Federated Learning for Semantic Parsing: Task Formulation, Evaluation Setup, New Algorithms

Tianshu Zhang[1]*, **Changchang Liu**[2], **Wei-Han Lee**[2], **Yu Su**[1], **Huan Sun**[1]
[1]The Ohio State University
[2]IBM Research
[1]{zhang.11535, su.809, sun.397}@osu.edu
[2]{changchang.liu33, wei-han.lee1}@ibm.com

Abstract

This paper studies a new task of federated learning (FL) for semantic parsing, where multiple clients collaboratively train one global model without sharing their semantic parsing data. By leveraging data from multiple clients, the FL paradigm can be especially beneficial for clients that have little training data to develop a data-hungry neural semantic parser on their own. We propose an evaluation setup to study this task, where we re-purpose widely-used single-domain text-to-SQL datasets as clients to form a realistic heterogeneous FL setting and collaboratively train a global model. As standard FL algorithms suffer from the high client heterogeneity in our realistic setup, we further propose a novel **LO**ss **R**eduction **A**djusted **R**e-weighting (`Lorar`) mechanism to mitigate the performance degradation, which adjusts each client's contribution to the global model update based on its training loss reduction during each round. Our intuition is that the larger the loss reduction, the further away the current global model is from the client's local optimum, and the larger weight the client should get. By applying `Lorar` to three widely adopted FL algorithms (FedAvg, FedOPT and FedProx), we observe that their performance can be improved substantially on average (4%-20% absolute gain under MacroAvg) and that clients with smaller datasets enjoy larger performance gains. In addition, the global model converges faster for almost all the clients.[1]

1 Introduction

Semantic parsing aims to translate natural language utterances into formal meaning representations such as SQL queries and API calls and can be applied to build natural language interfaces that enable users to query data and invoke services without programming (Berant et al., 2013; Thomason et al., 2015; Su et al., 2017; Campagna et al., 2017). Neural semantic parsers have achieved remarkable performance in recent years (Wang et al., 2020a; Rubin and Berant, 2021; Scholak et al., 2021). However, they are data-hungry; bootstrapping a neural semantic parser by annotating data on a large scale can be very challenging for many institutions, as it requires the annotators to have intimate knowledge of formal programs. One natural thought is to leverage data from different institutions and train a unified model that can be used for all institutions. However, in practice, institutions such as hospitals, banks, and legal firms are prohibited from sharing their data with others, due to privacy concerns. Therefore, for institutions that only have very limited data, it is extremely hard to build their own neural semantic parsers.

Federated learning (FL) (Konečnỳ et al., 2016; McMahan et al., 2017; Yang et al., 2018) has turned out to be a popular training paradigm where multiple clients can collaboratively train a global model without exchanging their own data. In this paper, we study a new task of federated learning for semantic parsing. Through FL on the data scattered on different clients (e.g., institutions), we aim to obtain a global model that works well for all clients, especially those that have insufficient data to build their own neural models.

Towards that end, we propose an evaluation setup by re-purposing eight existing datasets that are widely adopted for text-to-SQL parsing, such as ATIS (Srinivasan Iyer and Zettlemoyer, 2017) and Yelp (Navid Yaghmazadeh and Dillig, 2017). These datasets demonstrate great heterogeneity, in terms of dataset sizes, language usage, database structures, and SQL complexity, as they were collected from the real life by different researchers, at different times, and for different purposes. Therefore, we use this collection to simulate a realistic scenario where eight clients with very different

*Work started during the internship at IBM T. J. Watson Research Center and continued at OSU.

[1]Our code and data are publicly available at https://github.com/OSU-NLP-Group/FL4SemanticParsing

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 12149–12163
July 9-14, 2023

data participate in the FL paradigm to jointly train a neural semantic parser.

Heterogeneity, where the data distributions and dataset sizes on different clients are different, is recognized as one of the biggest challenges in FL (McMahan et al., 2017; Reddi et al., 2020; Li et al., 2020a, 2021; Shoham et al., 2019; T Dinh et al., 2020). Existing work either uses synthetic data (Li et al., 2020a) or splits a classification dataset based on Dirichlet distribution (Lin et al., 2022) to simulate the non-IID federated learning setting, while we propose a more realistic setup to study this setting for semantic parsing. Pre-trained language models such as T5 (Raffel et al., 2020) have been shown as a powerful unified model for various semantic parsing tasks (Xie et al., 2022; Rajkumar et al., 2022), which can be leveraged to save us the efforts for client-specific model designs. Specifically, we adopt T5-base as our backbone semantic parser in the FL paradigm, and conduct extensive experiments and analysis using three widely-adopted FL algorithms: FedAvg (McMahan et al., 2017), FedOPT (Reddi et al., 2020) and FedProx (Li et al., 2020a).

As standard FL algorithms suffer from the high client heterogeneity in our realistic setup, we further propose a novel re-weighting mechanism for combining the gradient updates from each client during the global model update. The high-level idea is shown in Figure 1. Our intuition is that, for each client, the reduction of training loss during each round can signalize how far the current global model is away from the local optimum. By giving larger weights to those clients that have larger training loss reduction, the global model update can accommodate those clients better, thus mitigating potential performance degradation caused by high heterogeneity. We formulate this intuition as a re-weighting factor to adjust how much each client should contribute to the global model update during each round. Our proposed mechanism can be applied to all the three FL algorithms and experiments show that it can substantially improve both their parsing performance and their convergence speed, despite being very simple.

In summary, our main contributions are:

- To the best of our knowledge, we are the first to study federated learning for semantic parsing, a promising paradigm for multiple institutions to collaboratively build natural language interfaces without data sharing, which is especially beneficial for institutions with little training data.
- We propose an evaluation setup to simulate a realistic heterogeneous FL setting where different participating institutions have very different data. We re-purpose eight single-domain text-to-SQL datasets as eight clients, which demonstrate high heterogeneity in terms of dataset sizes, language usage, database structures, and SQL complexity.
- We propose a novel re-weighting mechanism, which uses the training loss reduction of each client to adjust its contribution to the global model update during each round. Experiments show that our re-weighting mechanism can substantially improve the model performance of existing FL algorithms on average, and clients with smaller training data observe larger performance gains. We discuss the limitations of our work and encourage future work to further study this task.

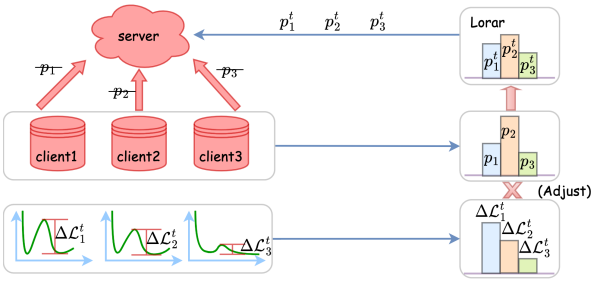

Figure 1: Our proposed re-weighting mechanism `Lorar` for the global model update in each round. The weight for each client (i.e., its contribution to the global model update) will be adjusted based on its loss reduction in each round. $\Delta\mathcal{L}_i^t$ means the training loss reduction of the *i-th* client in the *t-th* round.

2 Motivation and Task Formulation

Semantic parsing aims to translate natural language utterances into formal meaning representations and has numerous applications in building natural language interfaces that enable users to query data and invoke services without programming. As many institutions often lack data to develop neural semantic parsers by themselves, we propose a federated learning paradigm, where clients (i.e., "institutions") collaboratively train a global semantic parsing model without sharing their data.

There are two realistic settings of FL: cross-silo setting and cross-device setting (Kairouz et al.,

	Domain	Train	Dev	Test	SQL Pattern count	Questions / unique query count	Unique tables / query		SELECTs / query	
							μ	Max	μ	Max
Advising	Course Infomation	2629	229	573	174	21.7	3.0	9	1.23	6
ATIS	Flight Booking	4347	486	447	751	5.6	3.8	12	1.79	8
GeoQuery	US Geography	549	49	279	98	3.6	1.1	4	1.77	8
Restaurants	Restaurants/Food	228	76	74	17	16.4	2.3	4	1.17	2
Scholar	Academic Publication	499	100	218	146	4.2	3.2	6	1.02	2
Academic	Microsoft Academic	120	38	38	92	1.1	3	6	1.04	3
IMDB	Internet Movie	78	26	26	52	1.5	1.9	5	1.01	2
Yelp	Yelp Website	78	26	24	89	1.2	2	4	1	1

Table 1: Statistics for the heterogeneous text-to-SQL datasets. "μ": the average number under the measure. "Max": the max number under the measure.

2021; Lin et al., 2022). For the cross-silo setting, clients are large institutions, such as hospitals and companies, and the number of clients is limited in this setting. In general, they have large computational resources and storage to train and store a large model, and large communication costs between the server and clients are tolerated. For the cross-device setting, clients are small devices such as mobile phones and Raspberry Pis, thus there may exist a huge number of clients. They have limited computational resources and storage and only small communication costs between the server and clients are affordable. Here our FL for semantic parsing can be regarded as a cross-silo setting, where each client is a relatively large institution that hopes to build a natural language interface based on its user utterances and underlying data. Studying FL for semantic parsing under a cross-device setting could be interesting future work.

3 Evaluation Setup

As we are the first to study cross-silo FL for semantic parsing, there is no benchmark for this task. Thus we establish an evaluation setup by re-purposing eight single-domain text-to-SQL datasets (Finegan-Dollak et al., 2018) as eight "clients", which demonstrate high heterogeneity in terms of dataset sizes, domains, language usage, database structures and SQL complexity. Table 1 shows their statistics.

Given a natural language question and the database schema, text-to-SQL parsing aims to generate a SQL query. Here the question is a sequence of tokens and the database schema consists of multiple tables with each table containing multiple columns. Figure 7 in Appendix shows an example of this task. We adopt T5-base as our backbone model, which has been shown as an effective unified model for various semantic parsing tasks (Xie et al., 2022). Similarly as in previous work (Xie et al., 2022), we concatenate the question tokens with the serialized relational table schemas (table names and column names) as the model input and output a sequence of SQL tokens.

The heterogeneity of the eight clients is described in detail from the following perspectives.

Domain: The clients are from diverse domains. Some clients such as Scholar and Academic are from closer domains than others.

Dataset Size: The clients differ significantly in terms of dataset sizes. Here, we consider datasets with more than 1000 train examples as *large-sized* datasets, with 200∼1000 as *medium-sized* datasets, and with less than 200 as *small-sized* datasets. In our setup, we have 2 large-sized clients (Advising and ATIS), 3 medium-sized clients (Geoquery, Restaurants and Scholar), and 3 small-sized clients (Academic, IMDB and Yelp).

Diversity: "SQL pattern count" shows the number of SQL patterns in the full dataset. The patterns are abstracted from the SQL queries with specific table names, column names and variables anonymized. A larger value under this measure indicates greater diversity. In our benchmark, Advising, ATIS and Scholar have larger diversity than the other datasets.

Redundancy: "Questions per unique SQL query" counts how many natural language questions can be translated into the same SQL query (where variables are anonymized). A larger value indicates higher redundancy in the dataset. Intuitively, the higher the redundancy, the more easily a model can make correct predictions. In our benchmark, the redundancy for Advising and Restaurants

is higher than the other datasets.

Complexity: "Unique tables per SQL query" (where variables in the SQL query are anonymized) represents how many unique tables are mentioned in one query. "SELECTs per query" counts how many SELECT clauses are included in one query. The larger these two measures, the more complex the dataset is and the more difficult for a model to make predictions. In our benchmark, Advising and ATIS are more complex.

4 FL for Semantic Parsing

In this section, we first introduce the background of FL, more specifically, its training objective, training procedure and three widely adopted FL algorithms. Then we describe the motivating insights and details of our proposed mechanism.

4.1 Background

Training Objective. Federated learning aims to optimize the following objective function:

$$\min_{w} \mathcal{F}(w) := \sum_{i=1}^{N} p_i \mathcal{L}_i(w) \quad where \quad \mathcal{L}_i(w) = \mathbb{E}_{b \sim \mathcal{D}_i}[f_i(w, b)]. \tag{1}$$

In Eqn. (1), $\mathcal{L}_i(w)$ denotes the local training objective function of the client i and N denotes the number of clients. $w \in \mathbb{R}^d$ represents the parameters of the global model. b denotes each batch of data. The local training loss function $f_i(w, b)$ is often the same across all the clients, while $\mathcal{D}_i$ denotes the distribution of the local client data, which is often different across the clients, capturing the heterogeneity. p_i is defined as the training size proportion in Eqn. (2), where $|\mathcal{D}_i|$ is the training size of client i.

$$p_i = |\mathcal{D}_i| \,/\, \sum_{i=1}^{N} |\mathcal{D}_i| \tag{2}$$

Training Procedure. Federated learning is an iterative process shown in Figure 2. The server initializes the global model, followed by multiple communication rounds between the server and clients. In each *communication round*, there are four steps between the server and clients. 1) In round t, the server sends the global model w^t to all the clients. 2) After clients receive the global model w^t as the initialization of the local model, they start to train it using their own data for multiple epochs and obtain the local model changes Δw_i^t during the local training stage. 3) The clients send their local model changes to the server. 4) The server aggregates the local model changes Δw_i^t collected from different clients as Eqn. (3) shows, and then uses the t-th round's global model w^t and the aggregated local model changes Δw^t to update the global model. As Eqn. (4) shows, w^{t+1} is the global model after the update. Here, η denotes the server learning rate. The server will send the updated model w^{t+1} to the clients, then the (t+1)-th round starts.

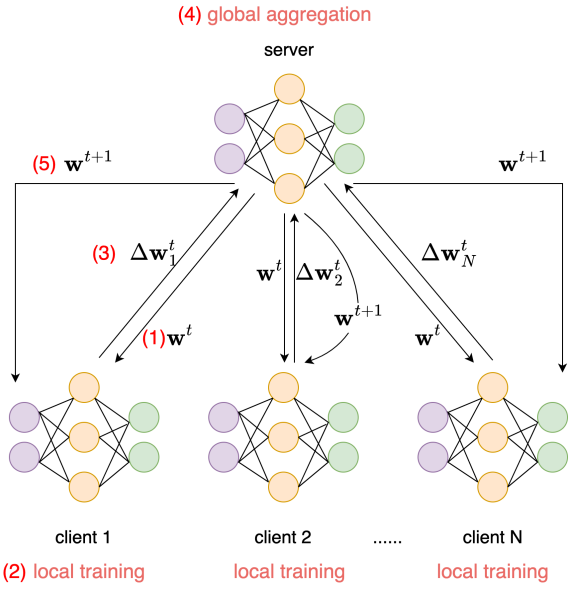

Figure 2: An overview of the FL procedure.

The above procedure will repeat until the algorithm converges.

$$\Delta w^t = \sum_{i=1}^{N} p_i \Delta w_i^t \tag{3}$$

$$w^{t+1} = w^t - \eta \Delta w^t \tag{4}$$

FL Algorithms. We explore three popular FL algorithms for our task:

Federated Averaging (FedAvg) (McMahan et al., 2017) uses stochastic gradient descent (SGD) as the local training optimizer to optimize the training procedure and uses the same learning rate and the same number of local training epochs for all the clients.

FedOPT (Reddi et al., 2020) is a generalized version of FedAvg. The algorithm is parameterized by two gradient-based optimizers: CLIENTOPT and SERVEROPT. CLIENTOPT is used to update the local models on the client side, while SERVEROPT treats the negative of aggregated local changes "$-\Delta w^t$" as a pseudo-gradient and applies it to the global model on the server side. FedOPT allows powerful adaptive optimizers on both

server side and client side.

FedProx (Li et al., 2020a) tries to tackle the statistical heterogeneity issue by adding an L2 regularization term, which constrains the local model to be closer to the local model initialization (i.e., the global model) during each round for stable training.

To summarize, for the local training stage, both FedAvg and FedOPT optimize the local training objective $f_i(w, b)$; for FedProx, it optimizes Eqn. (5), where μ is a hyperparameter and w^t is the local model initialization (i.e., the global model) during the t-th round.

$$\min_w h_i(w, b, w^t) := f_i(w, b) + \frac{\mu}{2}\|w - w^t\|^2 \tag{5}$$

For the cross-silo setting where all clients participate in training for each round, these three algorithms optimize Eqn. (1) during the FL process.

4.2 Our Proposed Re-weighting Mechanism

Motivating Insights. Heterogeneity, where the data distributions and dataset sizes on different clients are different, is recognized as one of the biggest challenges in FL, which usually leads to performance degradation for clients. Here, we uniquely observe the clients' heterogeneity from the perspective of their training loss reduction. Take Restaurants and Yelp as two example clients. Figure 3 shows their training loss variation w.r.t. "Step". Here the "Step" is the number of iteration steps for each client during training. Adjacent high and low points in the figure correspond to one communication round. When the curve goes down, it means the client is in the local training stage. When the curve goes up, it means the server has updated the global model based on the aggregated local model changes from all clients and each client starts a new round of local training with the updated global model as the local model initialization. Since for different clients, the dataset sizes and the local training epochs are different, for the same communication round, the "Step" for different clients is different.

As we can see, after each round, the global model deviates from the optimization trajectory of each client. Thus the reduction of the training loss can signalize how far the global model is away from the client's local optimum. As p_i decides how much each client contributes to the global model update, we give larger weights to those clients who have larger training loss reduction to make the global model update accommodate them better, thus mitigating potential performance degradation caused by high heterogeneity.

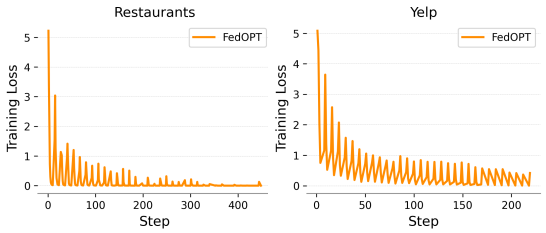

Figure 3: Training loss variation of two clients: Restaurants and Yelp. The reduction of training loss during each round can signalize how far the global model is away from the client's local optimum.

Proposed Mechanism. Based on the above insights, we use the training loss reduction to adjust the weight of each client, so as to reschedule its contribution to the global model update. The final weight is formulated as Eqn. (6), where $\Delta\mathcal{L}_i^t$ is the training loss reduction during the t-th round.

$$p_i^t = |\mathcal{D}_i|\,\Delta\mathcal{L}_i^t / \sum_{i=1}^{N} |\mathcal{D}_i|\,\Delta\mathcal{L}_i^t \tag{6}$$

FedAvg, FedOPT, FedreProx with our proposed mechanism are summarized in Algorithm 1 in Appendix.

5 Experiments

Datasets. We re-purpose eight datasets: ATIS (Srinivasan Iyer and Zettlemoyer, 2017; Deborah A. Dahl and Shriber, 1994), GeoQuery (Srinivasan Iyer and Zettlemoyer, 2017; Zelle and Mooney, 1996), Restaurants (Tang and Mooney, 2000; Ana-Maria Popescu and Kautz, 2003; Giordani and Moschitti, 2012), Scholar (Srinivasan Iyer and Zettlemoyer, 2017), Academic (Li and Jagadish, 2014), Advising (Finegan-Dollak et al., 2018), Yelp and IMDB (Navid Yaghmazadeh and Dillig, 2017) as eight clients. These datasets have been standardized to the same SQL style by Finegan-Dollak et al. (2018). Their characteristics have been described in Section 3. We follow "question split" datasets preprocessed by Finegan-Dollak et al. (2018) to split the train, dev and test data, which means we let the train, dev and test examples have different questions but the same SQL queries are allowed. For Advising, ATIS, GeoQuery and Scholar, we directly use the original question split as our split. For Restaurants, Academic, IMDB and Yelp, since the data sizes are

relatively small, the original question split uses 10 splits for cross validation without specifying train, dev and test examples. Given FL is costly as we need multiple GPUs to finish one experiment, we fix the train, dev and test set by randomly selecting 6 splits as the train set, 2 splits as the dev set and 2 splits as the test set.

Evaluation Metrics. 1) Exact Match (EM): a prediction is deemed correct only if it is exactly the same as the ground truth (i.e., exact string match), which is widely used for text-to-SQL parsing (Finegan-Dollak et al., 2018). All the evaluations in our experiments consider the values generated in the SQL query. 2) MacroAvg: The arithmetic mean of EM across all clients, which treats each client equally. 3) MicroAvg: The total number of correct predictions on all the clients divided by the total test examples, which treats each test example equally.

Learning Paradigm. We compare three learning paradigms: finetuning, centralized and FL. 1) *Finetuning*: we individually finetune our backbone model (T5-base) on the training data of each client. 2) *Centralized*: we merge the training data of all the clients and finetune our backbone model on the merged training data to obtain one model. 3) *FL*: we leverage eight clients and a server to learn a global model without sharing each client's local data. By comparing individual finetuning and FL, we can show the benefit of FL for some clients, especially for small-sized clients. The centralized paradigm is less practical compared with the other two paradigms due to privacy considerations. However, it can serve as a useful reference to help validate how effective an FL algorithm is in fully exploiting heterogeneous data across multiple clients.

Implementation Details. We implement the FL algorithms and T5-base model based on FedNLP (Lin et al., 2022), FedML (He et al., 2020) and UnifiedSKG (Xie et al., 2022). We use Adafactor (Shazeer and Stern, 2018) as the optimizer for finetuning and centralized paradigms, and as the client optimizer[2] for FL paradigm, since it has been shown as the best optimizer to optimize the T5 model. More details are in Appendix A.1.

For the computing resources, we use 1 NVIDIA A6000 48GB GPU for finetuning, with batch size 8. We use 2 NVIDIA A6000 48GB GPUs for centralized training, with batch size 8. We use 5 NVIDIA A6000 48GB GPUs for all federated learning experiments. Specifically, one GPU is used as the server and the other four GPUs are used as 8 clients, with each GPU accommodating 2 clients. The batch size for clients GeoQuery, Restaurants, Scholar, Academic, IMDB and Yelp is 4, and for clients Advising and ATIS is 8.

[2]Note we use Adafactor as the local optimizer for FedAvg, so the FedAvg in our paper is slightly different from the original proposed FedAvg, which uses stochastic gradient descent(SGD) as the local optimizer.

6 Results and Analysis

6.1 Main Results

Centralized vs. Finetuning. As Table 2 shows, compared with the individual finetuning setting, the model performance under the centralized setting has been improved on all the datasets except Scholar. *This means merging all the data to train a model, which increases the size and diversity of training data, can improve the model's generalization ability and lead to improvement for most datasets.* This observation also motivates us to leverage these datasets to study FL for semantic parsing, which is a more practical paradigm than the centralized one.

Effectiveness of `Lorar` in FL. Applying our proposed `Lorar` mechanism can substantially improve the performance of all three FL algorithms overall. As Table 2 shows, for FedOPT, our proposed FedOPT$_{\texttt{lorar}}$ performs substantially better or similarly on all clients, except for a slight drop on GeoQuery and Scholar. Moreover, on the three smaller datasets: Academic, IMDB and Yelp, `Lorar` brings much larger performance gains. For FedAvg and FedProx, in addition to these three datasets, `Lorar` also brings substantial improvements on two medium-sized clients: Restaurants and Scholar. These observations validate the effectiveness of our proposed mechanism under different FL algorithms and across different clients.

We additionally analyze these three FL algorithms and their performance variation with and without using `Lorar` under different communication rounds. More details are included in Appendix A.2 and A.3.

FL vs. Finetuning/Centralized. As Table 2 shows, the original FedOPT outperforms finetuning on GeoQuery and IMDB, which shows that FL can boost the model performance for some clients. In addition, although there is still a gap between existing FL algorithms (FedOPT, FedAvg, and FedProx) and the centralized setting, by equipping them with our proposed `Lorar`, we can reduce the

	Advising†	ATIS†	GeoQuery§	Restaurants§	Scholar§	Academic*	IMDB*	Yelp*	MacroAvg	MicroAvg
Finetuning	84.47	53.91	72.76	98.65	74.31	57.89	26.92	33.33	62.78	71.47
Centralized	85.51	56.38	79.21	100	72.48	65.79	61.54	41.67	70.32	74.21
FedOPT	79.76	51.23	**77.42**	**98.65**	**66.51**	50	34.62	8.33	58.32	68.49
FedOPT$_{lorar}$	**80.98**	**52.35**	75.99	**98.65**	64.68	**68.42**	**38.46**	**20.83**	**62.55**	**69.39**
FedAvg	**76.44**	**50.11**	59.86	72.97	38.07	2.63	7.69	12.5	40.03	57.89
FedAvg$_{lorar}$	74.69	49.89	**68.82**	**98.65**	**52.29**	**65.79**	**46.15**	**25**	**60.16**	**63.91**
FedProx	**74.52**	**50.56**	65.95	81.08	38.53	10.53	3.85	8.33	41.67	58.84
FedProx$_{lorar}$	73.12	49.66	**67.38**	**98.65**	**48.17**	**63.16**	**46.15**	**20.83**	**58.39**	**62.42**

Table 2: Main results for different learning paradigms and FL algorithms. "†": large-sized clients. "§": medium-sized clients. "*": small-sized clients.

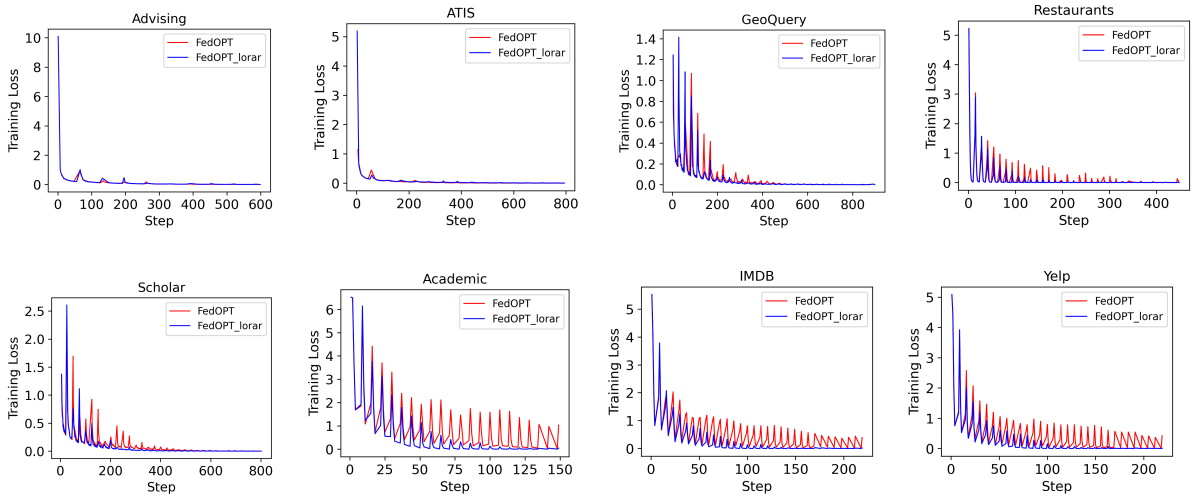

Figure 4: Training loss variation on eight clients for FedOPT and FedOPT$_{\texttt{lorar}}$.

gap by 4-20 points (i.e., absolute difference under MacroAvg). It is worth noting that institutions are often reluctant or prohibited to share their data in practice, especially for SQL data that may directly reveal private database content. Therefore, the centralized paradigm is impractical. Nonetheless, it can serve as a useful reference to help validate how effective an FL algorithm is in fully exploiting heterogeneous data across multiple clients. The results show that our benchmark provides a challenging testbed for a realistic FL problem, and there is still a large room to further improve the FL algorithms.

6.2 Training Loss Analysis

To better understand how `Lorar` affects the training process in FL, we show the training loss variation for FedOPT and FedOPT$_{\texttt{lorar}}$ in Figure 4. For FedOPT, we can see for larger datasets such as Advising and ATIS, the training converges much faster and the global model is closer to the client's local optimum within very few rounds. While for smaller datasets such as Academic, IMDB and Yelp, the training loss oscillates widely, which means the global model converges slower for these clients (if at all). After applying `Lorar`, however, *the training loss converges faster on almost all the clients*, which means the global model can get close to the client's local optimum more quickly and easily.

6.3 Alternative Weighting Mechanisms

As FedOPT performs best among all three FL baselines, we use it to compare `Lorar` with alternative weighting mechanisms. As Table 3 shows, `Lorar`, which considers both the training set size and the loss reduction in the weight, can achieve the best results. Comparing FedOPT$_{lr}$ (i.e., FedOPT with only loss reduction considered in the weight) and FedOPT$_{\texttt{lorar}}$, we can see removing the training set size from the weight will lead to a large drop under MacroAvg and MicroAvg, which indicates that training set size is an important factor during the aggregation. This is intuitive since for those clients which have more training data, their local models tend to be more reliable and more general-

	Advising	ATIS	GeoQuery	Restaurants	Scholar	Academic	IMDB	Yelp	MacroAvg	MicroAvg
FedOPT	79.76	51.23	77.42	**98.65**	**66.51**	50	34.62	8.33	58.32	68.49
FedOPT$_{lr}$	75.04	**53.47**	75.63	**98.65**	62.39	60.53	34.62	**25**	60.67	67.12
FedOPT$_{equal}$	76.96	53.02	**77.78**	**98.65**	63.3	63.16	34.62	20.83	61.04	68.13
FedOPT$_{lorar}$	**80.98**	52.35	75.99	**98.65**	64.68	**68.42**	**38.46**	20.83	**62.55**	**69.39**

Table 3: Alternative weighting mechanisms for FedOPT on the test set of our proposed benchmark, where FedOPT only uses a client's training set size (w/o loss reduction) as its weight, FedOPT$_{lr}$ only uses a client's loss reduction during each round (w/o train set size) as its weight, FedOPT$_{\texttt{lorar}}$ considers both factors as its weight (Eqn. (6)) and FedOPT$_{equal}$ gives each client equal weight (w/o considering both factors).

izable. We also compare with FedOPT$_{equal}$ where all clients are given the same weight. We can see that our FedOPT$_{\texttt{lorar}}$ yields superior performance. The conclusion can also be verified in Figure 6 in Appendix, where we show their performance variation under different communication rounds.

6.4 Impact from Dataset Heterogeneity

(1) The impact of diversity, redundancy and complexity: In Table 2 and 3, for Restaurants, the results of finetuning, centralized training, and varying weighting mechanisms of FedOPT are pretty close and all very high (close to 100%), which shows it is a relatively easy dataset for any learning paradigm and weighting mechanism. Looking at Table 1, Restaurants has the smallest "SQL pattern count" (i.e., lowest diversity), second largest "Questions per unique SQL query" (i.e., second highest redundancy), close to the smallest "Unique tables per query" and "SELECTs per query" (i.e., close to lowest complexity), which makes models easily learn from this dataset (Section 3). For other datasets, they have higher diversity, lower redundancy, or higher complexity, which makes models harder to make predictions and the performance is generally lower than Restaurants. (2) The impact of dataset size: Smaller datasets tend to have lower performance, as shown in Table 2, which means they are harder to learn in general due to lack of data; however, they can benefit more from our proposed FL paradigm.

7 Related Work

Text-to-SQL. Text-to-SQL problem which translates natural language questions to SQL queries has been studied for many years. There have been several single-database text-to-SQL datasets such as Geoquery (Srinivasan Iyer and Zettlemoyer, 2017) and ATIS (Srinivasan Iyer and Zettlemoyer, 2017), which map from natural language questions to SQL queries on a single database. Finegan-Dollak et al., 2018 curate eight datasets to unify their SQL format. These datasets cover a variety of domains and have different characteristics of the tables and SQL, which provide us a foundation to study the heterogeneous FL for the text-to-SQL problem.

One line of work designs special models for the text-to-SQL task such as designing a relation-aware self-attention mechanism for the Transformer model to better encode the relation of the column mappings (Wang et al., 2020a) or adding constraints to the decoder (Scholak et al., 2021) to generate valid SQL queries, while another line of work tries to directly finetune a pre-trained language model such as T5 (Xie et al., 2022; Raffel et al., 2020; Rajkumar et al., 2022). As directly finetuning T5 has shown great performance and allows us to use a unified model architecture for all clients and the server, we choose T5-base as the backbone model in our work.

Heterogeneity in Federated Learning. Heterogeneity is one of the major challenges in federated learning. Existing work (McMahan et al., 2017; Reddi et al., 2020; Li et al., 2020a, 2021; Shoham et al., 2019; T Dinh et al., 2020; Li et al., 2022) shows that heterogeneity can cause performance degradation. Several methods have been proposed to address this issue. For instance, FedOPT (Reddi et al., 2020) uses powerful adaptive optimization methods for both the server and clients, while FedProx (Li et al., 2020a) (and pFedMe (T Dinh et al., 2020)) regularizes the local training procedure. However, based on our observations in Section 6.1, our mechanism significantly outperforms these methods. Other work that aims to address the heterogeneity issue in FL includes FedNova (Wang et al., 2020b) and Li et al., 2020b. Specifically, FedNova (Wang et al., 2020b) uses the local training update steps to normalize the server aggregation, and Li et al., 2020b proposes to optimize the power-scaled training objective. Compared to FedNova, we use a more direct indicator, training

loss reduction, to adjust the weight for each client during aggregation. Different from Li et al., 2020b, our proposed simple yet effective mechanism does not require modification of the local client optimization step or additional tuning of any related hyperparameter.

8 Conclusions

To the best of our knowledge, we are the first to study federated learning for semantic parsing. Specifically, we propose a realistic benchmark by re-purposing eight single-domain text-to-SQL datasets. Moreover, we propose a novel loss reduction adjusted re-weighting mechanism (`Lorar`) that is applicable to widely adopted FL algorithms. By applying `Lorar` to FedAvg, FedOPT and FedProx, we observe their performance can be improved substantially on average, and clients with smaller datasets enjoy larger performance gains.

Limitations

In this work, we address the heterogeneity challenge in the task of FL for semantic parsing, by leveraging the reduction of training loss signal. Our work is motivated from the FL training procedure perspective to adjust the contribution of each client during the global model aggregation stage, but how each client's data contribute to the final global model is still unclear. As the data of different clients contain different information, what kind of information of each client is helpful and can be more directly linked and utilized to facilitate the FL training is worth more efforts in future work.

In addition, our proposed re-weighting mechanism is a universal technique for cross-silo FL. Thus generalizing our proposed re-weighting mechanism to a broader range of tasks beyond semantic parsing, and further studying under what kind of conditions, `Lorar` can make a huge difference for FL would be interesting future work to pursue.

Acknowledgements

The authors would like to thank colleagues from the OSU NLP group and all anonymous reviewers for their thoughtful comments. This research was supported in part by NSF OAC 2112606, NSF IIS 1815674, NSF CAREER 1942980, and Ohio Supercomputer Center (Center, 1987). The work done at IBM research was sponsored by the Combat Capabilities Development Command Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-13-2-0045 (ARL Cyber Security CRA). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Combat Capabilities Development Command Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on. We thank Chaoyang He for his help during reproducing FedNLP. We thank Wei-Lun (Harry) Chao for valuable discussion.

References

Oren Etzioni Ana-Maria Popescu and Henry Kautz. 2003. Towards a theory of natural language interfaces to databases. In *Proceedings of the 8th International Conference on Intelligent User Interfaces*, pages 149–157.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544, Seattle, Washington, USA. Association for Computational Linguistics.

Giovanni Campagna, Rakesh Ramesh, Silei Xu, Michael Fischer, and Monica S Lam. 2017. Almond: The architecture of an open, crowdsourced, privacy-preserving, programmable virtual assistant. In *Proceedings of the 26th International Conference on World Wide Web*, pages 341–350.

Ohio Supercomputer Center. 1987. Ohio supercomputer center.

Michael Brown William Fisher Kate Hunicke-Smith David Pallett Christine Pao Alexander Rudnicky Deborah A. Dahl, Madeleine Bates and Elizabeth Shriber. 1994. Expanding the scope of the ATIS task: The ATIS-3 corpus. *Proceedings of the workshop on Human Language Technology*, pages 43–48.

Catherine Finegan-Dollak, Jonathan K. Kummerfeld, Li Zhang, Karthik Ramanathan, Sesh Sadasivam, Rui Zhang, and Dragomir Radev. 2018. Improving text-to-SQL evaluation methodology. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 351–360, Melbourne, Australia. Association for Computational Linguistics.

Alessandra Giordani and Alessandro Moschitti. 2012. Automatic generation and reranking of sql-derived answers to nl questions. In *Proceedings of the Second International Conference on Trustworthy Eternal Systems via Evolving Software, Data and Knowledge*, pages 59–76.

Chaoyang He, Songze Li, Jinhyun So, Mi Zhang, Hongyi Wang, Xiaoyang Wang, Praneeth Vepakomma, Abhishek Singh, Hang Qiu, Li Shen, Peilin Zhao, Yan Kang, Yang Liu, Ramesh Raskar, Qiang Yang, Murali Annavaram, and Salman Avestimehr. 2020. Fedml: A research library and benchmark for federated machine learning. *arXiv preprint arXiv:2007.13518*.

Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. 2021. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210.

Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. 2016. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*.

Fei Li and H. V. Jagadish. 2014. Constructing an interactive natural language interface for relational databases. *Proceedings of the VLDB Endowment*, 8(1):73–84.

Qinbin Li, Yiqun Diao, Quan Chen, and Bingsheng He. 2022. Federated learning on non-iid data silos: An experimental study. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pages 965–978. IEEE.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2020a. Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems*, 2:429–450.

Tian Li, Maziar Sanjabi, Ahmad Beirami, and Virginia Smith. 2020b. Fair resource allocation in federated learning. In *International Conference on Learning Representations*.

Xiaoxiao Li, Meirui Jiang, Xiaofei Zhang, Michael Kamp, and Qi Dou. 2021. Fedbn: Federated learning on non-iid features via local batch normalization. In *International Conference on Learning Representations*.

Bill Yuchen Lin, Chaoyang He, Zihang Ze, Hulin Wang, Yufen Hua, Christophe Dupuy, Rahul Gupta, Mahdi Soltanolkotabi, Xiang Ren, and Salman Avestimehr. 2022. FedNLP: Benchmarking federated learning methods for natural language processing tasks. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 157–175, Seattle, United States. Association for Computational Linguistics.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 1273–1282. PMLR.

Isil Dillig Navid Yaghmazadeh, Yuepeng Wang and Thomas Dillig. 2017. Sqlizer: Query synthesis from natural language. In *International Conference on Object-Oriented Programming, Systems, Languages, and Applications, ACM*, pages 63:1–63:26.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Nitarshan Rajkumar, Raymond Li, and Dzmitry Bahdanau. 2022. Evaluating the text-to-sql capabilities of large language models. *arXiv preprint arXiv:2204.00498*.

Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and H Brendan McMahan. 2020. Adaptive federated optimization. In *International Conference on Learning Representations*.

Ohad Rubin and Jonathan Berant. 2021. SmBoP: Semi-autoregressive bottom-up semantic parsing. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 311–324, Online. Association for Computational Linguistics.

Torsten Scholak, Nathan Schucher, and Dzmitry Bahdanau. 2021. PICARD: Parsing incrementally for constrained auto-regressive decoding from language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9895–9901. Association for Computational Linguistics.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4596–4604. PMLR.

Neta Shoham, Tomer Avidor, Aviv Keren, Nadav Israel, Daniel Benditkis, Liron Mor-Yosef, and Itai Zeitak. 2019. Overcoming forgetting in federated learning on non-iid data. *CoRR*, abs/1910.07796.

Alvin Cheung Jayant Krishnamurthy Srinivasan Iyer, Ioannis Konstas and Luke Zettlemoyer. 2017. Learning a neural semantic parser from user feedback. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 963–973.

Yu Su, Ahmed Hassan Awadallah, Madian Khabsa, Patrick Pantel, Michael Gamon, and Mark Encarnacion. 2017. Building natural language interfaces to web apis. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 177–186.

Canh T Dinh, Nguyen Tran, and Josh Nguyen. 2020. Personalized federated learning with moreau envelopes. *Advances in Neural Information Processing Systems*, 33:21394–21405.

Lappoon R. Tang and Raymond J. Mooney. 2000. Automated construction of database interfaces: Intergrating statistical and relational learning for semantic parsing. In *2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 133–141.

Jesse Thomason, Shiqi Zhang, Raymond J Mooney, and Peter Stone. 2015. Learning to interpret natural language commands through human-robot dialog. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

Bailin Wang, Richard Shin, Xiaodong Liu, Oleksandr Polozov, and Matthew Richardson. 2020a. RAT-SQL: Relation-aware schema encoding and linking for text-to-SQL parsers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7567–7578, Online. Association for Computational Linguistics.

Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor. 2020b. Tackling the objective inconsistency problem in heterogeneous federated optimization. *Advances in neural information processing systems*, 33:7611–7623.

Tianbao Xie, Chen Henry Wu, Peng Shi, Ruiqi Zhong, Torsten Scholak, Michihiro Yasunaga, Chien-Sheng Wu, Ming Zhong, Pengcheng Yin, Sida I. Wang, Victor Zhong, Bailin Wang, Chengzu Li, Connor Boyle, Ansong Ni, Ziyu Yao, Dragomir Radev, Caiming Xiong, Lingpeng Kong, Rui Zhang, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2022. UnifiedSKG: Unifying and multi-tasking structured knowledge grounding with text-to-text language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 602–631, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Timothy Yang, Galen Andrew, Hubert Eichner, Haicheng Sun, Wei Li, Nicholas Kong, Daniel Ramage, and Françoise Beaufays. 2018. Applied federated learning: Improving google keyboard query suggestions. *arXiv preprint arXiv:1812.02903*.

John M Zelle and Raymond J Mooney. 1996. Learning to parse database queries using inductive logic programming. In *Proceedings of the national conference on artificial intelligence*, pages 1050–1055.

A Appendix

Algorithm 1:

Input: local datasets $\mathcal{D}_i$, number of communication rounds T, number of local epochs E, server learning rate η, client learning rate η_i

Output: the final global model w^T

Server executes:

for $t \in 0, 1, 2, ..., T$ **do**

  Sample a set of clients C_t[a]

  for $i \in C_t$ ***in parallel*** **do**

    Send the global model w^t to client i

    $\Delta w_i^t, |\mathcal{D}_i|\,\Delta\mathcal{L}_i^t \leftarrow$ **LocalTraining**(i, w^t)

  $\Delta w^t = \sum_{i \in C_t} p_i^t \Delta w_i^t$

  For FedOPT/FedAvg/FedProx: $p_i = |\mathcal{D}_i| \,/\, \sum_{i \in C_t} |\mathcal{D}_i|$

  For ours (`Lorar`): $p_i^t = |\mathcal{D}_i|\,\Delta\mathcal{L}_i^t / \sum_{i \in C_t} |\mathcal{D}_i|\,\Delta\mathcal{L}_i^t$

  $w^{t+1} \leftarrow \mathrm{w}^t - \eta \Delta w^t$

return w^T

Client executes:

FedAvg/FedOPT: $\mathcal{L}(w; b) = \sum_{(x,y) \in b} f(w; x; y)$

FedProx: $\mathcal{L}(w; b) = \sum_{(x,y) \in b} f(w; x; y) + \frac{\mu}{2}\|w - w^t\|^2$

LocalTraining(i, w^t)

$w_i^t \leftarrow w^t$

for *epoch* $k = 0, 1, 2, ..., E$ **do**

  for *each batch* $b = \{x, y\}$ *of* $\mathcal{D}_i$ **do**

    $w_i^t \leftarrow w_i^t - \eta_i \nabla \mathcal{L}_i^{t,k}(w_i^t; b)$

$\Delta w_i^t \leftarrow w^t - w_i^t$

$\Delta\mathcal{L}_i^t \leftarrow \max \mathcal{L}_i^t - \min \mathcal{L}_i^t$

return $\Delta w_i^t, |\mathcal{D}_i|\,\Delta\mathcal{L}_i^t$ to the server

[a] We use all clients in our experiments.

A.1 Implementation Details

We use T5-base (Raffel et al., 2020) as the model for text-to-SQL task in all three learning paradigms (finetuning, centralized and FL), as it has been shown as an effective unified model for various semantic parsing tasks in UnifiedSKG (Xie et al., 2022). For all three FL algorithms, we implement them based on FedNLP (Lin et al., 2022) and FedML (He et al., 2020). We use Adafactor (Shazeer and Stern, 2018) as the optimizer for

finetuning and centralized paradigms, and as the client optimizer[3] for FL paradigm, since it has been shown as the best optimizer to optimize for the T5 model.

For the FL paradigm, we tune hyperparameters for FedOPT, FedAvg and FedProx as follows. For FedOPT, we test all the combinations of the server learning rate from {0.001, 0.01 0.1, 0.5, 1} and {w/ 0.9, w/o} server momentum. We found 1 as the server learning rate and 0.9 as the server momentum is the best hyperparameter combination. For FedProx, we vary μ from {0.0001, 0.001, 0.01, 0.1, 1} and use the dev set to choose the best model. We finally choose the best hyperparameter 0.0001 in our experiment. For all the federated learning paradigms, we set local training epochs as 6 for two large datasets: ATIS and Advising. We set the local training epoch as 12 for all the other six datasets. We let all the clients participate in each round and we train the entire process for 60 rounds (which lasts around 60 hours). And we test the global model performance on the merged dev set for every 5 communication rounds to choose the best model. We use the best global model to evaluate on all eight test sets to get the global model performance on each client.

For the finetuning paradigm, we finetune T5-base on each dataset for a maximum of 200 epochs. We use the dev set of each client to choose the best model and then evaluate the model on each test set.

For the centralized paradigm, we merge all eight training sets and then finetune T5-base for a maximum of 200 epochs on the merged dataset to get one centralized model. We merge all eight dev sets and use the merged dev set to choose the best model. Then we evaluate the centralized model on each test set.

For all finetuning, centralized and federated learning paradigms, we set the input length as 1024 and the output length as 512. We try learning rate in {1e-5, 1e-4, 1e-3}. We finally choose 1e-4 for the centralized paradigm, and 1e-4 for Advising, ATIS, Geoquery and Yelp in the finetuning paradigm and FL paradigm. We use 1e-3 for Restaurants, Scholar, Academic and IMDB in the finetuning paradigm and FL paradigm.

For the computing resources, we use 1 NVIDIA A6000 48GB GPU for finetuning, with batch size 8. We use 2 NVIDIA A6000 48GB GPUs for centralized training, with batch size 8. We use 5 NVIDIA A6000 48GB GPUs for all federated learning experiments. Specifically, one GPU is used as the server and the other four GPUs are used as 8 clients, with each GPU accommodating 2 clients. The batch size for clients GeoQuery, Restaurants, Scholar, Academic, IMDB and Yelp is 4, and for clients Advising and ATIS is 8.

[3]Note we use Adafactor as the local optimizer for FedAvg, so the FedAvg in our paper is slightly different from the original proposed FedAvg, which uses stochastic gradient descent(SGD) as the local optimizer.

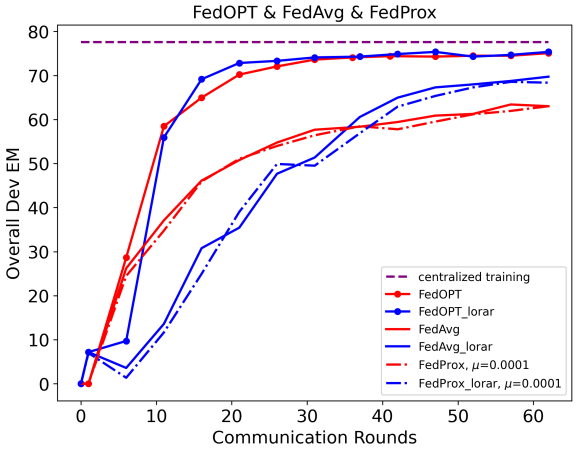

Figure 5: Overall dev performance, also equivalent to the MicroAvg on eight clients' dev sets. All the red curves show the original FL algorithms. All the blue curves show the algorithms after applying `Lorar`.

A.2 Comparison of FL Baselines.

We treat FedAvg, FedOPT and FedProx as our FL baselines. As Figure 5 shows, among FedAvg, FedOPT and FedProx, FedOPT performs the best, achieving the closest performance to the centralized paradigm and the fastest convergence speed. FedAvg and FedProx have similar performances, and both of them have a large gap with FedOPT. This indicates that the server's adaptive optimizer which only exists in FedOPT plays an important role to improve the performance.

A.3 Performance Variation under Varying Communication Rounds.

In Figure 5, comparing the performance of FL baselines with ours, FedOPT$_{\texttt{lorar}}$ performs slightly better than FedOPT. We hypothesize the small gap between FedOPT and the centralized paradigm limits the room for `Lorar` to show a large gain over FedOPT. For FedAvg and FedProx, we can see that applying `Lorar` performs significantly better, which demonstrates the effectiveness of leveraging the loss reduction to adjust the weights.

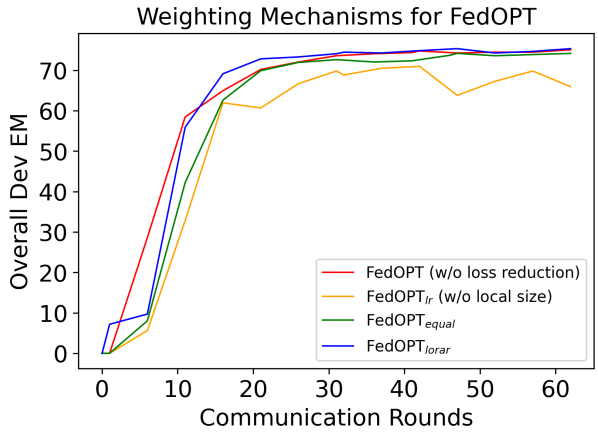

Figure 6: Alternative weighting mechanisms for FedOPT on the dev set of our proposed benchmark. Recall that FedOPT uses a client's training set size (w/o loss reduction) as its weight, FedOPT$_{lr}$ refers to only using a client's loss reduction during each round (w/o train set size) as its weight, while FedOPT$_{\texttt{lorar}}$ considers both factors (Eqn. (6)). FedOPT$_{equal}$ means each client gets equal weight.

Table: Restaurant

ID	Name	Food_Type	City_Name	Rating
321	Courtyard Cafe	American	Alameda	2.7
391	Bamboo Garden	Asian	San Bruno	2.0
...	...	...	...	...

Table: Geographic

City_Name	County	Region
Alameda	Alameda County	Bay Area
San Bruno	San Mateo County	Bay Area
...	...	...

Table: Location

Restaurant_ID	House_Number	Street_Name	City_Name
1	242	Church St	San Francisco
14	1520	Park St	Alameda
...	...	...	...

Question: Give me a restaurant in Alameda.

SQL:

```
Select Location.House_Number, Restaurant.Name
From Location, Restaurant
Where Location.City_Name = "Alameda"
      And Restaurant.ID = Location.Restaurant_ID
```

Figure 7: An overview of the text-to-SQL task.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
"Limitations" Section

☐ A2. Did you discuss any potential risks of your work?
Not applicable. Left blank.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
"Abstract" Section and "Introduction" Section (1)

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

"Evaluation Setup" (Section 3)

☑ B1. Did you cite the creators of artifacts you used?
"Experiments" (Section 5)

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not applicable. Left blank.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
"Evaluation Setup" (Section 3)

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
"Evaluation Setup" (Section 3)

C ☑ Did you run computational experiments?

"Implementation Details" (Section 5)

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
"Implementation Details" (Section 5 and Appendix)

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
"Implementation Details" (Section 5 and Appendix)

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
"Implementation Details" (Section 5 and Appendix), "Results" (Section 6)

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
"Implementation Details" (Section 5 and Appendix)

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.