Fedivertex: a Graph Dataset based on Decentralized Social Networks for Trustworthy Machine Learning

Marc Damie
University of Twente
Enschede, The Netherlands
m.f.d.damie@utwente.nl

Edwige Cyffers
Institute of Science and Technology Austria
Klosterneuburg, Austria

Abstract

Decentralized machine learning – where each client keeps its own data locally and uses its own computational resources to collaboratively train a model by exchanging peer-to-peer messages – is increasingly popular, as it enables better scalability and control over the data. A major challenge in this setting is that learning dynamics depend on the topology of the communication graph, which motivates the use of real graph datasets for benchmarking decentralized algorithms. Unfortunately, existing graph datasets are largely limited to for-profit social networks crawled at a fixed point in time and often collected at the user scale, where links are heavily influenced by the platform and its recommendation algorithms. The Fediverse, which includes several free and open-source decentralized social media platforms such as Mastodon, Misskey, and Lemmy, offers an interesting real-world alternative. We introduce *Fedivertex*, a new dataset of 182 graphs, covering seven social networks from the Fediverse, crawled weekly over 14 weeks. We release the dataset along with a Python package to facilitate its use, and illustrate its utility on several tasks, including a new *defederation* task, which captures a process of link deletion observed on these networks.

1 Introduction

Decentralized machine learning [19] has gained significant popularity in the past years. In this paradigm, each node possesses a local dataset and some computational resources, and collaborates with other participants by exchanging messages through peer-to-peer communication during the training of a global, potentially personalized, machine learning model. In comparison to federated learning [27], which keeps data local but orchestrates training through a central server, decentralized learning offers additional flexibility as it avoids the bottlenecks and single points of failure that arise from centralized supervision. The shift towards decentralized learning can also be motivated by trust, as the communication graph can reflect users' chosen collaborations–often referred to as nodes, instances, clients, or participants in this context. The network topology has an impact on the learning dynamics [28], particularly in the presence of data heterogeneity [24, 54] and in terms of privacy guarantees [45].

Decentralized learning has numerous real-world use cases [19], as nodes can represent healthcare institutions or sensors distributed across installations. One of the most compelling applications is for decentralized social networks [6]. In such cases, the graph often captures complex and diverse relationships, which explains its popularity in machine learning. In particular, this motivates various graph learning tasks [52], including community detection, node classification, and edge prediction. Social network dynamics also raise interesting questions related to polarization and time-evolving properties of the graph [53]. Addressing these questions requires access to relevant social network datasets that allow studying these properties.

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

The Fediverse – a contraction of "federation" and "universe" – provides decentralized and interoperable online social services. It is often seen as an alternative [2] to major social networks operated by for-profit companies, and it promotes a very different culture. The Fediverse is decentralized across many servers, called *instances*. Anyone can run an instance, which operates independently under the moderation of its owner, and instances collaborate with each other: a user of a given instance can interact with and follow users from other instances. Since 2018, the Fediverse has adopted ActivityPub [32], a protocol and open standard that provides a client-to-server API for creating and modifying content, as well as a federated server-to-server protocol for delivering notifications and content across servers. This enables interoperability between different instances and software. The diversity of platforms, the growing number of users, and the international impact of the Fediverse make it an interesting object of study for the machine learning community. In particular, agents in the Fediverse tend to be more aware of the potential ethical issues of machine learning than traditional users of social networks [49], and more interested in new features and improvements. This aligns closely with the goals of Trustworthy Machine Learning and the paradigm of collaborative learning, where agents are expected to monitor their participation based on expected benefits.

In this work, we provide the first dataset covering multiple software platforms in the Fediverse, called *Fedivertex*, to enable researchers to easily run experiments on decentralized machine learning tasks and to benchmark several graph learning tasks. By surveying seven different platforms and constructing different types of graphs, we are able to capture the diverse dynamics at play in the Fediverse. In particular, a striking difference from many mainstream social networks is the so-called *defederation* process [30], in which instances choose to sever ties with other instances, often due to a disagreement on moderation or security practices. While new link prediction is often regarded as the primary task for time-evolving graphs [20], a major dynamic in the Fediverse is this complementary edge deletion. Our dataset is the first to enable the study of this phenomenon. The current 14 distinct timestamps for each graph are a starting point to study the evolution over time, and we plan to continue to update the dataset in the future. More precisely, our contributions are as follows:

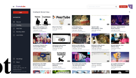
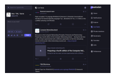
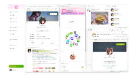
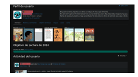
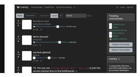
- (i) We introduce *Fedivertex*, a large and diverse graph dataset based on the Fediverse. More precisely, our dataset encompasses seven Fediverse platforms, resulting in 182 graphs: 13 different graphs each with a sequence of 14 snapshots obtained through weekly web crawls over a period of three months.
- (ii) We provide a Python package, `fedivertex`, available through PyPI, to easily access and use our dataset. The package includes built-in preprocessing tools to download and prepare the graphs for machine learning tasks. We demonstrate its usefulness by benchmarking several existing decentralized learning algorithms.
- (iii) We formalize a novel graph analysis task: *defederation prediction*, which aims to predict which edges or nodes will be removed from the graph at the next iteration, and we propose baselines for this task.

2 Related work

Decentralized machine learning. Federated learning and fully decentralized learning are increasingly studied [27, 36, 43], with various algorithms based on gossip [8, 21, 22, 29, 37, 44, 46, 51] or random walks [17, 26, 21, 40]. These results highlight the importance of communication graphs for the quality of the final model, the speed of convergence in the presence of heterogeneous data [31], personalization [5] and privacy guarantees [12, 13, 15, 45], which motivates the use of recent real-world social networks.

Social network datasets and analysis. Machine learning frequently relies on small social networks, such as the Karate Club [55] or citation networks [18, 42]. Several larger digital social networks are also available via platforms like SNAP [34], in particular Facebook and Twitter graphs. It has been shown that for-profit platforms influence user graphs, as their recommendations about whom to follow accelerate the triadic closure process and exacerbate inequality in popularity [50, 56]. This motivates the study of social networks that do not follow this trend. In particular, the Fediverse enables analysis at the level of servers rather than at the level of individual users, an approach that captures entities more likely to develop consistent collaboration policies. Prior work on the Fediverse remains limited, often focusing on a single network or on interactions between a fixed pair of networks, and typically does not provide reusable datasets [1, 23, 56].

Table 1: Overview of the Fedivertex social networks. The number of instances and users has been extracted on May 13, 2025 from FediDB, a reference database for the Fediverse communities. These numbers are indicative as the networks evolve over time.

	Peertube	Mastodon	Pleroma	Misskey	Friendica	Bookwyrm	Lemmy
Type	Video streaming	Micro blogging	Micro blogging	Micro blogging	Micro blogging	Book cataloging	Social news
1st release	2018	2016	2017	2014	2010	2022	2019
Screenshot							
# instances	1333	9652	616	1206	101	95	564
# users	583k	8 102k	76k	1 071k	12k	51k	520k
Graphs	follow	federation active user	federation active user	federation active user	federation	federation	fed. + blocks intra-instance cross-inst.

3 Fedivertex Dataset

3.1 Fediverse software and graphs

In the Fediverse, software is run by servers referred to as *instances*, without any centralized control or coordination. Each instance hosts a subset of *users* and has its own internal rules and moderation. Despite maintaining sovereignty over their rules and storing data locally, instances are not isolated from each other, as they all use the same protocol and standard: ActivityPub [32]. This protocol enables communication between instances and even across services. For instance, a video from PeerTube can be shared on Mastodon, and the resulting post can be viewed from Misskey – unlike traditional social network silos, where a Facebook user cannot use their account to read tweets or watch YouTube videos. One can think of this interoperability similarly to email, where a user from one provider (e.g., Gmail) can send a message to another (e.g., Outlook). ActivityPub includes both a federation protocol – a server-to-server protocol that allows instances to share information – and a social API–a client-to-server protocol that allows users to send information to their instance. A user's data is stored on their respective instance but can be duplicated and cached on other instances to be accessible to other users. When two users communicate, only their respective instances – and possibly a third instance hosting the interaction – are aware of the message. As a result, data permanence, confidentiality, and moderation depend on the instance.

Fedivertex focuses on interactions between *instances* within a given software platform. In particular, for each social network (except Peertube), the **federation graph** models the undirected communication graph between instances within that network. Federation graphs are naturally dense, because two instances are connected with an edge if they have interacted at least once. We selected seven of the most popular software platforms in the Fediverse to ensure sufficient activity for graph-based analysis. Our selection covers diverse types of social network to reflect a range of communication dynamics. We summarize the dataset in table 1 and present each of them in more detail below.

3.2 Fediverse social networks

Peertube provides an alternative to YouTube. Users can watch, bookmark, and comment on videos, subscribe to channels, and create private and public playlists. Video search was added in 2020 with SepiaSearch but remains limited; recommendation features are also a limitation compared to Youtube. An instance u can follow an instance v to let its users see all the videos posted on v. We report this **follow graph** as a directed graph with edges of weight 1 for following.

Mastodon was created as an alternative to Twitter in 2016 and is supported by the German non-profit organization Mastodon gGmbH. Users post short-form status messages of up to 500 characters, known as "toots." It has experienced several surges in popularity, often in reaction to changes on Twitter, and is sometimes adopted in parallel with it [25]. In addition to the federation graphs, introduced above, we also build a weighted, directed **active user graph**, with one node per instance. For each instance u, we take its 10k most recently active users. Whenever one of those users follows

```python
from fedivertex import GraphLoader

loader.list_graph_types("mastodon")
# List available graphs for a given software, here federation and active_user

G = loader.get_graph(software = "mastodon", graph_type = "active_user", index = 0, only_largest_component = True)
# G contains the Networkx graph of the giant component of the active users graph at the 1st date of collection
```

Listing 1: Code example from the `Fedivertex` package

someone on instance v, we increase the edge weight by 1. Thus weight of the edge from u to v measures how much content seen on u originates from v. The graph thus contains self-loop as users follow others on the same instance.

Pleroma is a microblogging software similar to Mastodon, and we thus also report active user graph. Principal difference is allowing longer posts by default, up to 5000 characters, and offering a lightweight implementation that can potentially run on a Raspberry Pi.

Misskey is a microblogging platform as well, on which we report the active user graph. It was created in 2014 by Japanese software engineer Eiji "syuilo" Shinoda and allows posts of up to 3000 characters.

Friendica emerged in 2010 as an alternative to Google+ and Facebook, making it the oldest social network in our study, and does not support metadata for active users graph with our crawler.

Bookwyrm allows users to track their reading activity, write book reviews, and follow friends. Launched in 2022 by Mouse Reeve, it can be seen as an alternative to Goodreads.

Lemmy is organized into communities dedicated to specific topics, where users share links and discuss their content. Although communities are local to an instance, users can subscribe to those hosted by other instances and participate in discussions across instances. In addition to the federation graph, we report two other graphs. Firstly, the **intra-instance graph** where the instance u is linked to v if an user of u has published a message on instance v. This graph is directed and very sparse. Then, in **cross-instance graph**, two instances are connected as soon as there exists a pair of users who published a message in the same thread, but possibly on a third instance. This is an undirected graph, denser that the previous one.

3.3 Fedivertex package and availability

Our dataset can be directly downloaded from Kaggle [14]. To facilitate its use, we also provide a Python package, `Fedivertex`, which allows users to directly load the graphs in NetworkX format through an easy-to-use interface, as shown in listing 1. We facilitate interaction with Fedivertex package by releasing several notebooks to analyze the graphs. Finally, we follow the Gephi convention for graph encoding, allowing the graph CSV files to be opened directly in this software [4].

3.4 Construction via Web crawling

For each of the 13 graphs introduced above, we produce a new version every week (thus presenting 14 different timestamps of each of the graph at the moment of the article submission). To identify all available servers for a given software, we query the Fediverse Observer[1], which provides a curated list of Fediverse instances commonly accepted by the community. We then query each of these instances to compute the edges of the graph. Relying on the Fediverse Observer list helps minimize server load and allows us to benefit from existing curation. Notably, the Fediverse Observer's crawler runs daily and is also open-source. We release the code of our crawler for reproducibility and to allow extensions to other social media or other scraping parameters.

3.5 Ethical concerns

Our work aims to bring more attention to the Fediverse social networks, who could benefit from Trustworthy Machine Learning applications, for instance to assist in moderation task [6] or with user experience and recommendation systems. However, Fediverse software has often been developed to

[1] `https://fediverse.observer`

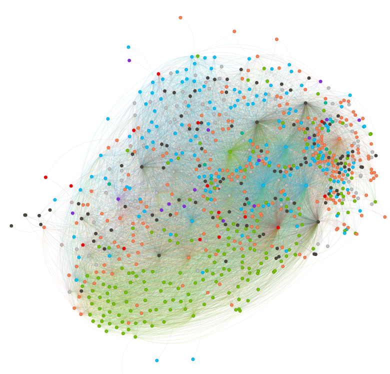
avoid various downsides of hastily deployed machine learning models, from toxicity to invisible filter bubbles, and dark patterns to poor accuracy [9, 41]. It is thus part of the challenge and the motivation to focus on tasks where the improvement for the user overcomes the possible drawbacks. Hence, we decide to illustrate our datasets only on tasks that could respect the Fediverse mindset.

The collection of the dataset raises two major problems, the privacy and the possible disturbance for the service. Concerning the impact of the crawling, we designed our crawler to minimize the impact on the servers, by limiting the size of the queries, using a delay of 0.4 second between requests on a given instance to avoid disturbing their operations. We do not disguise our requests into real users' ones and we use a clearly identifiable user agent providing a direct contact to us. We respect the policy of the instances by following the instructions given by `robots.txt` files.

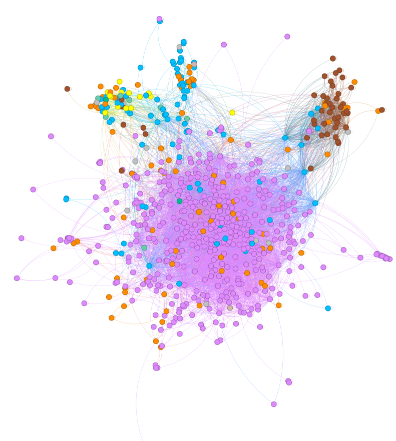
In order to respect the privacy of the users, we did several design choices. First, our dataset is instance-based and not user-based, which is a better granularity for privacy. Second, we only report general metadata but never store actual messages or content from the social networks. Third, we only use public API endpoints, which do not require accounts on this platform: we do not try to circumvent these privacy practices by creating accounts to access more information. Forth, we also respect informal privacy practices. For instance, we ignore all users using the hashtag #NOBOT in their profile as it is an informal anti-bot policy on Mastodon. Finally, we limit the access by post-processing instances names to avoid direct clicking links. The whole scraping process was supervised by the legal department of our institutes to ensure compliance with GDPR.

4 Dataset analysis

4.1 Dataset properties



(a) Peertube Follow graph on April 28th 2025. Colors encode the official language of each instance, with green for French, blue for English, black for German, red for Italian and orange when there is no official language. Interactive version: `https://marc.damie.eu/peertube_graph/index.html`



(b) Misskey Active User graph after removal of `Misskey.io` on May 7th, same colors than left figure, and Chinese in yellow, Japanese in purple and Korean in brown. Interactive version: `https://marc.damie.eu/misskey_graph/index.html`

Figure 1: Examples of graphs communities based on the official languages in Fedivertex dataset

The Fedivertex dataset presents different characteristics depending on both the graph and the software considered. We share in this section a few observations. First, all graphs are provided with their temporal evolution, with new nodes and edges appearing or disappearing each week. For readability, we present only a subset of the graphs and refer the reader to appendix A and the notebooks for a more systematic review[2].

Communities. Fedivertex contains language labels for Peertube, Lemmy, Bookwyrm, and Misskey. For Peertube, the labels are directly extracted from the instance descriptions. For the others, we

[2] `https://www.kaggle.com/code/marcdamie/exploratory-graph-data-analysis-of-fedivertex`

```
G = loader.get_graph(software = "misskey", graph_type = "active_user")
lang_labels = [data["description_language"] for node_name, data in G.nodes(data=True)]
```

Listing 2: Code example to extract language information from the Misskey active user graphs

Table 2: Small world properties of the Fedivertex graphs and SNAP Social graphs

Graph	Directed	Avg. Degree	Avg. Path Length	Cluster. Coef.	Small-world σ
Bookwyrm Federation	No	55	1.23	0.89	1.14
Friendica Federation	No	156	1.41	0.85	1.41
Lemmy Federation	No	355	1.18	0.94	1.15
Lemmy Intra-instance	Yes	13	2.03	0.69	3.80
Lemmy Cross-instance	Yes	42	1.60	0.82	1.89
Mastodon Active user	Yes	125	2.09	0.73	21.95
Misskey Federation	No	317	1.66	0.76	2.21
Misskey Active user	Yes	19	2.36	0.62	20.78
Peertube Follow	Yes	23	2.82	0.53	4.45
Pleroma Federation	No	269	1.64	0.82	2.26
Pleroma Active user	Yes	7	3.95	0.30	2.53
Facebook Ego	No	47	3.7	0.61	39.44
Github	No	29	3.25	0.14	31.49
Wikipedia Vote	No	15	3.25	0.17	519.64

infer the label from the language of the instance description. The labels can be easily used for label prediction tasks through our API, as described in appendix B. We report on fig. 1a the labels for the Peertube graph, which exhibits a clear French-speaking community and Misskey, which is dominated by the Japanese community but also exhibits smaller Korean and Chinese communities, and we refer to section 5.3 for the associated prediction task.

Graph statistics. We report few metrics in table 2 that are usually applied for social networks. All the reported graphs exhibit small-world properties to an extend, as they satisfies $\sigma > 1$, which means that a node is more likely to connect to the neighbors of its neighbors and the average path length is small. However, the strength of the small world properties depends on the software.

4.2 Comparison with existing graph datasets

We compare our graph with the most popular social network graphs from SNAP [34]. We include the Wikipedia Vote graph [33]– which encodes all Wikipedia voting data up to January 2008, representing each user who participated in a vote as a node and adding a directed edge from node i to node j if user i voted for user j– as well as the Twitter and Facebook Ego graphs [35], corresponding respectively to the Follow and Friend relationships. We also include the GitHub graph [48], where nodes are developers who have starred at least 10 repositories and edges represent mutual follower relationships. On fig. 2, while GitHub and Twitter exhibit the classical power-law decay over much of the support of the distribution, consistently with preferential attachment networks [3], the degree distribution of Fedivertex is more diverse. We note some similarities between the Facebook Ego graph and the Peertube follow graph, with a smooth concave decay followed by a short power-law

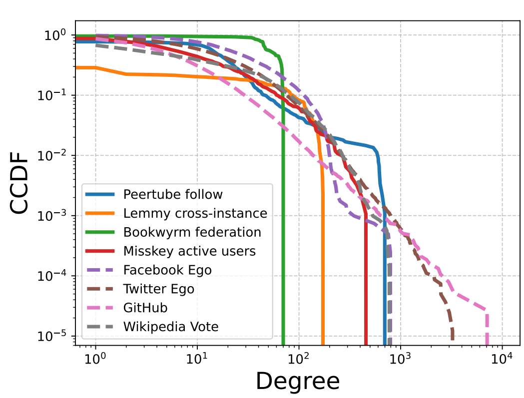


Figure 2: Complementary Cumulative Distribution Function (CCDF) of the degree for several Fedivertex graphs and other widely-used social networks. A normalized version is available in appendix B

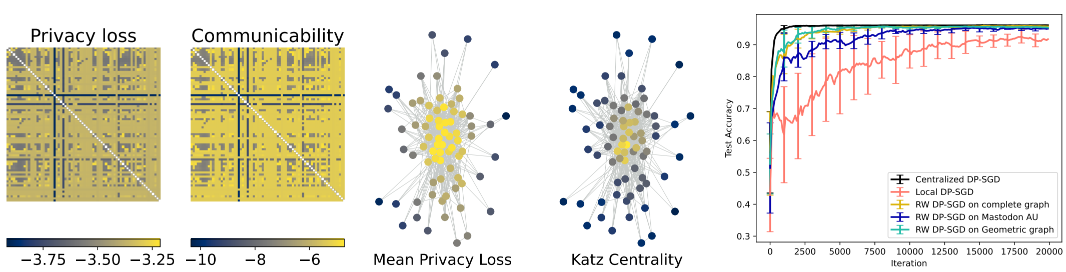
tail for the most popular users. Most of the graphs, however, exhibit only the a concave curve, which suggests that the attachment dynamics in Fedivertex differ from those in traditional social networks, allowing for a smoother distribution of node importance. This difference with Fedivertex is confirmed by the statistics of table 2: Fedivertex has a wider range of average path length (from 1.18 to 3.95 versus from 3.25 to 3.7) and of degree (from 7 to 355 versus 15 to 47) and have smaller small-world σ. A more systematic comparison is available in this notebook[3].

5 Applications

5.1 Decentralized machine learning and statistics

Fedivertex graphs are particularly well adapted to experiments testing fully decentralized machine learning, as they provide a credible use-case scenario. We illustrate this by reproducing the main figures of [12]. This paper proposes training a global model with differential privacy by performing a random walk on the communication graph: at each step, the stochastic gradient is computed on the local dataset of the current node and sanitized by adding Gaussian noise. The paper derives privacy guarantees in the Pairwise Network Differential Privacy setting, where each pair of nodes has a specific privacy budget depending on their relative position, a high budget corresponds to a greater risk of leaking information. In particular, the paper establishes a connection between the structure of these privacy budgets and the communicability of the graph, showing that nodes close to each other have higher privacy budget than far apart ones. Using graphs with different topologies is interesting to verify that similar patterns appear for privacy losses and known graph quantities such as centrality or communicability. Finally, the paper claims to be quite efficient in terms of privacy–utility trade-offs.

On fig. 3, we see that the link between communicability and privacy budgets is clear on Fedivertex graphs, with the same patterns visible in fig. 3a and fig. 3b. However, it also shows that real-world graphs can be more challenging in terms of convergence, as fig. 3c exhibits slower convergence on the Mastodon active-user graph than on a synthetic graph with the same number of nodes. This could be explained by the presence of nodes with low centrality, typically connected by only a single edge, which makes it harder for the random walk to visit them frequently enough within the chosen number of steps compared with the more regular graph tested in the original paper. To ease comparison, we provide more background on the task and the original figures in appendix B.



(a) Bookwyrm federation graph communicability and privacy loss, logarithmic scale

(b) Mean privacy loss and Katz centrality for Lemmy intra-instance graph

(c) Test accuracy on private logistic regression on the Houses dataset for graph with 3852 nodes

Figure 3: Numerical experiments reproducing the results of [12] with Fedivertex graphs

5.2 New links and defederation prediction

A key feature of Fedivertex is its support for temporal graph analysis, as we release weekly snapshots for each graph. Understanding temporal changes in graphs is seen as a major challenge in graph learning [39, 47]. This is particularly relevant in social networks, where the creation of links often speeds up the triadic closure of the graph, as friends of friends tend to become friends over time [16, 56], especially when platforms actively recommend new connections [50]. We refer to table 2 for Fedivertex graphs' clustering coefficients.

[3] `https://www.kaggle.com/code/marcdamie/fedivertex-vs-snap-social-graphs/notebook`

Table 3: Comparison of Adamic-Adar (AA), Common Neighbors (CN), Jaccard and Random method to predict new edge (Add) and edge deletion (Del) on different graphs by reporting the number of correct predictions in Top-K scores (the higher the better). We report the average of three runs over disjoint periods of time.

Graph	**AA**		**CN**		**Jaccard**		**Random**	
	Add	Del	Add	Del	Add	Del	Add	Del
Mastodon AU (Top 1000)	38	10	40	9	14	10	0.7 ± 0.5	6 ± 1.4
Misskey AU (Top 200)	4.3	1.3	4.3	1.7	0.7	2	0.2 ± 0.2	2 ± 0.8
Misskey federation (Top 1000)	494	62	491	63	396	60	42 ± 6	51 ± 3.7

An interesting behavior observed in the Fediverse is that edges between instances are sometimes deleted, a phenomenon that has received little attention so far, likely because edge deletions are uncommon in other social networks. However, in the context of Fedivertex, predicting deletions is interesting for several reasons. First, in some platforms, deletions are as important and can dominate the change in the number of edges and nodes, as seen in fig. 4. Secondly, avoiding centralization around a single central server is a key challenge in the Fediverse, and understanding defederation [30] could help maintain a sufficiently decentralized structure. Finally, new link prediction and edge deletion are complementary tasks that may benefit from being studied jointly.

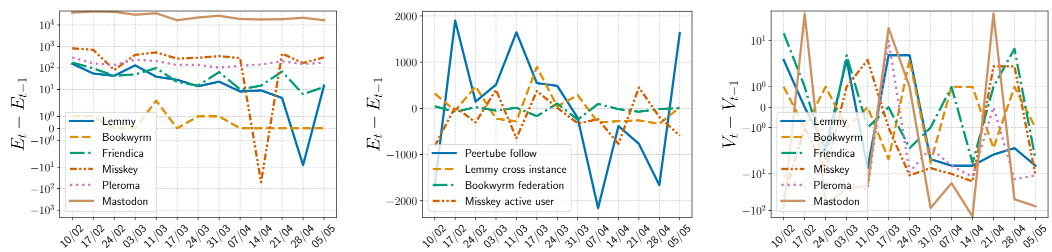
From fig. 4, we observe that federation graphs are the most stable over time, as one could expect from their construction in comparison to active users or cross-instance graphs, where activity can fluctuate. However, while some networks grow during the studied period (Friendica federation and Lemmy cross-instance), others show variations dominated by the loss of edges (such as the Pleroma federation or Misskey active users). More precisely, by restricting our analysis to the subgraph of the nodes present at all 14 dates, most federation graphs have an increasing number of edges, with sometimes sharp drops as we can see for Misskey and Lemmy in fig. 4a. Overall, the network is thus growing, but also shows defederation peaks that are quicker than our weekly crawling. In fig. 4b, we can see that the other graphs do not share this clear increasing trend, but tend to alternate between more edge creation and more edge deletion. Finally, the number of nodes itself varies over time, with new servers appearing and others being deleted, leading to the complex evolution reported in fig. 4c.

We formally introduce the defederation prediction task. Let $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t)$ be the graph collected at time t and compare it to the graph $\mathcal{G}_{t'} = (\mathcal{V}_{t'}, \mathcal{E}_{t'})$ collected at $t' > t$. Let $\mathcal{G}_c$ be the subgraph induced by $\mathcal{V}_c = \mathcal{V}_t \cap \mathcal{V}_{t'}$; the goal is to predict all edges either created $(e \in (\mathcal{V}_c \times \mathcal{V}_c) \cap (\mathcal{E}_{t'} \setminus \mathcal{E}_t))$ or deleted $(e \in (\mathcal{V}_c \times \mathcal{V}_c) \cap (\mathcal{E}_t \setminus \mathcal{E}_{t'}))$ between t and t'. The possible new edges lie in the set $(\mathcal{V}_c \times \mathcal{V}_c) \setminus \mathcal{E}_t$, whereas the deleted ones are in $\mathcal{E}_t$, a set significantly smaller if the graph is sparse. Similarly, one could predict nodes that drop from the graph. In particular, reliable prediction could help detect instances which stopped running because of technical problems despite being active in the graph, and possibly provide technical help to such instances. More formally, the goal would be to predict $\mathcal{V}_t \setminus \mathcal{V}_{t+1}$ given $\mathcal{G}_t$.

New link prediction can be done based on the topology [38], by using the fact that similar nodes are more likely to connect. Methods are thus often based on computing scores for each possible pair of nodes, and then return as prediction the edges with the highest scores. Common scores include the number of common neighbors, the Jaccard score, and the Adamic-Adar score. These scores are then evaluated by looking among the top-K predictions how many are indeed new edges, as we report in table 3. Intuitively, deletion could be seen as the opposite of edge creation, so we propose as a baseline, to return the edges with the lowest scores. However, this approach has limited success for federation graphs. We believe this might be due to defederation being extremely quick, and thus the granularity of our current dataset does not seem sufficient to achieve better than random. An interesting future work could be to use our crawler with higher frequency during defederation periods. It might also indicate that other methods should be developed for this task, opening interesting questions for future work. We refer the reader to appendix C for more analysis.

5.3 Community detection

The Fedivertex social networks are used by many communities that might overlap. The same communities might also use several of the Fedivertex software platforms. To illustrate the feasibility



(a) Variation of the number of edges over time, on the subgraph of the nodes appearing at all iterations for federation graphs, logarithmic scale

(b) Variation of the number of edges over time, on the subgraph of the nodes appearing at all iterations, for various type of graphs

(c) Variation of the number of nodes over time for the seven software of Fedivertex, logarithmic scale

Figure 4: Temporal evolution of Fedivertex graphs

of community detection with our dataset, we use the official languages of each instance as ground truth labels, as shown in fig. 1, and test three algorithms: Louvain [7], Greedy Modularity [10], and Label Propagation [11] on the Peertube follow graph and the Misskey active user graph. To avoid many unique labels, we keep only the 5 most represented languages in each graph and ignore the other nodes using other languages. Results are in table 4. We assess the quality of this detection using the Adjusted Rand Index (ARI) and the Adjusted Mutual Information (AMI). The Rand index corresponds to the proportion of node pairs belonging to the same cluster that are classified as such (i.e., the sum of true positives for all classes) over all node pairs, and the adjusted version corresponds to normalization with respect to a random clustering. Similarly, the AMI score corresponds to the mutual information between the ground truth and predicted labels, adjusted for chance. Finally, we report modularity for each clustering – a metric for unsupervised clustering that assesses the inherent quality of the partitioning. This suggests that other labels might be suitable as well for clustering the graphs. No method dominates in this benchmark, highlighting that our graphs exhibit diverse structures which may challenge algorithms in different ways. Experiments are in notebook.[4]

Other Fedivertex graphs can be used for community detection, and additional labels could be derived from the data, for example, based on the names of the instances or their official descriptions. It is also possible to track the evolution of communities over time.

Table 4: Performance of several community detection algorithms (average of 100 runs for Louvain).

Graph	Algorithm	ARI	AMI	Modularity
Peertube follow	Louvain	0.055	**0.123**	**0.2168**
	Greedy Modularity	**0.061**	0.110	0.209
	Label Propagation	0.008	0.029	0.003
Misskey active user	Louvain	0.097	0.250	0.611
	Greedy Modularity	0.014	0.165	**0.513**
	Label Propagation	**0.229**	**0.255**	0.027

6 Conclusion

In this work, we introduce `Fedivertex`, a dataset modeling interactions between instances across several software platforms of the Fediverse. This is the first dataset publicly released to enable reproducible experiments on graphs from the Fediverse, and it allows the study of more diverse graph dynamics than existing social network datasets. We hope that these graphs can foster machine learning research on this topic and contribute to the development of trustworthy decentralized machine learning, notably on the Fediverse. Among possible applications, this dataset could support the development of decentralized spam detection, the prediction of new or deleted links, the prevention of instance shutdowns through early prediction, and many other tasks related to time-evolving graphs.

[4] `https://www.kaggle.com/code/marcdamie/community-detection-on-fedivertex`

References

[1] Vibhor Agarwal, Aravindh Raman, Nishanth Sastry, Ahmed M Abdelmoniem, Gareth Tyson, and Ignacio Castro. Decentralised moderation for interoperable social networks: A conversation-based approach for Pleroma and the Fediverse. In *Conference on Artificial Intelligence (AAAI)*, volume 18, 2024.

[2] Jacopo Anderlini and Carlo Milani. Emerging Forms of Sociotechnical Organisation: The Case of the Fediverse. In Emiliana Armano, Marco Briziarelli, and Elisabetta Risi, editors, *Digital Platforms and Algorithmic Subjectivities*, volume 24. University of Westminster Press, 2022.

[3] Albert-László Barabási and Márton Pósfai. *Network science*. Cambridge University Press, Cambridge, 2016.

[4] Mathieu Bastian, Sebastien Heymann, and Mathieu Jacomy. Gephi: an open source software for exploring and manipulating networks. In *Proceedings of the international AAAI conference on web and social media*, 2009.

[5] Aurélien Bellet, Rachid Guerraoui, Mahsa Taziki, and Marc Tommasi. Personalized and Private Peer-to-Peer Machine Learning. In *Conference on Uncertainty in Artificial Intelligence (AISTATS)*, 2018.

[6] Haris Bin Zia, Aravindh Raman, Ignacio Castro, Ishaku Hassan Anaobi, Emiliano De Cristofaro, Nishanth Sastry, and Gareth Tyson. Toxicity in the Decentralized Web and the Potential for Model Sharing. *Proc. ACM Meas. Anal. Comput. Syst.*, 6(2), June 2022.

[7] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10), October 2008.

[8] Stephen Boyd, Arpita Ghosh, Balaji Prabhakar, and Devavrat Shah. Randomized gossip algorithms. *IEEE transactions on information theory*, 52(6), 2006.

[9] Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, and Xiangnan He. Bias and debias in recommender system: A survey and future directions. *ACM Trans. Inf. Syst.*, 41(3), February 2023.

[10] Aaron Clauset, M. E. J. Newman, and Cristopher Moore. Finding community structure in very large networks. *Physical Review E*, 70(6), December 2004. Publisher: American Physical Society.

[11] Gennaro Cordasco and Luisa Gargano. Community detection via semi-synchronous label propagation algorithms. In *2010 IEEE International Workshop on: Business Applications of Social Network Analysis (BASNA)*, December 2010.

[12] Edwige Cyffers, Aurélien Bellet, and Jalaj Upadhyay. Differentially private decentralized learning with random walks. *International Conference on Machine Learning (ICML)*, 2024.

[13] Edwige Cyffers, Mathieu Even, Aurélien Bellet, and Laurent Massoulié. Muffliato: Peer-to-peer privacy amplification for decentralized optimization and averaging. *Advances in Neural Information Processing Systems (NeurIPS)*, 35, 2022.

[14] Marc Damie and Edwige Cyffers. Fedivertex, 2025. URL: https://www.kaggle.com/ds/6877842.

[15] Florine W. Dekker, Zekeriya Erkin, and Mauro Conti. Topology-based reconstruction prevention for decentralised learning, 2023.

[16] Daniel M. Dunlavy, Tamara G. Kolda, and Evrim Acar. Temporal link prediction using matrix and tensor factorizations. *ACM Trans. Knowl. Discov. Data*, 5(2), February 2011.

[17] Mathieu Even. Stochastic Gradient Descent under Markovian Sampling Schemes, February 2023. arXiv:2302.14428 [cs, math].

[18] C. Lee Giles, Kurt D. Bollacker, and Steve Lawrence. Citeseer: an automatic citation indexing system. In *Proceedings of the Third ACM Conference on Digital Libraries*, DL '98, New York, NY, USA, 1998. Association for Computing Machinery.

[19] Ehsan Hallaji, Roozbeh Razavi-Far, Mehrdad Saif, Boyu Wang, and Qiang Yang. Decentralized Federated Learning: A Survey on Security and Privacy. *IEEE Transactions on Big Data*, 10(2), April 2024.

[20] Mohammad Al Hasan and Mohammed J. Zaki. A Survey of Link Prediction in Social Networks. In Charu C. Aggarwal, editor, *Social Network Data Analytics*. Springer US, Boston, MA, 2011.

[21] Hadrien Hendrikx. A principled framework for the design and analysis of token algorithms, May 2022. Number: arXiv:2205.15015 arXiv:2205.15015 [cs, math].

[22] Hadrien Hendrikx, Francis Bach, and Laurent Massoulié. Dual-free stochastic decentralized optimization with variance reduction. In *Conference on Neural Information Processing Systems (NeurIPS)*, volume 33. Curran Associates, Inc., 2020.

[23] Shiori Hironaka, Mitsuo Yoshida, and Kazuyuki Shudo. Comparing user activity on X and Mastodon. In *2024 IEEE International Conference on Big Data (BigData)*. IEEE, December 2024.

[24] Kevin Hsieh, Amar Phanishayee, Onur Mutlu, and Phillip Gibbons. The Non-IID Data Quagmire of Decentralized Machine Learning. In *International Conference on Machine Learing (ICML)*. PMLR, November 2020. ISSN: 2640-3498.

[25] Ujun Jeong, Paras Sheth, Anique Tahir, Faisal Hammad Alatawi, H. Russell Bernard, and Huan Liu. Exploring platform migration patterns between Twitter and Mastodon: A user behavior study, 2023.

[26] Björn Johansson, Maben Rabi, and Mikael Johansson. A randomized incremental subgradient method for distributed optimization in networked systems. *SIAM Journal on Optimization*, 20(3), 2010.

[27] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Hubert Eichner, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaid Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konecný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Hang Qi, Daniel Ramage, Ramesh Raskar, Mariana Raykova, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. Advances and Open Problems in Federated Learning. *Foundations and Trends® in Machine Learning*, 14(1–2), 2021.

[28] Hanna Kavalionak, Emanuele Carlini, Patrizio Dazzi, Luca Ferrucci, Matteo Mordacchini, and Massimo Coppola. Impact of Network Topology on the Convergence of Decentralized Federated Learning Systems. In *2021 IEEE Symposium on Computers and Communications (ISCC)*, September 2021. ISSN: 2642-7389.

[29] Anastasia Koloskova, Nicolas Loizou, Sadra Boreiri, Martin Jaggi, and Sebastian Stich. A unified theory of decentralized SGD with changing topology and local updates. In *International Conference on Machine Learing (ICML)*, volume 119 of *Proceedings of Machine Learning Research*. PMLR, 13–18 Jul 2020.

[30] Samantha Lai, Yoel Roth, Kate Klonick, Mallory Knodel, Evan Prodromou, and Aaron Rodericks. New Paradigms in Trust and Safety: Navigating Defederation on Decentralized Social Media Platforms. Technical report, Carnegie Endowment for International Peace, April 2025.

[31] Batiste Le Bars, Aurélien Bellet, Marc Tommasi, Erick Lavoie, and Anne-Marie Kermarrec. Refined convergence and topology learning for decentralized sgd with heterogeneous data. In *Conference on Uncertainty in Artificial Intelligence (AISTATS)*, volume 206 of *Proceedings of Machine Learning Research*. PMLR, 25–27 Apr 2023.

[32] Christine Lemmer-Webber, Jessica Tallon, Erin Shepherd, Amy Guy, and Evan Prodromou. Activitypub w3c recommendation, 2018.

[33] Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. Predicting positive and negative links in online social networks. In *Proceedings of the 19th international conference on World wide web*, WWW '10, New York, NY, USA, April 2010. Association for Computing Machinery.

[34] Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection, 2014.

[35] Jure Leskovec and Julian Mcauley. Learning to Discover Social Circles in Ego Networks. In *Conference on Neural Information Processing Systems (NeurIPS)*, volume 25. Curran Associates, Inc., 2012.

[36] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3), May 2020.

[37] Xiangru Lian, Ce Zhang, Huan Zhang, Cho-Jui Hsieh, Wei Zhang, and Ji Liu. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. In *Conference on Neural Information Processing Systems (NeurIPS)*, NIPS'17, Red Hook, NY, USA, 2017. Curran Associates Inc.

[38] David Liben-Nowell and Jon Kleinberg. The link-prediction problem for social networks. *Journal of the American Society for Information Science and Technology*, 58(7), March 2007.

[39] Soumen Majhi, Matjaž Perc, and Dibakar Ghosh. Dynamics on higher-order networks: a review. *Journal of The Royal Society Interface*, 19(188), March 2022.

[40] Xianghui Mao, Kun Yuan, Yubin Hu, Yuantao Gu, Ali H. Sayed, and Wotao Yin. Walkman: A Communication-Efficient Random-Walk Algorithm for Decentralized Optimization. *IEEE Transactions on Signal Processing*, 68, 2020.

[41] Antonis Matakos, Evimaria Terzi, and Panayiotis Tsaparas. Measuring and moderating opinion polarization in social networks. *Data Mining and Knowledge Discovery*, 31(5), July 2017.

[42] Andrew Kachites McCallum, Kamal Nigam, Jason Rennie, and Kristie Seymore. Automating the construction of internet portals with machine learning. *Information Retrieval*, 3(2), 2000.

[43] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Conference on Uncertainty in Artificial Intelligence (AISTATS)*, volume 54 of *Proceedings of Machine Learning Research*. PMLR, 20–22 Apr 2017.

[44] Aryan Mokhtari and Alejandro Ribeiro. Dsa: Decentralized double stochastic averaging gradient algorithm. *Journal of Machine Learning Research*, 17(61), 2016.

[45] Abdellah El Mrini, Edwige Cyffers, and Aurélien Bellet. Privacy attacks in decentralized learning. *International Conference on Machine Learning (ICML)*, 2024.

[46] Giovanni Neglia, Chuan Xu, Don Towsley, and Gianmarco Calbi. Decentralized gradient methods: does topology matter? In *Conference on Uncertainty in Artificial Intelligence (AISTATS)*, 2020.

[47] Emanuele Rossi, Benjamin Paul Chamberlain, Fabrizio Frasca, Davide Eynard, Federico Monti, and Michael M. Bronstein. Temporal graph networks for deep learning on dynamic graphs, 2020.

[48] Benedek Rozemberczki, Carl Allen, and Rik Sarkar. Multi-scale attributed node embedding, 2019.

[49] Thomas Struett, Aram Sinnreich, Patricia Aufderheide, and Robert Gehl. Can this platform survive? governance challenges for the Fediverse. *SSRN Electronic Journal*, 2023.

[50] Jessica Su, Aneesh Sharma, and Sharad Goel. The effect of recommendations on network structure. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, Republic and Canton of Geneva, CHE, 2016. International World Wide Web Conferences Steering Committee.

[51] Hanlin Tang, Xiangru Lian, Ming Yan, Ce Zhang, and Ji Liu. D^2: Decentralized Training over Decentralized Data. In *International Conference on Machine Learing (ICML)*, 2018.

[52] Lei Tang and Huan Liu. Graph Mining Applications to Social Network Analysis. In Charu C. Aggarwal and Haixun Wang, editors, *Managing and Mining Graph Data*. Springer US, Boston, MA, 2010.

[53] Riitta Toivonen, Lauri Kovanen, Mikko Kivelä, Jukka-Pekka Onnela, Jari Saramäki, and Kimmo Kaski. A comparative study of social network models: Network evolution models and nodal attribute models. *Social Networks*, 31(4), October 2009.

[54] Thijs Vogels, Lie He, Anastasiia Koloskova, Sai Praneeth Karimireddy, Tao Lin, Sebastian U Stich, and Martin Jaggi. RelaySum for Decentralized Deep Learning on Heterogeneous Data. In *Advances in Neural Information Processing Systems*, volume 34. Curran Associates, Inc., 2021.

[55] Wayne Zachary. An information flow model for conflict and fission in small groups. *Journal of anthropological research*, 33, 11 1976.

[56] Matteo Zignani, Sabrina Gaito, and Gian Paolo Rossi. Follow the "Mastodon": Structure and Evolution of a Decentralized Online Social Network. *Proceedings of the International AAAI Conference on Web and Social Media*, 12(1), June 2018. Number: 1.

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 list of contributions in the introduction matches the content of the paper

   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: We detail our choice of software for the datasets and of the collected material for each nodes, in particular with respect to the high privacy requirements in the Fediverse. We also let some tasks for future work.

   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: [NA]

Justification: The paper does not contains new theoretical results,

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: We release the dataset, the crawler and the code.

   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: [Yes]

Justification: The dataset is publicly available on Kaggle. We use as required Croissant and we provide a Python package to interact with the dataset as well. We also provide Kaggle notebooks demonstrating the use of our Python package and some of our experiments.

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: we provide the code of the experiments and details in appendix.

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: We report error bars when relevant.

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: we detail how the crawling was performed. The examples of computation on the datasets where performed locally on a regular laptop. Most experiments can be executed on a generic Kaggle notebook in a few hours.

   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: This work conforms with the Code of Ethics

   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: [Yes]

    Justification: The paper discusses the advantage of having datasets closer to real use-case for Trustworthy Decentralized Machine Learning. The privacy implications of the datasets are also discussed.

    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 choices made when constructing the datasets limit the risk of misuses.

    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 code from other paper is cited and we respect all the requirements for the new datasets we produce.

    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: [Yes]

    Justification: This is one of the central contributions of the paper.

    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: LLM were only used for editing purposes.

    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.