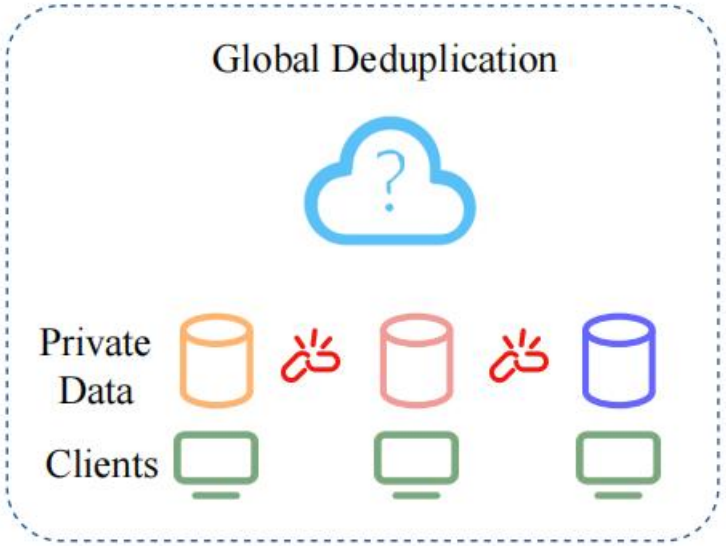
FedRW: Efficient Privacy-Preserving Data Reweighting for Enhancing Federated Learning of Language Models

Pukang Ye

East China Normal University

Overview

- Key problem: Data deduplication in federated LLM training.
  - Data duplication → memorization, model performance↓, attacks↑...
  - Global deduplication across multiple clients cannot be directly resolved due to privacy constraints.



Overview of SOTA work

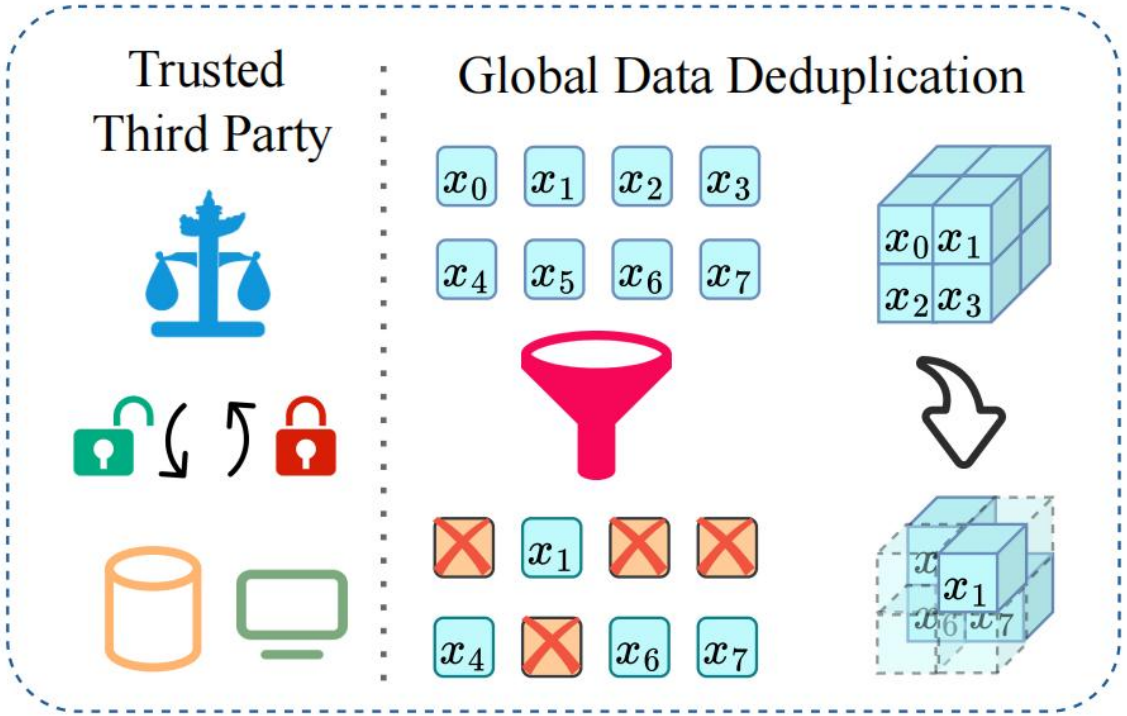
- EP-MPD[1]

Pros

- data privacy

Cons

- hard deduplication
- computational/communication overheads
- reliance on trusted 3rd parties



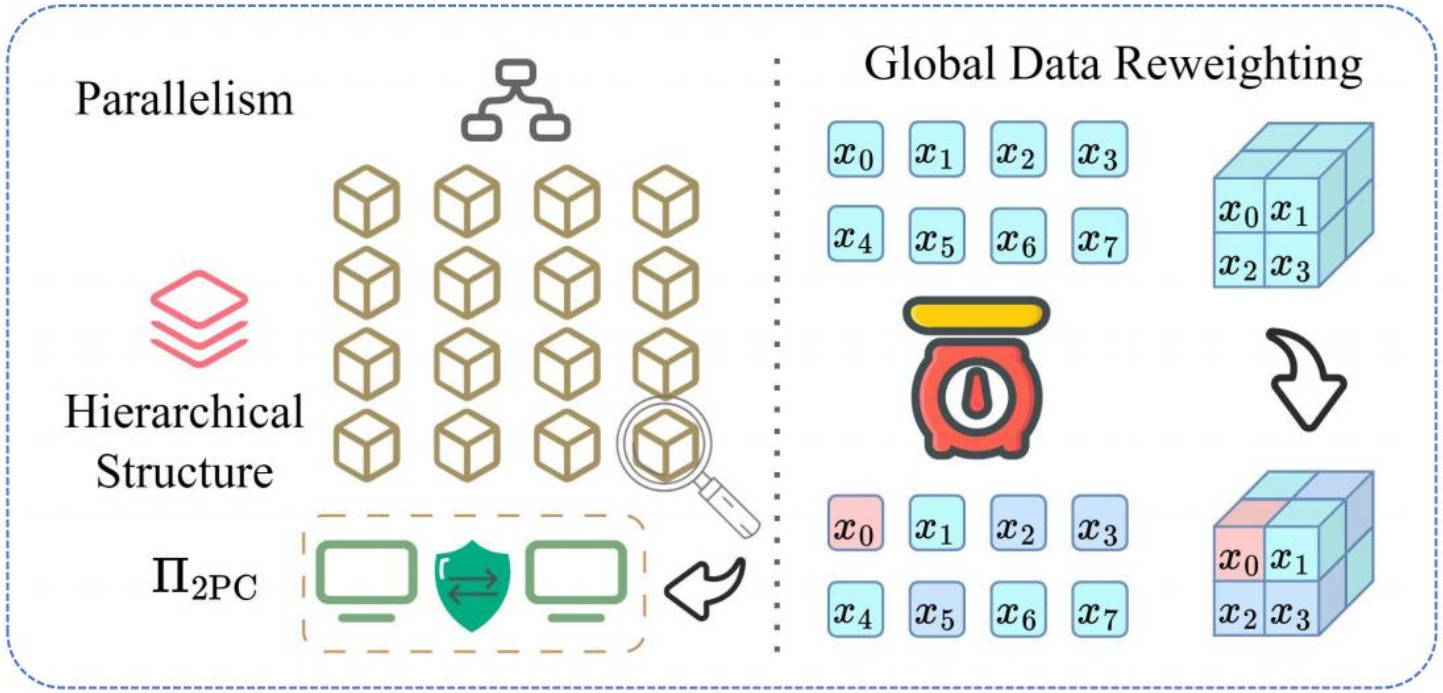
[1] A. Abadi, et al. "Privacy-preserving data deduplication for enhancing federated learning of language models." NDSS'25

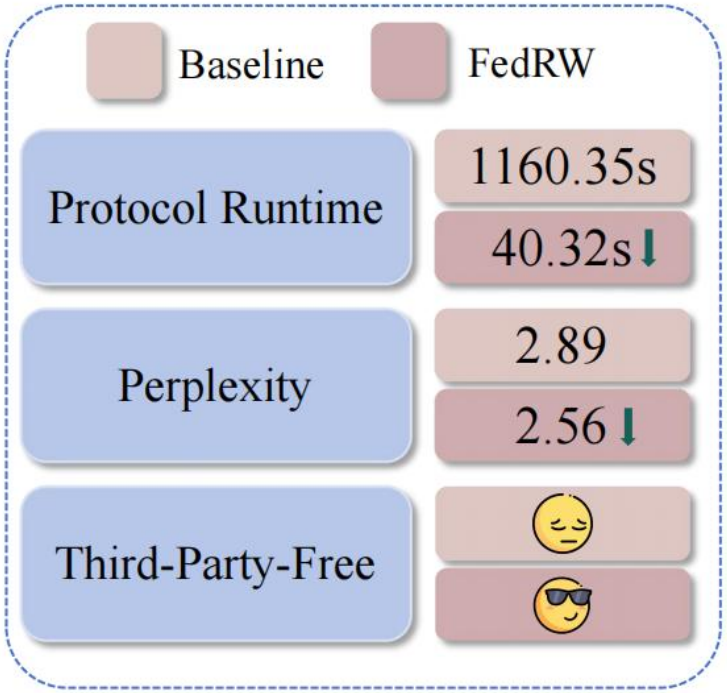
Our solution - FedRW

- Soft deduplication - global data reweighting

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

→ ***MPC Problem***





Our solution - FedRW

- "N choose 2" formula → Parallelism $O(n^2) \rightarrow O(2^{\lceil log_2 n \rceil} - 1)$

Level 3	$\Pi_{2PC}(P_1, P_5)$	$\Pi_{2PC}(P_2, P_6)$	$\Pi_{2PC}(P_3, P_7)$	$\Pi_{2PC}(P_4, P_8)$
	$\Pi_{2PC}(P_1, P_6)$	$\Pi_{2PC}(P_2, P_7)$	$\Pi_{2PC}(P_3, P_8)$	$\Pi_{2PC}(P_4, P_5)$
	$\Pi_{2PC}(P_1, P_7)$	$\Pi_{2PC}(P_2, P_8)$	$\Pi_{2PC}(P_3, P_5)$	$\Pi_{2PC}(P_4, P_6)$
	$\Pi_{2PC}(P_1, P_8)$	$\Pi_{2PC}(P_2, P_5)$	$\Pi_{2PC}(P_3, P_6)$	$\Pi_{2PC}(P_4, P_7)$
Level 2	$\Pi_{2PC}(P_1, P_3)$	$\Pi_{2PC}(P_2, P_4)$	$\Pi_{2PC}(P_5, P_7)$	$\Pi_{2PC}(P_6, P_8)$
	$\Pi_{2PC}(P_1, P_4)$	$\Pi_{2PC}(P_2, P_3)$	$\Pi_{2PC}(P_5, P_8)$	$\Pi_{2PC}(P_6, P_7)$
Level 1	$\Pi_{2PC}(P_1, P_2)$	$\Pi_{2PC}(P_3, P_4)$	$\Pi_{2PC}(P_5, P_6)$	$\Pi_{2PC}(P_7, P_8)$

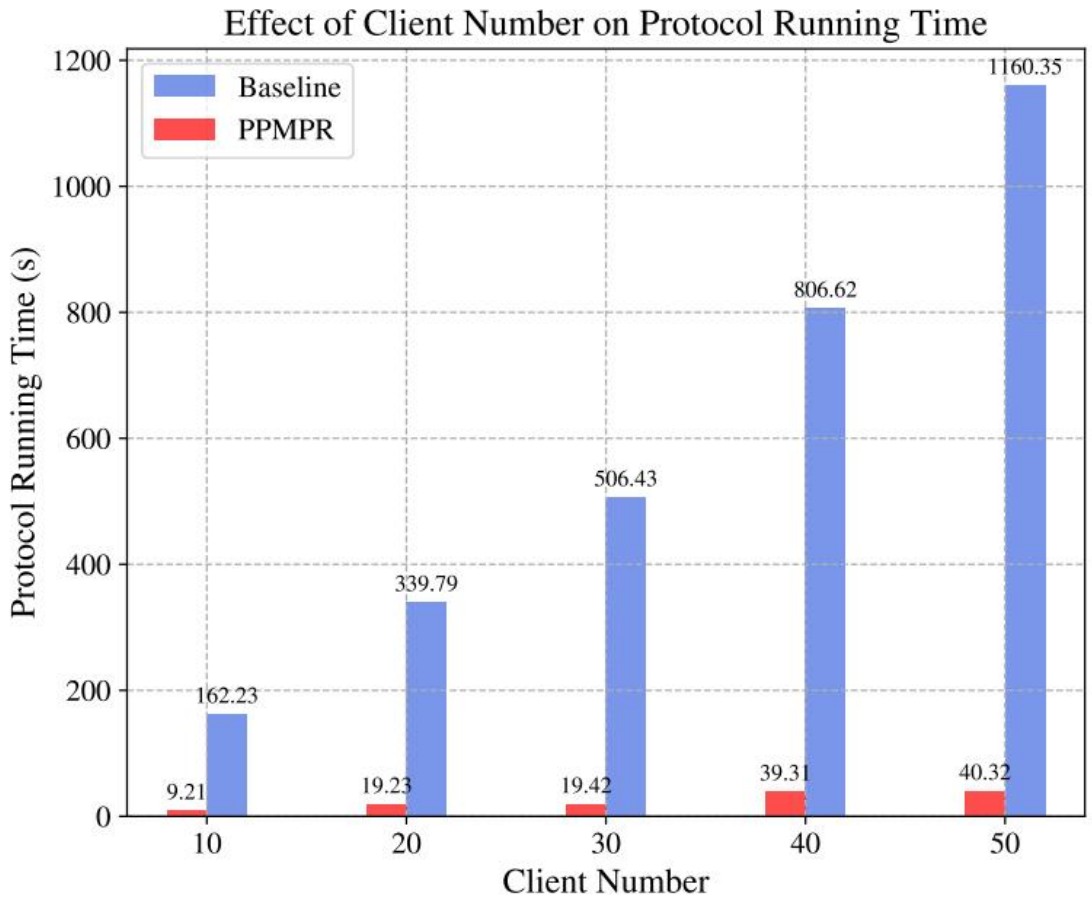
$$\vec{a} := (1,2,3,4), \quad \vec{b} := (5,6,7,8)$$
$$\vec{b'} \leftarrow \texttt{RotL}(\vec{b}, 0), \quad row_1 \leftarrow \{(\vec{a_i}, \vec{b'_i}) | i = 1,2,3,4\}$$
$$\vec{b'} \leftarrow \texttt{RotL}(\vec{b}, 1), \quad row_2 \leftarrow \{(\vec{a_i}, \vec{b'_i}) | i = 1,2,3,4\}$$
$$\vec{b'} \leftarrow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

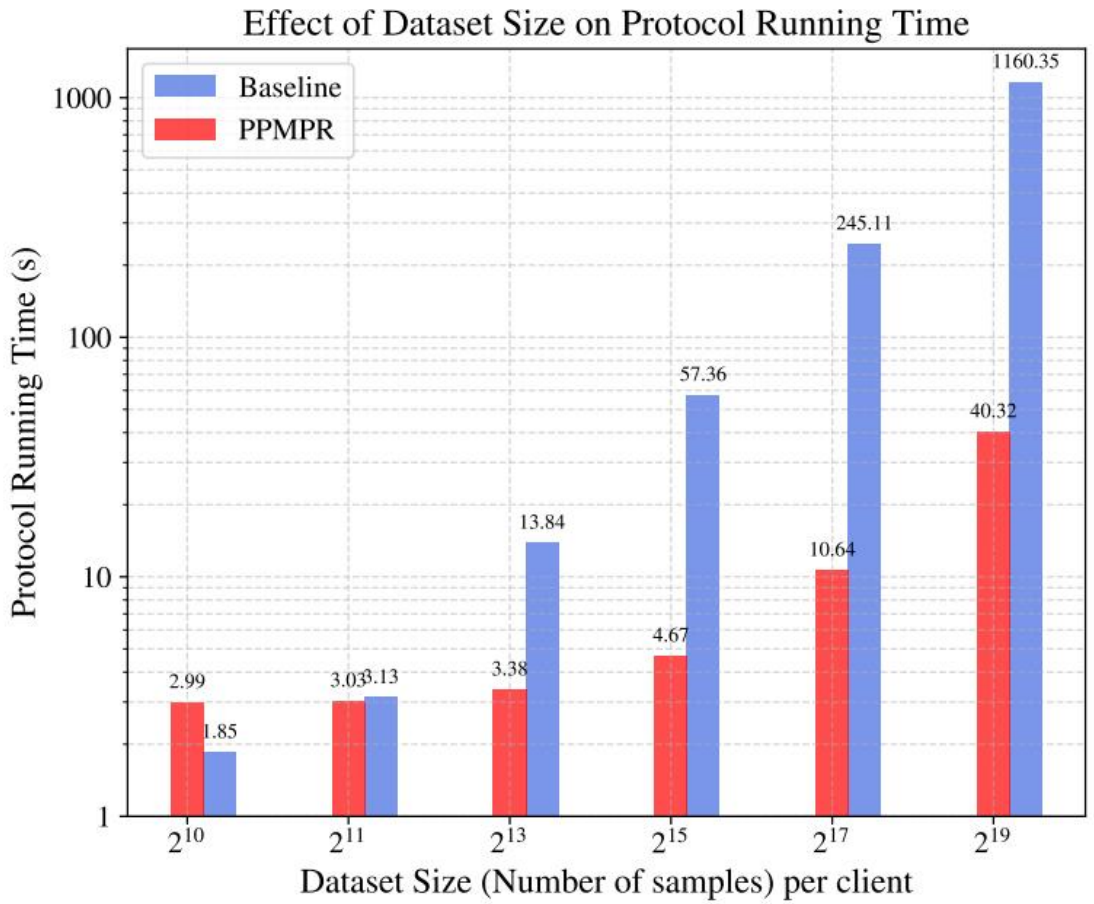
Enhanced Training

- Frequency-based loss reweighting

$$\vec{\mathcal{W}} = \frac{1}{\ln(\vec{\mathcal{C}} + \vec{1}) + \vec{\varepsilon}} \qquad \mathcal{L}_{\text{batch}} = \frac{\sum_{i=1}^{B} \vec{\mathcal{W}}_i \cdot \ell_i^{(t)}}{\sum_{i=1}^{B} \vec{\mathcal{W}}_i}$$

Evaluation: Preprocessing (28.78×)





- For clients (10-50) with 2^{19} data per client and 30% duplication, PPMPR exhibits 17.61-28.78 $\times$ speedup.
- For 50 clients, PPMPR outperforms the baseline by 4.09-28.78 $\times$ with increasing dataset size.

Evaluation: Model Performance (+11.42%)

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

Method	**Dataset**											
Duplication	Haiku			Rotten Tomatoes			Short Jokes			Sonnets		
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**

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**

Evaluation: Model Performance

- Evaluation on mainstream models

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**

Evaluation: Model Performance

- Evaluation on larger 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**

Evaluation: Model Performance

- Evaluation on Non-IID settings

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**

- Quantity & Label Skew: we categorize 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.
- 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.

Thank you for listening!