] S. Keelveedhi, M. Bellare, and T. Ristenpart, "{DupLESS}:{Server-Aided} encryption for deduplicated storage," in *22nd USENIX security symposium (USENIX security 13)*, pp. 179–194, 2013.

[33] P. Rindal and P. Schoppmann, "Vole-psi: fast oprf and circuit-psi from vector-ole," in *Annual international conference on the theory and applications of cryptographic techniques*, pp. 901–930, Springer, 2021.

[34] bfbarry, "Haiku dataset." `https://www.kaggle.com/datasets/bfbarry/haiku-dataset`, 2021.

[35] B. Pang and L. Lee, "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales," in *Proceedings of the ACL*, 2005.

[36] A. Moudgil, "Short jokes dataset." `https://www.kaggle.com/datasets/abhinavmoudgil95/short-jokes`, 2017.

[37] HuggingFace and contributors, "Poetry dataset." `https://huggingface.co/datasets/merve/poetry`, 2022.

[38] A. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pp. 142–150, 2011.

[39] Lambent, "Shakespeare sonnets diffused dataset." https://huggingface.co/datasets/Lambent/shakespearesonnetsdiffused, 2023. Accessed: 2024-05-04.

[40] Trelis, "Tiny shakespeare dataset." https://huggingface.co/datasets/Trelis/tiny-shakespeare, 2022.

[41] passionate nlp, "Twitter sentiment analysis dataset." https://www.kaggle.com/datasets/jp797498e/twitter-entity-sentiment-analysis, 2021.

[42] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[43] HuggingFace and contributors, "Distilgpt2." https://huggingface.co/distilgpt2, 2019.

[44] N. Muennighoff, A. Rush, B. Barak, T. Le Scao, N. Tazi, A. Piktus, S. Pyysalo, T. Wolf, and C. A. Raffel, "Scaling data-constrained language models," *Advances in Neural Information Processing Systems*, vol. 36, pp. 50358–50376, 2023.

[45] A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv, *et al.*, "Qwen3 technical report," *arXiv preprint arXiv:2505.09388*, 2025.

[46] A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, *et al.*, "Qwen2.5 technical report," *arXiv preprint arXiv:2412.15115*, 2024.

[47] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan, *et al.*, "The llama 3 herd of models," *arXiv e-prints*, pp. arXiv–2407, 2024.

[48] R. Canetti, "Universally composable security: A new paradigm for cryptographic protocols," in *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, pp. 136–145, IEEE, 2001.

[49] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims in the abstract and introduction align with the proposed method (FedRW), its design (PPMPR), and empirical results, including privacy guarantees and performance benefits. See Section 1.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The discussion in Section 5.2 acknowledges limitations such as scenarios where baseline performs comparable. Section 6 and Appendix E indicate possible future developments.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Theoretical assumptions and proofs for PPMPR's security are provided in Appendix A, clearly defining the threat model and formal guarantees.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Sections 4 and 5, and Appendix C detail protocol implementations, datasets, and experimental settings sufficient to reproduce the results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: All datasets are publicly available with citations provided in Appendix C. We are working hard to promote the process of open source.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: See Appendix C.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The paper reports the average performance after repeated experiments for consistency.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: See Appendix C.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper complies with NeurIPS Code of Ethics in all aspects.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: The research conducted in the paper is to enhance privacy-preserving federated training of language models, without negative societal impacts.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risk.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The creators or original owners of the assets used in the paper have been appropriately recognized, and the licenses and terms of use have been explicitly mentioned and properly respected.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

Appendix

Contents

A Security **21**

B Parallel Ochestration Algorithm **24**

C Experimental Details **24**

D Sensitivity Analysis **25**

E Future Work **26**

A Security

A.1 Security Proof

Modern cryptographic protocols are typically analyzed under the simulation-based security paradigm, which formalizes security by comparing a protocol's behavior in the real world to that in an ideal world.

In the ideal world, a trusted third party honestly executes the desired functionality. All parties submit their inputs to the trusted third party, and the trusted third party returns the correct outputs to the designated parties. In contrast, the real world involves actual protocol execution among potentially adversaries without a trusted third party.

A protocol is said to be secure if for every adversary in the real world, there exists a simulator in the real world such that no external environment can tell whether it is interacting with a real world or an ideal functionality. This paradigm ensures that the protocol leaks no more information than what is inherently revealed by the output of the ideal functionality.

A.2 Universal Composability Model

The Universal Composability (UC) [48] framework provides a rigorous model for analyzing the security of cryptographic protocols under arbitrary adversarial conditions. It ensures that a protocol remains secure even when composed with other protocols, making it robust against complex attack scenarios.

In the ideal world, all parties interact through a TTP that computes the desired functionality f, ensuring privacy and correctness. In the real world, parties execute a protocol Π without a TTP. A semi-honest adversary $\mathcal{A}$ may observe internal states but not deviate from the protocol.

A protocol Π is UC-secure if, for any adversary $\mathcal{A}$ in the real world, there exists a simulator $\mathcal{S}$ that produces a view indistinguishable from $\mathcal{A}$'s view in the ideal world. This ensures that the protocol's behavior in the real world is as secure as the ideal world.

A.3 Threat Model

In this work, we consider a **semi-honest adversary model** in federated learning (FL), where all participants follow the protocol honestly but may attempt to infer additional information from observations. For the scope of this work, we assume no active collusion among parties. While more active or malicious threats—such as inference, backdoor, or reconstruction attacks—exist, these are considered orthogonal to the primary objective of this study.

A protocol Π securely computes a functionality $f : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^* \times \{0,1\}^*$, where $f = (f_1, f_2)$. For inputs (x, y), outputs $(f_1(x, y), f_2(x, y))$ are returned to respective parties. Extensions to multi-party settings are implied.

A protocol Π is secure against semi-honest adversaries if:

Definition 1 (Security)**.** *For any semi-honest adversary $\mathcal{A}$, there exist probabilistic polynomial-time (PPT) simulators Sim_1, Sim_2 such that:*

$$\{\text{Sim}_1(x, f_1(x,y))\}_{x,y} \equiv_c \{\text{View}_1^{\Pi,\mathcal{A}}(x,y)\}_{x,y}, \tag{10}$$

$$\{\text{Sim}_2(y, f_2(x,y))\}_{x,y} \equiv_c \{\text{View}_2^{\Pi,\mathcal{A}}(x,y)\}_{x,y}. \tag{11}$$

Here, $\text{Sim}_i(w, f_i(x,y))$ denotes a view based on simulator i's input $w \in (x,y)$ and Π's output $f_i(x,y)$. $\text{View}_i^{\Pi,\mathcal{A}}(x,y)$ represents $\mathcal{A}$'s observation on party i's view during protocol execution. $\equiv_c$ denotes computational indistinguishability, meaning no PPT algorithm can distinguish the two distributions.

A.4 Formal Definition of Ideal Functionality

We provide the formal definitions of the ideal functionalities employed in Section 4, as detailed in tables 10 to 12.

Table 10: Ideal functionality $f_{\text{Two-Party PSI}}$

Parameters:	Client P_1 holds input $X_1 = \{x_1^1, ..., x_m^1\}$, and client P_2 holds input $X_2 = \{x_1^2, ..., x_n^2\}$.
Functionality:	• Input $X_1 = \{x_1^1, ..., x_m^1\}$ from P_1, and $X_2 = \{x_1^2, ..., x_n^2\}$ from P_2. • Output $X_1 \cap X_2$.

Table 11: Ideal functionality f_{2PC}

Parameters:	Client P_1 holds input $X_1 = \{x_1^1, ..., x_m^1\}$, and client P_2 holds input $X_2 = \{x_1^2, ..., x_n^2\}$.
Functionality:	• Input $X_1 = \{x_1^1, ..., x_m^1\}$ from P_1, and $X_2 = \{x_1^2, ..., x_n^2\}$ from P_2. • Output $\vec{C_1}$ and $\vec{C_2}$.

Table 12: Ideal functionality f_{PPMPR}

Parameters:	Each client P_i holds a local dataset $X_i = \{x_1^i, \ldots, x_{m_i}^i\}$, where $i \in \{1, \ldots, n\}$.
Functionality:	• Input $X_i = \{x_1^1, ..., x_{m_i}^1\}$ from P_i. • Output $\vec{C_i}$.

A.5 Security of Protocols

Theorem 1. *Π_{2PC} securely implements the ideal functionality f_{2PC} in the semi-honest model.*

Proof. As described in f_{2PC}, we construct a simulator to simulate the behavior of the corrupted party.

Case 1: P_1 is corrupted.

The simulator Sim_1 receives P_1's input X_1 and its output from f_{2PC}, which is $\vec{C_1} = [freq_{X_2}(x_1^1), ..., freq_{X_2}(x_{m_1}^1)]$.

1. During the PSI phase, P_1 acts as the sender and receives nothing. As PSI protocol securely implements corresponding ideal functionality, then P_1 learns nothing about X_2 beyond what is revealed by the intersection $X_1 \cap X_2$. Sim_1 can simulate this as an empty view with its own inputs X_1.

2. P_1 receives the intersection set $\mathcal{I}$ and the frequency set $\mathcal{F}_2$ from P_2. Sim_1 can construct a simulated intersection $\mathcal{I}'$ and a simulated frequency set $\mathcal{F}_2'$ based on X_1 and $\vec{C_1}$. For each $x_k^1 \in X_1$:

- If $freq_{X_2}(x_k^1) > 0$, then x_k^1 is added to $\mathcal{I}'$, and its corresponding frequency in $\mathcal{F}_2'$ is set to $freq_{X_2}(x_k^1)$.
- If $freq_{X_2}(x_k^1) = 0$, then x_k^1 is not in $\mathcal{I}'$.

The outputs $(\mathcal{I}', \mathcal{F}_2')$ in the ideal world are indistinguishable from the outputs $(\mathcal{I}, \mathcal{F}_2)$ in the real world, as they perfectly match P_1's output $\vec{C}_1$.

3. P_1 sends its frequency set $\mathcal{F}_1$ to P_2. Sim_1 can generate $\mathcal{F}_1$ using X_1 and the intersection set $\mathcal{I}$ (or $\mathcal{I}'$).

The view of P_1 consists of its input X_1, messages sent ($\mathcal{F}_1$), and messages received ($\mathcal{I}, \mathcal{F}_2$). The simulated view $(X_1, \mathcal{I}', \mathcal{F}_1, \mathcal{F}_2')$ is computationally indistinguishable from the real view $(X_1, \mathcal{I}, \mathcal{F}_1, \mathcal{F}_2)$.

Case 2: P_2 is corrupted.

The simulator Sim_2 receives P_2's input X_2 and its output from f_{2PC}, which is $\vec{C}_2 = [freq_{X_1}(x_1^2), ..., freq_{X_1}(x_{m_2}^2)]$.

1. During the PSI phase, P_2 acts as the receiver and receives the intersection set $\mathcal{I}$. Given the security of the PSI protocol, P_2 learns nothing about X_1 beyond what is revealed by the intersection $X_1 \cap X_2$. Sim_2 can construct a simulated intersection $\mathcal{I}'$ based on X_2 and $\vec{C}_2$. For each $x_k^2 \in X_2$:

- If $freq_{X_1}(x_k^2) > 0$, then x_k^2 is added in $\mathcal{I}'$.
- If $freq_{X_1}(x_k^2) = 0$, then x_k^2 is not in $\mathcal{I}'$.

2. P_2 sends $\mathcal{I}$ (or $\mathcal{I}'$) and $\mathcal{F}_2$ to P_1. Sim_2 can perfectly simulate this using X_2 and $\mathcal{I}'$.

3. P_2 receives $\mathcal{F}_1$ from P_1. Sim_2 can construct a simulated $\mathcal{F}_1'$ based on $\vec{C}_2$ and $\mathcal{I}'$. For each $x \in \mathcal{I}'$, the corresponding frequency in $\mathcal{F}_1'$ would be $freq_{X_1}(x)$.

The view of P_2 consists of its input X_2, messages sent ($\mathcal{I}, \mathcal{F}_2$), and messages received ($\mathcal{F}_1$). The simulated view $(X_2, \mathcal{I}', \mathcal{F}_2, \mathcal{F}_1')$ is computationally indistinguishable from the real view $(X_2, \mathcal{I}, \mathcal{F}_2, \mathcal{F}_1)$.

Since the view of both corrupted parties can be simulated given their input and output from f_{2PC}, Π_{2PC} securely realizes f_{2PC} in the semi-honest model. □

Theorem 2. *Π_{PPMPR} securely implements the ideal functionality f_{PPMPR} in the semi-honest model.*

Proof. We construct a simulator Sim_{PPMPR} for a corrupted P_k that receives P_k's input X_k and its final output the global frequency vector $\vec{\mathcal{C}}_k$. from the ideal functionality f_{PPMPR}

1. P_k initializes $\vec{\mathcal{C}}_k$ using its local frequencies $freq_{X_k}(x)$. This is a local computation, and Sim_{PPMPR} can perform the same step.

2. P_k performs Π_{2PC} with every other client P_j (for $j \neq k$). After each execution, P_k receives a vector $\vec{C}_k$ and updates $\vec{\mathcal{C}}_k \leftarrow \vec{\mathcal{C}}_k + \vec{C}_k$.

- For each interaction between P_k and an honest P_j, the security proof for Π_{2PC} guarantees that a simulator Sim_{2PC} can generate a view for P_k that is indistinguishable from the real view, using only X_k and the output $\vec{C}_k$.
- Since the final $\vec{\mathcal{C}}_k$ is the sum of P_k's local frequencies and pairwise learned frequencies, the overall view of P_k is the collection of views with the $n-1$ executions of Π_{2PC}. Sim_{PPMPR} can invoke Sim_{2PC} for each interaction between P_k and P_j to generate a view to simulate this combination.
- Since f_{PPMPR} only outputs the final $\vec{\mathcal{C}}_k$, Sim_{PPMPR} cannot obtain each partial $\vec{C}_k$. However, it can generate intermediate $\vec{C}_k'$ for each interaction such that their sum (plus the initial vector) equals the known final $\vec{\mathcal{C}}_k$. Given that Π_{2PC} securely reveals only $freq_{X_j}(x)$ for intersecting samples, the exact distribution does not leak additional information to P_k beyond what f_{PPMPR} allows.

3. After $n-1$ rounds, P_k outputs the final $\vec{\mathcal{C}}_k$.

The view of P_k consists of its input X_k, its initial local frequencies, and the collection of outputs from all $n-1$ pairwise Π_{2PC} executions. Since each Π_{2PC} is secure against semi-honest adversaries and its view can be simulated, the collection of these simulated views can be combined by Sim_{PPMPR} securely. Therefore, Sim_{PPMPR} constructs a view for P_k that is computationally indistinguishable from its view in a real execution. Thus, Π_{PPMPR} securely realizes f_{PPMPR} in the semi-honest model. □

B Parallel Ochestration Algorithm

To support the parallel acceleration strategy introduced in Section 4.3, we formally describe the orchestration logic in Algorithm 1. The algorithm organizes client pairs in a structured matrix manner, ensuring that each client performs 2PC protocols with all others while maximizing concurrency. Specifically, it proceeds in $\lceil \log_2 n \rceil$ hierarchical levels, where clients are iteratively grouped into blocks and scheduled to engage in pairwise protocols via index cyclic rotation. The orchestration guarantees correctness while enabling efficient parallelization.

Algorithm 1 Parallel Orchestration for Efficient Execution of PPMPR

Input: n clients $P_1, \ldots, P_n$ with local datasets $X_1, \ldots, X_n$
Output: Global frequency vectors $\vec{\mathcal{C}_1}, \ldots, \vec{\mathcal{C}_n}$ for samples of each client
Initialize local frequencies: $\vec{\mathcal{C}_i} \leftarrow freq_{X_i}(\cdot)$ for all i
Let $m \leftarrow \lceil \log_2 n \rceil$ ▷ Total number of levels
for $l = 1$ to m **do**
    Partition clients into 2^{m-l} contiguous blocks of equal size
    for all pairs of blocks (A, B) **do**
        Let $\vec{a} \leftarrow$ indices in A, $\vec{b} \leftarrow$ indices in B
        for $k = 0$ to $|\vec{b}| - 1$ **do**
            $\vec{b'} \leftarrow \texttt{RotL}(\vec{b}, k)$ ▷ Left-rotate indices in $\vec{b}$
            for $i = 1$ to $|\vec{a}|$ **do**
                in parallel: run $\Pi_{\text{2PC}}(P_{\vec{a_i}}, P_{\vec{b'_i}})$ to update $\vec{\mathcal{C}}_{\vec{a_i}}, \vec{\mathcal{C}}_{\vec{b'_i}}$
            end for
        end for
    end for
end for
return $\{\vec{\mathcal{C}_1}, \ldots, \vec{\mathcal{C}_n}\}$

C Experimental Details

Datasets. In this section, we summarize the detailed information of each dataset used in the experiment. As illustrated in Table 13, these datasets span diverse text domains and reflect a wide range of structural and lexical properties. The table presents the source, sample size, average sequence length, and a brief description for each dataset.

Table 13: Basic information of experimental datasets

Dataset	# Samples	Avg. Sequence Length	Description
Haiku [34]	15,281	100	Short-form structured 3-line poems
Short Jokes [36]	231,657	100	Concise User-written short jokes
Rotten Tomatoes [35]	10,662	200	Movie review snippets expressing sentiment
IMDB [38]	49,999	500	Full-length movie reviews with richer narrative structure
Sonnets [39]	460	400	William Shakespeare's 14-line poems
Poetry [37]	573	1000	Modern and classic free-form poems by various authors
Plays [40]	521	1000	Dramatic scripts from William Shakespeare with dialogic structure
Twitter [41]	74,000	50	Tweets labeled with sentiment categories

For all datasets, we adopt a standard 80/20 train/test split. For movie review datasets, only the review texts are retained, and the sentiment labels are discarded during training. For the *Short Jokes* dataset, we randomly sample 50,000 entries to ensure tractable training time across 10 clients. In cases where

datasets such as *IMDB* already contain a predefined test set, we merge the original training and test partitions, shuffle the combined set, and then re-split it according to the 80/20 ratio.

Environments. We conduct all secure protocol procedures, including Π_{2PC} and Π_{PPMPR}, on a virtualized server equipped with a 4-core Intel Xeon 2.20GHz CPU and 32GB RAM. For model training, we utilize a machine with a 20-core Intel Xeon Platinum 8457C CPU, 200GB RAM, and an NVIDIA H20 GPU with 96GB memory. All software is executed under the Linux environment. Each experiment in preprocessing is repeated four times, and we report the average performance for consistency.

Hyperparameters. We adopt FedAvg [24] as the underlying federated optimization algorithm. For GPT-2 Large and DistilGPT2, we train each client for 1–2 and 1–5 local epochs, respectively, until convergence, with a total of 3–5 communication rounds. For Qwen3-0.6B, Qwen2.5-0.5B-Instruct, and Llama-3.2-1B-Instruct, we train each client for 2 local epochs with a total of 2-5 communication rounds. For Qwen2.5-3B-Instruct and Qwen2.5-7B-Instruct, we train each client for 1–2 local epochs with a total of 1-2 communication rounds to avoid overfitting. The models are optimized using AdamW [49] with learning rates ranging from 1-5 $\times$ 10^{-5}. A linear warm-up schedule is applied, reserving 10% of training steps for warm-up. To stabilize training, we apply ℓ_2-norm gradient clipping with a threshold of 1.0. The maximum sequence length is set between 50 and 1000, depending on the dataset, and batch sizes range from 2 to 8 with gradient accumulation steps adjusted accordingly to maintain effective batch size.

Baseline. We follow the baseline implementation proposed in [25], which proposes a hard deduplication approach by pre-filtering duplicated training samples. Specifically, each client performs local deduplication to remove identical samples, which assumes that redundant data is uniformly detrimental, and the resulting datasets are used to train the model without further adjustment.

To ensure fair comparison, we utilize the official open-sourced code[3] and apply the same preprocessing pipeline and training settings as in FedRW. All datasets, tokenization schemes, model architectures, and evaluation metrics remain consistent across the baseline and our proposed method.

D Sensitivity Analysis

The discussion on sensitivity analysis focuses on the learning rate to assess FedRW's robustness. The analysis of epochs is omitted as we typically utilize a small number as standard practice to prevent overfitting.

We evaluated the model perplexity on DistilGPT2 and Qwen2.5-0.5B-Instruct under learning rates of 1e-3, 5e-4, 3e-4, 1e-4, 5e-5, and 3e-5. We selected the *Plays* dataset to investigate FedRW's generalizability, as it exhibited a significant performance gap in the main results .

Table 14: Model perplexity (↓) on *Plays* test set across various learning rates

Model	**Method**	**Learning Rate**					
		1e-3	5e-4	3e-4	1e-4	5e-5	3e-5
DistilGPT2	Baseline	16.12	16.38	16.13	14.21	15.07	14.18
	FedRW (Ours)	**14.42**	**14.79**	**14.74**	**13.17**	**14.50**	**12.76**
Qwen2.5-0.5B-Instruct	Baseline	12.86	11.07	10.63	11.50	11.77	10.67
	FedRW (Ours)	**11.48**	**9.35**	**8.15**	**8.14**	**9.92**	**9.81**

As shown in Table 14, FedRW robustly maintains its superior performance compared to the baseline and exhibits stable training behavior across the entire range of tested learning rates. This confirms that FedRW's advantage is not overly sensitive to the learning rate selection.

[3] `https://github.com/vdasu/deduplication`

E Future Work

Advanced Paradigms. FedRW's lightweight, modular design enables seamless integration into broader applications, including multimodal learning pipelines and flexible reweighting strategies. Integration with personalized FL (e.g., diverse model architectures or personalization strategies) and dynamic client adaptation (where clients join, leave, or exhibit varying computational capabilities) are also valuable aspects for future research.

Optimizations. Addressing semantic redundancy is a significant issue in large-scale real-world corpora for LLMs. It is prospective to leverage the representation learning capability of transformer-based architectures to extract semantic duplication.

Adversarial Security. FedRW primarily operates under a semi-honest threat model, which is standard and foundational for practical privacy-preserving protocols. Extending FedRW to resist malicious adversaries would be an interesting research direction. This could involve integrating mechanisms like Differential Privacy on sample frequencies or utilizing Zero-Knowledge Proofs to verify client consistency during pre-processing and training. These potential schemes trade off between model accuracy, data privacy, and computational overhead.