Towards Open Federated Learning Platforms: Survey and Vision from Technical and Legal Perspectives

Moming Duan, Qinbin Li, Linshan Jiang, Bingsheng He

***Abstract*—Traditional Federated Learning (FL) follows a server-dominated cooperation paradigm which narrows the application scenarios of FL and decreases the enthusiasm of data holders to participate. To fully unleash the potential of FL, we advocate rethinking the design of current FL frameworks and extending it to a more generalized concept: Open Federated Learning Platforms, positioned as a crowdsourcing collaborative machine learning infrastructure for all Internet users. We propose two reciprocal cooperation frameworks to achieve this: query-based FL and contract-based FL. In this survey, we conduct a comprehensive review of the feasibility of constructing open FL platforms from both technical and legal perspectives. We begin by reviewing the definition of FL and summarizing its inherent limitations, including server-client coupling, low model reusability, and non-public. In particular, we introduce a novel taxonomy to streamline the analysis of model license compatibility in FL studies that involve batch model reusing methods, including combination, amalgamation, distillation, and generation. This taxonomy provides a feasible solution for identifying the corresponding license clauses and facilitates the analysis of potential legal implications and restrictions when reusing models. Through this survey, we uncover the current dilemmas faced by FL and advocate for the development of sustainable open FL platforms. We aim to provide guidance for establishing such platforms in the future while identifying potential limitations that need to be addressed.**

***Index Terms*—Federated Learning, AI Licensing, Collaborative Machine Learning, Model Mining**

I. Introduction

The success of current data-driven AI relies on massive amounts of training data and follows a gather-and-analyze paradigm [1], which confronts with challenges of complying with rigorous data protection regulations such as OECD Privacy Guidelines [2] and GDPR [3]. Although data-centric AI is currently mainstream paradigm, Federated Learning [4], a novel distributed collaborative training framework, is gaining popularity in both academia and industry for its advantages in complying with privacy regulations [5].

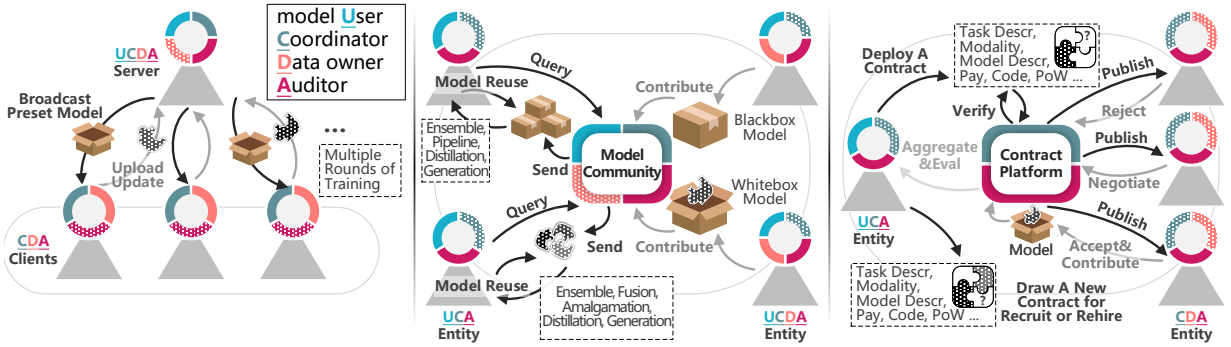
A typical workflow of FL systems is presented in Fig.1(a), where the entity with a modeling demand (FL server) first deploys the FL services, initializes the training task, and then distributes this task to participants with training data (FL clients) for modeling [6]. Based on this workflow pattern, FL frameworks have been derived with specialized improvements in communication [7]–[9], optimizaiton [10]–[12], robustness [13]–[15] and privacy [16]–[18]. While these fascinating improvements greatly enhance the utility of FL, they all follow a task-based interaction paradigm, in which an FL server dominates the cooperation. In this narrow interpretation of FL, the data owner is treated more like a worker than a collaborator and performs training primarily for the benefit of the server's goals. Due to the above defects, FL clients have little enthusiasm to participate, and the potential for redundant training also leads to low model reusability, further diminishing the efficiency of the FL systems. This explains why current FL frameworks are more akin to private distributed modeling services rather than open and sustainable modeling platforms that every user can access and benefit from, addressing many data silo applications.

In this paper, we try to answer the question: **Can we establish a sustainable open FL platform based on a novel reciprocal cooperation framework?** Obviously, to answer this quesion, it is insufficient to simply study the basic concepts of FL and investigate existing FL techniques. We also need to conduct a wide survey of potential techniques that can facilitate the construction of open FL platforms. To aid understanding, Fig. 1(b)(c) provide a first glimpse of two novel FL cooperation frameworks that we advocated:

- **Query-based FL**. It follows a loosely-coupled cooperation framework between entities (we use "entities" instead of "clients" to emphasizes equality), where any entity can freely upload their local models or query models from an open repository named *Model Community*. There are many valuable challenges that can be explored, such as how to query for models, how to reuse the retrieved models or how to transfer knowledge from these models, how to ensure the legal compliance between different model licenses, and how to protect the intelligent properties of released models (ref. §IV, §V and ‡II, ‡III, ‡IV, ‡V, where § denotes the main section and ‡ denotes the appendix section).
- **Contract-based FL**. It follows a mutual choice cooperation framework, where each entity can deploy model training contracts with specialized requirements such as task modality, execution environment, model architecture and license. Meanwhile, entities holding data can choose whether to accept the contract. We show the research topics in this setting include ML subtasks design and monetization (ref. §VII and ‡VI).

Moming Duan, Linshan Jiang, and Bingsheng He are with the National University of Singapore, Singapore, 119077. Email: {moming@nus.edu.sg, linshan@nus.edu.sg, hebs@comp.nus.edu.sg}.

Qinbin Li is with the University of California, Berkeley, USA, 94720. Email: {qinbin@berkeley.edu}



(a) Traditional FL (b) Query-based FL (c) Contract-based FL

Fig. 1. A schematic diagram of three cooperation frameworks of FL. (a) is the traditional FL platform, (b) (c) are the proposed open FL platforms. Four colors correspond to four roles in [19], and colors with grid lines indicate non-essential roles.

It is worth noting that the definitions of the four roles illustrated in Fig. 1 (i.e., model user, coordinator, data owner, auditor, ref. §III-A) are adopted for compatibility with the IEEE standard [19], and our proposals are also within the standard definitions of FL. The diagram in Fig. 1(a) illustrates the workflow of traditional FL, where all FL clients are required to accept the training schedule from the FL server and perform multiple rounds of local training and model averaging until the global model converges. In contrast, the entities in query-based FL and contract-based FL are proactive in their participation. We believe that these reciprocal cooperation frameworks have the potential to expand the prevalence of open FL and establish FL ecosystems.

A. Our Contribution

In contrast to previous surveys that primarily focused on the server-dominated cooperation framework in FL (ref. §II), our new survey explores the feasibility of reciprocal cooperation frameworks in FL. To the best of our knowledge, our work represents the first systematic survey in this area. The major contributions of this survey are as follows:

- We are the first to introduce the concept of open FL platforms by presenting two cooperation frameworks, namely query-based FL and contract-based FL, along with an overview of their key features and properties.
- We explore the query functionalities of online model repositories, such as Huggingface and OpenVINO, to investigate their feasibility for model query in query-based FL settings.
- We summarize the rights, restrictions, and enforcements of in-service model licenses and highlight the legal compliance and copyrightability issues in collaborative modeling. Additionally, we provide guidelines for selecting licenses to minimize conflicts and prevent license proliferation.
- We propose a taxonomy to streamline the legal compliance analysis in ML, which is also useful for quickly identifying suitable model reusing conceptss for open FL platforms. A comprehensive comparison of current FL studies based on this taxonomy is surveyed.
- We analyze the requirements for model protection in the context of query-based FL and identify applicable solutions from deep Intellectual Property (IP) protection. We also introduce the concept of designing ML microtasks for query-based FL.

The structure of this paper is shown in Fig. 2. We compare this survey to other related surveys and show our distinction in §II. In §III, we present the overview and point the limitations of traditional FL. We comprehensively explore the feasibility and challenges of query-based FL in §IV, which includes model query (§IV-B), model license comparison (§IV-C1) and license selection (§IV-C2). In §V, we present our taxonomy from a model reusing perspective. We introduce how to design ML microtasks in contract-based FL in §VII and we conclude this paper in §VIII. *Due to page limits, we defer some related discussions to the appendix and we use ‡ to remind you that detailed analysis can be found in the appendix.*

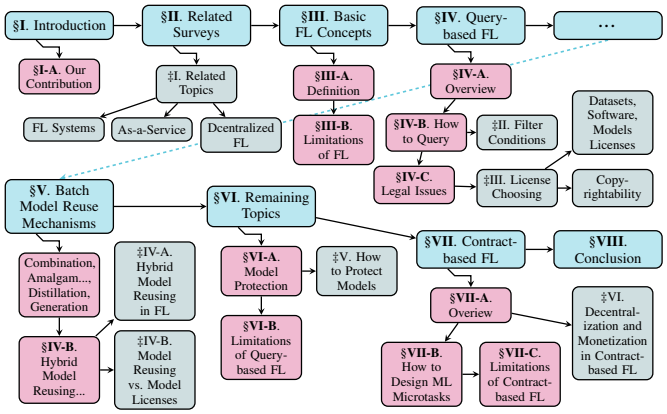


Fig. 2. Article Structure. §: Main Section Subsection, ‡: Appendix.

II. Related Surveys

Federated learning has become a buzzword in various fields, leading to the emergence of numerous FL studies. These works can be classified into three primary categories: FL systems design, FL applications and FL toolkits. The initial architectures and concepts for FL systems were summaried by Yang *et al.* [20]. They categorized FL into horizontal FL, vertical FL and federated transfer learning based on the distribution characteristics of data, which are written in IEEE Standard 3652.1-2020 [19], [21]. Following this, several surveys have emerged with a focus on enhancing FL system [4], [22]–[25]. From the algorithmic perspective, personlized FL [26], [27]

aims to learn personlized models for each client to address the challenge of statistical heterogeneity [28]. Meanwhile, the privacy-perserving computing platforms and model aggregation protocols for FL also been widely studied and summarized by [29]–[32]. Furthermore, many advanced FL architectures had been proposed, such as asynchronous [9], decentralized and blockchain-based FL frameworks [33]–[35].

Currently, most surveys extensively discuss the challenges of efficiency, heterogeneity, privacy in FL systems design, while the surveys from blockchain fields offer the most comprehensive review‡I. However, except for a few blockchain-based FL studies, most of the listed surveys just present the same story from different angles and backgrounds, i.e., a server sets the model training task and delegates it to data holders to complete. This *server-dominated* cooperation framework is a narrow implementation of FL systems. Therefore, this survey aims to fill the gap by investigating and surveying the associated tenchnologies that support more open and inclusive cooperation frameworks in FL systems, where all entities, whether they own the data or not, can benefit from it.

Distinction of Our Survey. This survey focuses on exploring the innovative FL cooperation frameworks, which involves some FL concepts such as decentralized FL, blockchain-based FL, few-shot FL, ML related platforms and services but goes beyond them. To the best of our knowledge, this is the first survey that focuses on the **cooperation frameworks** of FL‡I.

III. Basic Concepts of Federated Learning

A. Definition

Federated Learning [8], [19] is a collaborative machine learning modeling paradigm that enables sharing and aggregation of knowledge from multiple sources while maintaining the confidentiality of source data. Generally, FL systems consist of two main entities in terms of task organization: the server and the participants.

Furthermore, FL entities can also serve multiple functional roles to support advanced features such as privacy enhancement [16], [17], [36], participant scheduling [15], [37], model verification [38], [39] and incentive mechanisms [40]. Recall that there are four roles defined in the FL standard [19]:

- **Model User**. The FL model users can request for FL services and preset the targeted task, and then establish cooperation with participants who provide data. This role can leverage the benefits of collaborative training to improve the preformance of its objective models.
- **Coordinator**. The FL coordinators are responsible for providing FL services to all FL entities. This role involves setting up communication channels with entities, initializing the execution environment of participants [41], scheduling the training and aggregation workflows to improve system efficiency. Additionaly, the FL coordinator provides privacy control mechanisms [16], [30], [42] for model users and authorization verification for participants to maintain the security of FL systems. Furthermore, the coordinator can hold a validation dataset for evaluate the models contributed by participants or detect potential disturbances from Byzantine attacks [43].

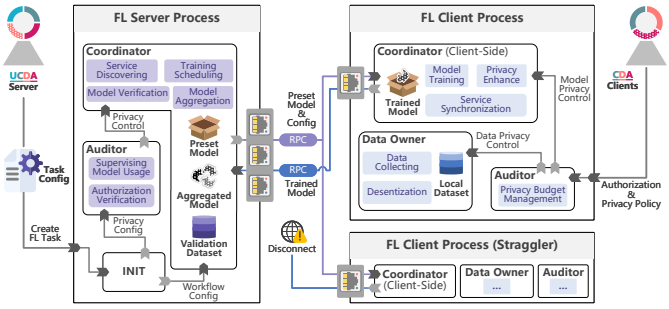


Fig. 3. An overview of traditional FL systems. (U: model User, C: Coordinator, D: Data owner, A: Auditor)

- **Data Owner**. The FL data owners are knowledge contributors of FL systems, they collect and desentize raw data to maintain a local dataset for federated training. Although they have full authority over data processing and modeling, they cannot share the raw data due to privacy concerns. To address these concerns, de-identification [44] and differential privacy [45] techniques can be applied to meet privacy budgets as required by privacy policies.
- **Auditor**. The FL auditors are responsible for formulating privacy control policies and establishing supervisory mechanisms that ensure the training process is compliant with data protection regulations (e.g., HIPAA [44], GDPR [3]) and preventing potential privacy breaches for both model users and data owners. Especially in FL, the latent knowledge in models can potentially reveal the sensitive information of training data [46]–[48], making it crucial for auditors to scrutinize the model transmission [49], [50] and verify the ownership [38], [39].

Fig. 3 illustrates the typical architecture of traditional FL systems, which consists of server part and client part as a distributed modeling toolkits. In a general setting, the server part is the central aggregator installed in a trusted cloud environment, while the client part of software can operate in different operating environments on client devices. The server and clients are connected via Internet and typically with the help of Remote Procedure Call (RPC) interface for coordinating [51]–[56]. We use four colors to represent the four FL roles and the colors with grid lines indicate non-essential roles. For example, in Fig. 3, the UCDA server takes on the roles of model user, coordinator and audior. However, there is no necessary to hold training or validation data, so the role of data owner is non-essential.

To illustrate traditional FL workflow, we leverage the vanilla FL framework Federated Averaging (FedAvg) [6], [8] as an example. First, the FL server pre-defines the objective modeling task and initializes the server process. Secondly, the coordinator in server-side specifies a preset global model and the operational parameters. Thirdly, the coordinator discovers the availability of clients' FL services, boardcasts the global model and training config to them. The training configuration contains bath size, local epoch round, optimizer parameters and so on. Then, the coordinator will wait for the trained results contributed by the coordinator in clients-side and drop those clients with network problems. Finally, the server aggregates the trained resultes received from various clients

into the global model and begins a new round based on this aggregated global model. The aggregation strategy adopted in FedAvg is the weighted average of model parameters based on the size of the local dataset, which means the global objective of FL can be regarded as a joint objective function of clients. Although the auditor component was not included in prototype FedAvg, it plays an important role in the business-ready FL frameworks [53], [57], [58].

However, in comparing FedAvg workflow described above with Fig. 3, it is easy to notice that the client part has been excluded. In traditional FL, the client-side process is tightly coupled with server-side process, leaving clients to either fully accept or reject the training schedule imposed by the server with compromised audit control. So the clients are not considered as an autonomous entities but rather work as subordinates to server. From this perspective, we summarize the limitations of traditional FL in the next section, which motivates us to explore more innovative sustainable FL cooperation frameworks.

B. Limitations of Traditional FL

Previous surveys [4], [20], [23], [27], [33], [35], [59], [60] has extensively discussed the challenges in FL systems from various aspects. However, the cooperation mechanism of FL systems has been overlook because almost all mainsteam FL frameworks follow a same prototype [8], which shapes the current FL form: a modeling software. We summarize three inherent limitations of traditional FL cooperation mechanisms: (1) **Server-client Coupling**, (2) **Low Model Reusability**, (3) **Non-public**.

1) Server-client Coupling: The invasive software deploy mode compromises the integrity of client environments and exposes them to new privacy risks. Specifically, the coordinator components (client-side) pushed by the server may not offer demanded privacy control mechanisms [8], [61], [62], or cause resource depletion on client-side [6], [36], [63], or even piggyback malicious executable codes [64]. So the auditor role of client is non-essential as depicted in Fig. 3, not only because the client maybe lacks a corresponding policy for FL training, but also because its privacy is not completely under its control. Likewise, the malicious clients can also exploit the vulnerability in the aggreagation strategy to currupt the FL tranining process [43], [65]–[67] or insert backdoors [68], [69]. In addition, the unstable network environment can drive clients to drop out from traning (i.e., straggler effect), thereby reducing system efficiency [66], [70]. Therefore, the server-client coupling design of traditional FL systems make them susceptible to unpredictable runtime environments, leading to system vulnerability and low reliability.

2) Low Model Reusability: The traditional FL scheduling follows a task-centric manner and terminates once the training reaches a preset number of rounds or meets traget metrics on global model set by FL server [6]. As a result, only FL server can guarantee having the latest global model after the task is terminated. This ad-hoc modeling paradigm results in low model reusability and transportability. Since only FL server is able to maintain the complete modeling trajectory, it is difficult for the client to roll back the training itself to eliminate the potential privacy risk. Meanwhile, the non-deliverable tasks scheduling mechanism also hinders inter-task model reuse.

3) Non-public: Except PySyft [58], the application scenarios of mainstream FL frameworks [52]–[55], [57], [61], [62], [71] aim to provide private collaborative ML traning service, and there is no any accessible FL platform for the public[‡I-A]. Although there have been real-world deployment practices of FL for the public with scales of millions [6] and billions [36], these have been carried out only by tech giants with a massive base of active users. For an individual user, there is no practical way to organize such a large-scale FL training network.

Due to the limitations in the cooperation mechanism mentioned above, data owners are not sufficiently motivated to participate in this server-take-all FL training network even if it is public accessible. Therefore, the cornerstone of buliding a sustainable open FL platform is to create a reciprocal FL cooperation framework, followed by corresponding multi-source knowledge aggregation strategies, which we survey through the following two themes:

In the query-based FL platform, which is an open model sharing and reusing platform empowered by the community for model mining, we explore a wide range of valuable topics, including the availability of up-to-date model repositories for model querying, legal compliance analysis between different model licenses, and copyright issues and IP protection in model reusing. In the contract-based FL platform, which is an opt-in model training network that allows multi-round communication and monetization, we explain how to select training frameworks according to microtasks and discuss the pros and cons of Web3-based methods. We will detail discuss these topics in the following sections.

IV. Query-based Federated Learning

A. Overview

Let us continue by establishing a sustainable open FL platform based on a query-based cooperation framework. An overview of this platform is presented in Fig. 4, the desin philosophy behind this framework is to break the coupling between FL server and clients. In the query-based FL systems, all traditional FL roles and components are maintained on an open model repository called Model Community. The Model Community privdes a one-stop ML models redistribution and reuse service, including model indexing, automatic batch model reuse, license management, privacy control and so on. In addition to large-scale pretrained models like BERT [72], BLOOM [73] with great generalization abilities, we also encourage individuals to upload their task-specific models trained on limited domain data to boost the knowledge mining within models, aka model mining [74]. The derivatives of model mining can learn representations from multiple domains, resulting in more promising performance that can be evaluated by platform users. Furthermore, the contributors can release models under applicable licenses, granting them distribution control and legal protection of their intellectual property (IP). In summary, the properties of query-based FL are: (1) **Model Agnostic**, as there are no restrictions

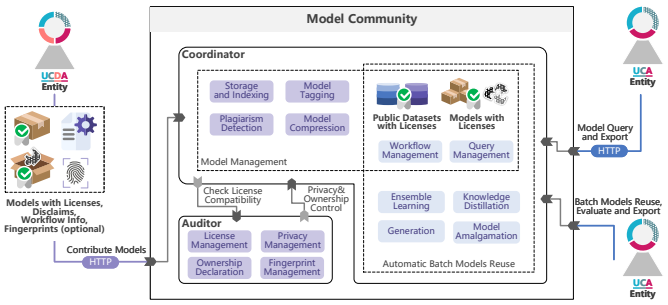


Fig. 4. An overview of query-based FL systems. (U: model User, C: Coordinator, D: Data owner, A: Auditor)

on the types and architectures of the models uploaded by users; (2) **Contactless**, as communication channels need not be maintained; (3) **Community-powered**, whereby sharing models enrichs the entire community.

Actually, we aim to advocate a novel Software-as-a-Service (SaaS) [75] ML platform with automatic batch model reuse integrated, which has potential to leverage the transportability of models to address previously unexplored ML problems. Due to the high computational demands of deep learning, current ML platforms primarily concentrate on computing, for example, MaaS, MLaaS, FLaaS provide ML models deployment and development services to handle user-specified tasks[‡I-B]. On the other hand, there are several ML platforms provide open model search and download services. So, can we leverage leverage off-the-shelf public model repositories to build a query-based FL system? Unfortunately, these sites are designed solely for sharing and are not suitable for more advanced functionalities such as model ensemble [76] and knowledge distillation (KD) [77], and we will explain the reasons in the following section.

B. How to Query for Models

To establish a query-based FL platform, the first thing that comes to mind is how to query for models. Unlike traditional ML model sharing repositories that mainly query for a specific model by name, it requires an efficiency approach to export a batch of target models that are ready for ensemble or distillation. We summarized the filter conditions of existing DNNs sharing repositories[1] in TABLE I. The prevailing method for querying models involves searching for the desired model by its name, used datasets, and the associated tasks. For instance, one might search for the model name GPT [78], models trained on the MNIST dataset [79], or models capable of performing image segmentation tasks. However, this model retrieval method requires the users have a strong priori knowledge in data science, thus raising the barrier for knowledge mining within models. As an example, there is no effective way to acquire a batch of image classfication models that contains the knowledge of *lesser panda* for further distillation. A compromise solution is to manually search the schema of each dataset one-by-one and subsequently search for models trained on those datasets.

As shown in TABLE I, most DNNs repositories are simply listing the description of input/output (e.g., NVIDIA NGC, OpenVINO) or even just present the source codes (e.g., Tensorflow Hub, Pytorch Hub). This lack of unified convention for model input/output poses a challenge for query-based FL. Additionally, most of DNNs repositories do not enable querying models by licenses, resulting in the cumbersome task of individually handling model licenses and ensuring compatibility among different licenses. Hence, it is imperative to reconsider the design of DNNs repositories to enable quick identification of readily reusable models for knowledge aggregation. We further suggest following filter conditions for query-based FL: 1) Data Description; 2) Workflow and History; 3) Software Dependency; 4) Fairness and Robustness[‡II].

The aforementioned filter conditions provide comprehensive coverage of the ML modeling process. However, there are additional requirements depending on the reuse mechanisms in the model retrieval side. For example, FedAvg [8] aggregates the local models weights element-wise, which requires full access to the models. In contrast, MoE [76] only ensembles a batch of model outputs, so the individual models can remain blackboxes in this scenario. So, in the context of software licenses or model licenses, the batch models reused by FedAvg should be released as source code, while those reused by MoE can be released as binary executable modules (e.g., dynamic linking). The above distinction is crucial for ensuring that model reuse results meet the legal framework, and this has been overlooked in traditional FL. We will expand on this topic in the following section.

C. Legal Considerations in Batch Model Reuse

Once we have acquired a batch of models that can contribute to the new target task, the next step is to reuse the knowledge of these pre-trained models, i.e., transfer their knowledge from source domain to the target domain [80]. However, before deciding on how to reuse the model, it is important to ensure that the rights and permissions have been obtained. This may involve reviewing the terms and conditions of the licenses under which the models were released or obtaining permission from the original creators or copyright holders. Therefore, in this section, we will not focus on the technical details of how to reuse models, which is already covered by many related surveys, such as Transfer Learning [80], Ensemble Leanring [81], Domain Adaptation [82], Knowledge Distillation [83], Deep Generative Models [84] and Model Fusion [85]. Meanwhile, the specific model reuse techniques used are at the user's discretion, and the query-based FL platform is not bounded or restricted to any particular reuse method. Innovatively, we study how to reuse batch of models, from the perspective of **legal compliance**.

The machine learning community benefits from the openness of ideas and code, and many high-impact ML conferences and journals encourage authors to publish their source code and dataset to research platforms like Papers With Code and

[1]Hugging Face: https://huggingface.co; Model Zoo: https://modelzoo.co; Tensorflow Hub: https://tfhub.dev; NVIDIA NGC: https://catalog.ngc.nvidia.com/models; OpenVINO: https://docs.openvino.ai/latest/model_zoo.html; Pytorch Hub: https://pytorch.org/hub

TABLE I
FILTER CONDITIONS AND CHARACTERISTICS OF DNNS REPOSITORIES. ✓: SUPPORTED, ✗: UNSUPPORTED, ! : INFORMATION PROVIDED BUT UNSEARCHABLE, LISTED IN DESCENDING ORDER BY NUMBER OF RELEASED MODELS. (ACCESSED ON JANUARY 17, 2024)

	DS Name	Model Architecture	Modality/Task	Tag	License	Input-Output	Batch Export	# of Models
Hugging Face	✓	✓	✓	✓	✓	!	✗	470,263
Model Zoo	✓	✓	✓	✓	✗	✗	✗	3,245
Tensorflow Hub	✓	✓	✓	✓	!	!	✗	2,186
NVIDIA NGC	!	✓	✓	✓	!	!	✗	680
OpenVINO	!	✓	✓	✗	!	!	✓	277
Pytorch Hub	!	✓	✗	✗	✗	!	✗	52

Code Ocean[2] to increase exposure and facilitate reproducibility. To restrict the use of ML techniques for unethical purposes (i.e., Deepfakes [86]) and protect the IP of creators, models are typically published under a license agreed upon by the licensor. Here, we summarized the granted rights, restrictions and enforcements of licenses for ML models posted on Hugging Face in TABLE II. The following sections will provide a detailed survey of these licenses.

1) Model Licensing Forms: As shown in TABLE II, ML models are licensed in three main forms: as software (e.g., Apache, MIT, GPL), as a model (e.g., OpenRAIL), and as content/database (e.g., CC BY, PDDL). The reason for the mixed use of licenses is the ambiguity in the dependency relationship between the ML code, model, and data. Thinking in terms of software, ML models can be released with reproducable code and considered as a component of software. So many Free and Open Source Software (FOSS) licenses [101] are naturally deferred for licensing of models. The most popular license is Apache-2.0, which is a permissive FOSS license that allows the freedom to make derivative works. However, the model building process also relies on a massive amount of data [102] that may be licensed under different licenses, which can lead to license conflicts. A real-world example is BERT [72], which was published under the Apache-2.0 license but pre-trained on English Wikipedia documents that are licensed under CC BY-SA 3.0. This changing of license violates the requirement of the CC BY-SA 3.0, which states that any contribution must be distributed under the **same license** as the original work.

Thinking in terms of content and database, some word embedding models like GloVe [103], compute words representations based on licensed open linguistic resources. These representations can be regarded as a translation of corpus and fall under the license of the original linguistic resources. A more complex scenario arises when the model is fine-tuned with other data that has a different license, for example, fine-tune RoBERTa [104] (licensed under MIT) with SQuAD2 [97] (licensed under CC BY 4.0). The tuned model can be interpreted as both derived works and combined works.

Not only limited to protecting the IP and controlling the diffusion of ideas, but AI companies and researchers are also concerned about licensees using their models for unethical purposes [105]–[107], which is usually not restricted by traditional licenses designed for software and content. We can infer the concerns of unethical use of GPT-2 [78] from its modified MIT license granted by its inventors, which states, *We don't claim ownership of the content you create with GPT-2, so it is yours to do with as you please. We only ask that you use GPT-2 responsibly and clearly indicate your content was created using GPT-2.* However, such a statement lacks legal enforcement, and users may avoid accountability by convincing themselves that despite their efforts to minimize harm, they could not predict the AI artifact they generated would be used for harmful purposes. On the other hand, the original licensing frameworks for software and content (e.g., MIT, CC BY) are not well suited to the data-driven ML. Many ML operations, such as training, fine-tuning, inference, and distillation, are not explicitly defined in these license clauses, leaving a potential legal loophole for licensees.

To address the unique challenges and considerations surrounding the use and distribution of ML models, several specific licenses for ML models have been proposed. CreativeML OpenRAIL-M license, proposed by Responsible AI [108], is the most popular model-specific license on Hugging Face and enables legally enforceable responsible use. By accepting this license, licensees must adhere to the use-based restrictions stated by the licensor, and these restrictions must also apply to derivative works. With a multitude of different model licenses available, it becomes a challenging and tedious work to reuse them in bulk. It is, therefore, imperative to establish guidelines for selecting the licenses for models and other related components that are ready for query-based FL.

2) License Choosing: In query-based FL, the model community aims to promote the reuse of models contributed by users, which raises unique concerns about model licensing:

- A model license ready for open FL paltforms should allow the **modification, combination and redistribution** of original works and any derived works.
- **Sublicensing** right should be granted to enable the republication of derived works after knowledge mining.
- Some licenses enforce the source of the derived works to be **disclosed** and prohibit their **commercial use**, which hinders model selling [109].
- Some licenses are **copyleft** (marked with * in TABLE II), which means the derivatives must be licensed under the same license or a compatible license, leading to potential license conflicts and proliferation [110].
- All granted rights are preferably **irrevocable** by the licensors [111].

Furthermore, it is important to consider the licensing of two other components when building and reusing models: data and algorithms, which may have entirely different license terms. Here, guided by the comparisons between different licenses outlined in TABLE II, we can summarize the preferences for selecting licenses in query-based FL to minimize conflicts[‡III]. Once we obtain the right to relicense the modification models,

[2]https://paperswithcode.com; https://codeocean.com

TABLE II
LICENSES FOR ML MODELS AVAILABLE ON HUGGING FACE WITH A FOCUS ON THEIR RIGHTS, RESTRICTIONS AND ENFORCEMENTS, GROUPED BY FOSS LICENSES, AI MODEL LICENSES, FREE CONTENT OR DATABASE LICENSES IN DESCENDING ORDER OF NUMBER OF MODELS (GPL, BSD, LGPL, CC LICENSES WITH UNSPECIFIED VERSIONS ARE EXCLUDED, THE SIMILAR REVISIONS ARE MERGED). ✓: PERMITED OR REQUIRED, ✗: NOT PERMITED OR NOT REQUIRED, ! : NOT EXPLICITLY PERMITED, * : COPYLEFT LICENSE, † : PUBLIC DOMAIN LICENSE. ONLY THE SOURCE CODE OF THE ORIGINAL WORK UNDER AFL-3.0 OR ARTISTIC-2.0 IS REQUIRED TO BE DISCLOSED. YOU MAY NOT DISTRIBUTE THE MODIFIED MATERIALS LICENSED UNDER CC-BY-NC-ND OR CC-BY-ND. (ACCESSED ON JANUARY 17, 2024)

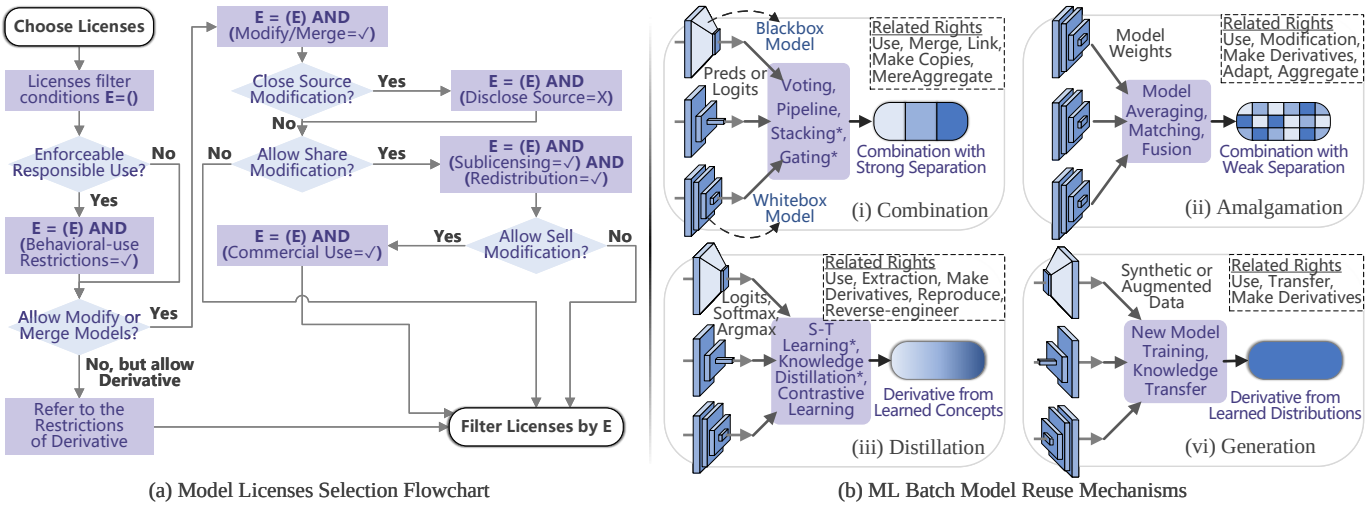
Licenses	Modify / Merge	Redistribution	Sublicensing	Commercial Use	Patent Use	Trademark Use	State Changes	Disclose Source	Responsible-use Restrictions	License/Attribution Preservation	# of Models	Licensed Materials / Remarks
Apache-2.0	✓	✓	✓	✓	✓	✗	✓	✗	✗	✓	65,985	BERT [72]
MIT	✓	✓	✓	✓	!	!	✗	✗	✗	✓	30,344	GPT-2 [78]
AFL-3.0	✓	✓	✓	✓	✓	✗	✓	(✓)	✗	✓	2,208	Italian-Legal-BERT [87]
*GPL-3.0	✓	✓	✗	✓	✓	✗	✓	✓	✗	✓	1,242	PersonaGPT [88]
Artistic-2.0	✓	✓	✓	✓	✓	✗	✓	(✓)	✗	✓	675	Include original source
BSD-3-Clause&-Clear	✓	✓	✓	✓	!	✗	✗	✗	✗	✓	636	CodeGen [89]/ A MIT-style license
†WTFPL-2.0	✓	✓	!	✓	!	!	✗	✗	✗	✗	409	A MIT-style permissive license
*AGPL-3.0	✓	✓	✗	✓	✓	✗	✓	✓	✗	✓	265	Extended GPL covers SaaS
†Unlicense	✓	✓	!	✓	!	!	✗	✗	✗	✗	254	A MIT-style permissive license
*GPL-2.0	✓	✓	✗	✓	!	!	✓	✓	✗	✓	91	Not compatible with GPL-3.0
*LGPL-3.0&2.1	✓	✓	✗	✓	✓	!	✓	✓	✗	✓	84	For software libraries
BSD-2-Clause	✓	✓	✓	✓	!	!	✗	✗	✗	✓	82	A MIT-style permissive license
BSL-1.0	✓	✓	✓	✓	!	!	✗	✗	✗	✓	77	A MIT-style permissive license
*OSL-3.0	✓	✓	✓	✓	✓	✗	✓	✓	✗	✓	55	Linking is not derivative work
*Ms-PL	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	43	Weak copyleft license
ECL-2.0	✓	✓	✓	✓	✓	✗	✓	✗	✗	✓	38	For education communities
Zlib	✓	✓	!	✓	!	!	✗	✗	✗	✓	30	Rename if modified
*MPL-2.0	✓	✓	✓	✓	✓	✗	✓	✓	✗	✓	23	State changes under MPL only
*EPL-2.0&1.0	✓	✓	✓	✓	✓	!	✗	✓	✗	✓	19	Can link proprietary license code
ISC	✓	✓	!	✓	!	!	✗	✗	✗	✓	15	MIT-style license w/o sublicense
*EUPL-1.1	✓	✓	✓	✓	✓	✗	✓	✓	✗	✓	15	License of EU covers SaaS
NCSA	✓	✓	✓	✓	!	✗	✗	✗	✗	✓	10	Include full text of license
PostgreSQL	✓	✓	!	✓	!	!	✗	✗	✗	✓	7	A MIT-style license
OpenRAIL	>Responsible AI License template, w/o full text										22,947	ControlNet [90]
CreativeML-OpenRAIL-M	✓	✓	✓	✓	✓	✗	✓	✗	✓	✓	15,591	Stable Diffusions v1 [91]
Llama2	✓	✓	✗	(✓)	✓	✗	✗	✗	✓	✓	3,538	Llama 2 [92]
OpenRAIL++	>Same as CreativeML-OpenRAIL-M										1,433	Stable Diffusion v2 [91]
BigScience-OpenRAIL-M	>Same as BigScience-BLOOM-RAIL-1.0										659	A general version of 1.0
BigScience-BLOOM-RAIL-1.0	✓	✓	✓	✓	✓	✗	✓	✗	✓	✓	527	BLOOM [73]
BigCode-OpenRAIL-M	>Same as BigScience-BLOOM-RAIL-1.0										320	StarCoder [93]
OPT-175B	✓	✗	✗	✗	✗	✗	✗	✗	✓	✓	≈ 94	OPT LLM [94]
SEER	>Same as OPT-175B, ban on reverse-engineer										≈ 23	SEER Vision Model [95]
CC-BY-NC-4.0&3.0&2.0	✓	✓	✗	✗	✗	✗	✓	✗	✗	✓	4,747	GALACTICA [96]
CC-BY-4.0&3.0&2.5&2.0	✓	✓	✗	✓	✗	✗	✓	✗	✗	✓	3,429	RoBERTa-SQuAD2.0 [97]
*CC-BY-NC-SA-4.0&3.0&2.0	✓	✓	✗	✗	✗	✗	✓	✓	✗	✓	1,783	LayoutLMv3 [98]
*CC-BY-SA-4.0&3.0	✓	✓	✗	✓	✗	✗	✓	✓	✗	✓	1,510	LEGAL-BERT [99]
CC-BY-NC-ND-4.0&3.0	(✓)	✗	✗	✗	✗	✗	✗	✗	✗	✓	406	NonCommercial, NoDerivatives
†CC0-1.0	✓	✓	!	✓	✗	✗	✗	✗	✗	✗	330	BlueBERT [100]
C-UDA	✓	✓	✓	✗	!	!	✗	✗	✓	✓	72	Data for computational use only
†PDDL	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗	50	Database-specific license
CC-BY-ND-4.0	(✓)	✗	✗	✓	✗	✗	✓	✗	✗	✓	47	Disallow making derivatives
*GFDL	>Same as GPL, a free document license										30	txtai-wikipedia
*ODbL	✓	✓	✗	✓	✗	✗	✓	✓	✗	✓	20	Automatic relicensing
*LGPL-LR	✓	✓	✗	✗	!	!	✓	✓	✗	✓	19	LGPL for linguistic resources
ODC-By	✓	✓	✗	✓	✗	✗	✗	✗	✗	✓	15	Automatic relicensing

the choice of a new license depends on the application scenario of models. We further provide a flowchart in Fig. 5(a) to guide the license selection in the context of model query and model reusing. In addition, to enhance the copyrightability of the reused models, we should avoid using any derivatives and generated content from models under proprietary licenses throughout the ML reusing lifecycle[‡III-D].

For now, we have provided a comprehensive perspective and suggestions regarding the regulations and legal issues related to batch model reusing with only one piece missing: the definition of terms and corresponding clauses for different reusing mechanisms in different licenses. The terms definition for model reusing in different licenses is a novel and interesting issue that is rarely discussed. For example, interpreting model reusing as creating derivatives or combinations would involve different clauses in the licenses. To provide a better understanding of these implications, let's first provide an overview of typical model reuse mechanisms.

V. BATCH MODEL REUSE MECHANISMS

In this section, we delve into the typical batch model reuse mechanisms from a technical perspective and introduce a new taxonomy designed to address the mismatch between license terms and technical terms. Therefore, instead of summarizing the batch model reuse mechanisms from a technological and algorithmic aspect, we propose grouping these mechanisms



(a) Model Licenses Selection Flowchart

(b) ML Batch Model Reuse Mechanisms

Fig. 5. (a) Flowchart for model licenses selection in the context of model query and model reusing. (b) Proposed taxonomy categorizing batch model reuse mechanisms based on the reused results.

based on the classification of their resulting outputs for ease of justifying license clauses. As shown in Fig. 5(b), there are four categories of batch model reuse mechanisms: **Combination, Amalgamation, Distillation, and Generation**, each resulting in different forms of outputs and may correspond to different regulations in licenses.

A. Combination

Combination [81] is a straightforward way to reuse batch of models (base learners), in which multiple models jointly contribute to the output by combination strategies such as averaging, voting, stacking [76], [112]. For regression estimates, averaging can improve the generalization by taking the mean of the outputs of all weak learners in a population. Additionally, the outputs of each learner can be weighted by extra parameters [113], which can be determined by stacking estimators [112], Bayesian approach [114] or backpropagation of gating networks [76]. Voting is a workaround strategy for classification tasks and also applicable for stacking and gating. Both stacking and gating rely on a holdout or validation dataset for calculating extra parameters, marked as * in Fig. 5(b). The difference is that gating can adapt the weights of each model's estimation based on the inputs, providing better generalizability performance of the combined model.

There are many advantages of combination mechanisms from the perspective of FL. First, the input spaces of base models can be unaligned, which is ideal for the scenario of vertical FL [115] where each client may have inconsistent features in their data. Secondly, especially for query-based FL, it can simultaneously support multiple types and heterogeneous models, which means that it does not rely on any prior assumptions of the models, such as whether they are DNNs or decision trees, released with raw weights (whitebox) or binary forms (blackbox). Thirdly, the tasks of models can be different if we pipeline the base models end-to-end, which is usually overlooked as a combination mechanism of models. Pipelining can fully leverage the transferability of models to solve previously unexplored ML problems. For instance, Gao *et al.* [116] proposed a zero-shot dense retrieval system named HyDE by pipelining a natural language generation (NLG) model [117] and a natural language understanding (NLU) model [118]. The generated content, which may lack factual grounding, from the NLG model is used as query embeddings to facilitate real document retrieval by the NLU model. Similarly, through query-based FL, we can query a vicarious NLG model for a novel scenario, such as ProGen [119] for protein sequences generation, and quickly adapt this system to proteomics. Not limited to that, we can query a batch of NLG models by a well-chosen filter condition and then combine models through averaging or gating to significantly expand the exploration space for knowledge discovery.

Lastly, the combinated models have strong separation from each other, meaning that we can add or remove a batch of models without significant changes to the remaining ones. Meanwhile, combination mechanisms do not rely on the transparency of models and support blackbox sharing. Thus, the base model can establish loose connections with other models only through run scripts, providing revocability of such combination and circumvention of the restrictions of licenses. On the other hand, instead of being treated as a challenge for FL [28], the statistical heterogeneity and model heterogeneity nature of these crowdsourced models can actually enrich population diversity, which is crucial for creating a good ensemble [120], [121].

However, the storage consumption of combined models increases linearly with the number of base learners, which can strain the communication resources of a collaborative model training network. As a result, alternative approaches involve amalgamating or merging multiple models to create a new consensus model. In the following, we provide a summary of these methods.

B. Amalgamation

Amalgamation involves combining models through model parameters granularity operations, such as median [122], [123] and coordinate-wise averaging with consideration of heterogeneity [8], [10], security [124], scalability [125], matching [126], [127], specificality [128], generalizability [129],

resulting in a combination with weak separation. This reusing approach is widely used in FL works and is often referred to as "aggregation" procedures for local models. However, in order to avoid confusion with the term "combination," which is frequently used interchangeably with "aggregation" in software licenses (e.g., Artistic, GPL), we opt to use the term "amalgamation" instead.

FedAvg [8] is the vanilla model averaging method in FL with many follow-up works. For instance, Sun *et al.* [124] introduced applying norm thresholding of local model updates to defend against backdoor attacks. Similarly, Blanchard *et al.* [122] proposed using more robust median-based amalgamation for resilience against Byzantine behavior. Consider the ordering of parameters, Wang *et al.* [126] match and average the NNs parameters layer-wise across clients, based on their similarities. Yu *et al.* [127] further attribute the misalignment in FL models to the non-IID training data, and propose allocating an independent structure for each class and updating models through a feature paired averaging strategy.

While amalgamation mechanisms can achieve a balance between model performance and resource efficiency by maintaining only one global model, they often rely on multiple rounds of communication to converge, which is not applicable in a query-based FL scenario. Therefore, instead of trying to directly concatenate multiple sources of models while dealing with intricate parameter mismatch, transferring the latent knowledge learned by local models to a new model is a good alternative (i.e. Federated Distillation [130], [131]).

Another direction of model amalgamation is leveraging Bayesian nonparametrics to learn the shared global latent structures among local models [132]–[134]. These methods, known as Model Fusion, can identify distributions of neural components across local models and only fuse the components with the same distribution, which can be regarded as a model compression between FedAvg (coordinate-wise averaging) and combination (w/o averaging). However, the model fusion strategies rely on multiple communication rounds to boost the fusion efficiency, and the model performance of one-shot fusion is even worse than that of Ensemble. Most recently, Su *et al.* [135], inspired by null-space in continual learning [136], [137], propose MA-Echo which leverages layer-wise projection matrices to preserve the original loss of local models after amalgamation. Their results present a moderate improvement in one-shot setting compared to FedAvg and ensemble.

Unfortunately, this improvement is not consistently observed in multiple-round experiments. Meanwhile, to tackle the issue of catastrophic forgetting, FedPR [138] follows similar ideas to facilitate the server's learning of visual prompts from clients for MRI reconstruction applications, but the improvement is limited even in multi-round setting.

It is worth noting that our taxonomy is based on the form of the resulting model, which may not be entirely consistent with the terminology used in the technical perspective. For example, Bayes Model Averaging (BMA) [114] estimates posterior probabilities of each model given the observed data, which results in a separable weighted model. Therefore, it should be classified as Combination instead of Amalgamation like FedAvg. This novel taxonomy method is useful for analyzing compatibility with licenses. For example, the coordinate-wise operations or the fusion of model parameters generate fine-grained combinations of models that are almost irreversible, which corresponds to clauses such as adapt, modify, dynamic link, and etc., in software licenses.

C. Distillation

Distillation was initially proposed by Hinton *et al.* [77] to transfer knowledge from a batch of independently trained neural network models (Specialists) to create a new Generalist model. Their motivation was to explore the parallelization of training of specialists and improve the efficiency of distributed NNs modeling [176]. Each specialist only learns fine-grained distinctions of a subset of classes, which is very similar to the non-IID setting in FL [177]. By using Knowledge Distillation (KD), we can also compress wide and deep teacher networks into lightweight student networks [178], which is promising for addressing system heterogeneity in cross-device FL [179]. Therefore, it is natural to extend the KD technologies to FL filed [146]. Many recent FL works [153], [155], [180] solely leverage KD without following the model averaging paradigm of FedAvg, we leave this discussion for later.

Despite directly retraining a Generalist model through KD, an alternative approach is to construct an ensemble of knowledge. For example, Furlanello *et al.* [181] consecutively generate student models with the guidance of knowledge distilled from earlier generations and find that the ensemble of multiple generations of internal models achieves state-of-the-art performance. Dvornik *et al.* [182] leverage the distilled knowledge from each learner to encourage cooperation and prediction diversity within the population, which leads to better ensemble results. In the context of FL, the *main advantage* of distillation is the decoupling between KD and knowledge learning. This allows us to split the model architecture for the purpose of system heterogeneity and efficiency [183], [184]. Moreover, the well-learned knowledge from clients only needs to be communicated once [162], [166], while the server can perform multiple epochs of local training to complete the transfer.

The drawback of KD is that it is data-dependent and the shared knowledge may be extracted from local sensitive data, which exposes a new attack surface for potential model inversion attacks [185], [186]. To mitigate this issue, some efforts [174], [175] have been made to add differential privacy noise [45] to the shared content.

In general, there are three mechanisms for avoiding the sharing of sensitive knowledge. First, push the KD procedure to the server-side, where the knowledge of local models is transferred to the global model through a validation set [139] or unlabeled dataset [140]–[144] held by the server. Second, we can keep the KD procedure at the client-side, allowing the knowledge of the global model [12], [145]–[151], other clients' models [152]–[155] or self-model [12], [145], [146], [156]–[159] to be transferred based on the local training data. In the above two strategies, only model parameters are exchanged in the training network, which means they can provide the same level of privacy protection as traditional FL.

The last mechanism is to assume that a public unlabeled dataset, which is non-sensitive, is accessible by both the

TABLE III
SUMMARY OF PRIVACY-PRESERVING **DISTILLATION** WORKS IN THE FIELD OF FL. SOME WORKS ARE LISTED MULTIPLE TIMES BECAUSE THEY CONTAIN MULTIPLE KD PROCEDURES WITH DIFFERENT STRATEGIES. WORKS WITH NAMING CONFLICTS ARE DISTINGUISHED BY SUBSCRIPT.

Strategies		FL Studies
KD@Server	w/ Validation Set	FedED [139]
	w/ Unlabeled Data	FedDF [140], One-shot FL [141], FedBE [142], PerAda [143], FedET [144]
KD@Client w/ Local Data	from Global Model	FedFusion [145], FedKD$_2$ [146], MOON [12], FedNTD [147], FedMLB [148], FedCAD [149], FedAlign$_1$ [150], FedAlign$_2$ [151]
	from Other Clients' Model	FedMatch [152], CCL [153], FedProto$_1$ [154], FedProto$_2$ [155]
	from Self Model	FedFusion [145], FedDistill [156], FedKD$_2$ [146], MOON [12], FCCL [157], pFedSD [158], RSCFed [159], CCL [153]
KD w/ Public Unlabeled Datasets		FedKT [160], FedMD [161], FedAD [162], FedMD-NFDP [163], FCCL [157], FedAUX [164], RHFL [165], FedKD$_1$ [166], Cronus [167], KT-pFL [168], DS-FL [169]
KD w/ Generated Data (DFKD)		DENSE [170], FedCAVE-KD [171], FedGen [172], FedFTG [173]
KD w/ Differential Privacy		FedKC [174], FedSSL [175]

server and clients for KD [157], [160]–[169]. Sharing these extracted contents will not raise any privacy concerns, and only minimal communication is generated during KD for the purpose of aligning sample IDs. In cases where such public datasets are not available on the server, a recent approach known as Data-Free Knowledge Distillation (DFKD) [187] regenerates batches of data based on layer activation statistics or spectrum coefficients collected during training phase. Then, this synthetic data is used for distillation. DENSE [170] is the first attempt to extend DFKD to FL. It leverages the ensemble of local models to guide the training of a data generator on the server, and the generated data is then used to distill the knowledge from local models to the global model. FedCAVE-KD [171] leverages locally trained conditional autoencoders (CVAEs) [188] to generate samples based on the data distribution of clients. These CVAEs are sent to server used to construct a global generator via KD, which will later provide synthetic training data for the global discriminator.

It is worth noting that in the query-based FL setting, direct access to the original data is not available, thus the second mechanism mentioned earlier cannot be directly applied. A circumvention method is to train a generator following the inspiration of DFKD. Fortunately, this is praticable if the workflow and history information of modeling are tracked and queryable, as we advocated in §IV-B. Recalling that, as shown in Fig. 5(b), Generation is the last category in our taxonomy. Actually, such a hybrid model reuse strategy is quite common in FL. For example, the previously mentioned DENSE [170] incorporates three model reuse mechanisms: Combination (creating an ensemble), Generation (generating synthetic data), and Distillation. Therefore, our taxonomy can cover traditional FL works, such as FedAvg and MOON [12], as well as the broad sense FL, including Federated Distillation [130], [131] and Ensemble Learning [189], [190]. We provide a comparison of these hybrid works in §V-E. The summarization of above privacy-perserving KD works is given in TABLE III.

D. Generation

Generation is designed to generate synthetic samples that resemble the original data distribution by building a probabilistic model [191] or deep learning model [84], [188], [192] that can capture the underlying distribution pattern and latent structure of original data. Generally speaking, generation techniques can be classified into three categories: data-level, probabilistic, and representation-based approaches. Data-level approaches involve sample granularity operations such as interpolation [193], [194] and augmentation [195] to generate synthetic features based on the original feature space of data and share. Even though these methods are training-free and easy to implement, they cannot be directly applied to the FL setting due to privacy concerns. Recent proposed FedMix [196] aims to alleviate the negative effect of non-IID data by using *mixup* [194], where the average of local data is linearly interpolated with the training data to generate augmented samples. However, the potential risk of data leakage when sharing the mixup data is not comprehensively evaluated in the original work.

Probabilistic approachs aim to estimate the real data distribution using probabilistic method. For example, Markov Chain Monte Carlo (MCMC) [191] methods construct a Markov Chain that converges to the desired target distribution by iteratively proposing new states based on the current state of the chain and acceptance probabilities. Gaussian Mixture Models (GMM) assumes the data is generated from a mixture of Gaussian distributions and can generate new samples by sampling from the learned distributions. To generate the high-dimensional structured data, representation-based approaches try to reconstruct the data from laten feature space. For instance, Variational Autoencoders (VAEs) [188] learn the distribution of the latent representation space given the observed data and then use a decoder network to reconstruct data based on sampled latent representations. Generative Adversarial Networks (GANs) [192] train a generator network to produce samples that resemble realistic data by optimizing an adversarial objective against a discriminator network.

Compared to the other model reuse mechanisms, generation has three unique advantages. The first advantage is visualization and verification. Unlike the extracted knowledge in distillation, the quality of generated content can be visualized and validated by humans. This capability aids in assessing the contributions made by participants in terms of generating valuable content. Second, the flexibility of generation methods allow us to generate data with any desired amount or class, which enables more effective handling of imbalanced [193] and non-IID [173] data. The third advantage is multi-format sharing. Participants have the freedom to choose the form of their contributions. For example, they can upload the learned generative models in source code (e.g., Stable Diffusion [91], GPT-2 [78]) or binary form, upload synthetic data, or provide model inference APIs like ChatGPT. This sharing policy can greatly empower the model community in open FL platforms, fostering collaboration and knowledge sharing.

TABLE IV
CLASSIFICATION OF **GENERATION** WORKS IN THE FIELD OF FL. SOME OF THE WORKS ARE ALSO LISTED IN TABLE III BECAUSE THEY UTILIZE HYBRID MODEL REUSE MECHANISMS.

	Enriching Training Set	Improving Generalization	Enabling Semi-supervised Learning
For Training	FedSage+ [197], GFL [198], FRD [200]	Fed-ZDA [199], FedDG [203], FOSTER [201], DynaFed [204], SDA-FL [205], FedMix [196], FedCAVE-Ens [171]	FedDISC [202], SemiFL [206]
For KD	DENSE [170], FedBE [142], FedDyn [131], FedZKT [207]	FedGen [172], FedFTG [173], FD+FAug [130], FedCAVE-KD [171]	FedSSL [175]

Given the aforementioned advantages, generation methods have been extensively studied in the field of FL. As summaried in TABLE IV, these works can be classified into two main categories: generation for training [171], [197]–[206], generation for KD [130], [131], [142], [170]–[173], [175], [207], and serving three purposes: enriching the training set [131], [142], [170], [197], [198], [200], [207], improving generalization ability [130], [171]–[173], [199], [201], [203]–[205], and enabling semi-supervised learning [175], [202], [206]. As an example of generation for training, FedSage+ [197] trains a missing neighbors generator to mend the links between cross-subgraph nodes, thereby increasing the connectivity of local data and benefiting from this collaboration across clients. Previously mentioned FedCAVE-KD [171] is an example of generation for KD, where locally trained CVAEs and local label distributions are uploaded to the server for DFKD, ensuring privacy while also enhancing generalization of global model. Another example is FedGen [172], where the generator is maintained by the server and sent to clients in each round. The generator has knowledge about the global view of the data distribution, which is used to KD into the local models, thereby enhancing their generalizability. The last application of generation is in semi-supervised learning, which is a common real-world scenario where the client data is unlabeled. For example, FedDISC [202] leverages the average and cluster centroids of hidden representations across pseudo-labels as input to a pre-trained diffusion model, aiming to generate high-quality samples for training.

In fact, the three purposes of generation correspond to three types of data heterogeneity in FL [177]: Quantity Skew, Label Distribution Skew (non-IID), and Missing Labels, which are challenging to address with traditional model amalgamation methods. The hybrid model reusing strategies, which leverage each other's strengths, have become a common paradigm in recent FL studies [131], [143], [151], [171], [198], [201], [202]. Therefore, in the next section, we will summarize the popular hybrid model reusing studies in FL and then filter the studies that are suitable for query-based FL platforms.

E. Hybrid Model Reusing and Model Licenses

Following the taxonomy we introduced in §V, it can be observed that almost all FL studies can be regarded as a permutation of four model reuse mechanisms: Combination, Amalgamation, Distillation, and Generation. To enhance our understanding of current model reusing studies and identify methods applicable for constructing an open FL platform, we provide a comprehensive summary in TABLE V. We employ different colors and fonts in TABLE V to emphasize the distinctions among studies, while certain processes have been omitted without ambiguity. In addition, we have listed the main goals of each study, and only those with explicit designs, experiments, or proof are counted. Please refer to the table caption for the explanation of our denotations. As an example, the process of RSCFed [159] involves the following steps: ① KD, the knowledge is extracted as the softmax values from the self model based on local private data (ref. TABLE III) and performed at the client-side; ② Model exponential moving averaging performed at the client-side; ③ Simple Model Averaging performed two times at the server-side. These processes are repeated across multiple communication rounds until completion. In this way, we can categorize these works and make intuitive comparisons between them‡IV-A.

Through disassembling hybrid model reusing into the four mechanisms, we can further analyze the corresponding clauses in model licenses perspective for FL studies. Then we can easily identify the applicable clauses for different reusing mechanisms‡IV-B and analyze potential licenses conflicts[3].

VI. REMAINING TOPICS IN QUERY-BASED FL

A. Model Protection

In the previous sections, we have extensively discussed the construction of a query-based FL platform from both technical and legal perspectives. However, model management and protection continues to be a significant concern, raising questions such as *How can I protect my models from plagiarism after they are released?* Our conclusion is using **dynamic fingerprinting strategies with blackbox verification support**‡V, such as DeepJudge [211] and Zest [212].

B. Limitations of Query-based FL

In query-based FL systems, the processes of model production and reuse are decoupled to maximize the autonomy of each participant. However, this losse cooperation paradigm is no longer compatiable with online collaboration ML frameworks, which means that it cannot fully harness participants' communicational and computational resource to enhace the training performance. Therefore, even though the server-dominated cooperation frameworks like FedAvg have limitations, as described in §III-B, it is still worthwhile to provide support for its underlying distributed ML training methods to enhance the flexibility and compatibility of our open FL platforms. In the next section, we illustrate another cooperation framework named contract-based FL, which follows a mutual choice design philosophy similar to crowdsourcing platforms [213]. It can serve as an extension of traditional FL and query-based FL.

VII. CONTRACT-BASED FEDERATED LEARNING

A. Overiew

Let us consider another open FL platform based on a contract-based cooperation framework, where cooperation can be built by publishing ML tasks and accepting ML collaboration requests. An overview of this platform is presented in

[3]We developed ModelGo to implement this idea.

TABLE V

COMPARATIVE ANALYSIS OF FL STUDIES CATEGORIZED BY OUR TAXONOMY FOR BATCH MODEL REUSE MECHANISMS. STUDIES APPLICABLE AND CONDITIONAL APPLICABLE TO QUERY-BASED FL ARE MARKED WITH DIFFERENT COLORS; PURPLE DENOTES OPERATIONS COMPLETED ON SERVER, OR KNOWLEDGE DISTILLED FROM GLOBAL OR CONSENSUS MODEL; BLUE DENOTES OPERATIONS COMPLETED ON CLIENTS, OR KNOWLEDGE DISTILLED FROM LOCAL, PERSONLIZED, OR GENERATIVE MODELS; **KNOWLEDGE** OR **GENERATED CONTENT** BASED ON **PUBLIC, PROXY OR GENERATED** DATA, AND *Knowledge* OR *Generated Content* BASED ON *Local, Private or Sensitive* DATA; [ ]*1: ONE ROUND OF COMMUNICATION (AKA ONE-SHOT), [ ]*N: MULTIPLE ROUNDS OF COMMUNICATION, PROCESSES AHEAD ...[ ] ARE PERFORMED ONLY ONCE (I.E. PREPROCESSING), PROCESSES INSIDE [...] ARE MAIN FUNCTIONAL PART; SLASH "/": MODEL TRAINING BASED ON *private* DATA, COMMA ",": MODEL TRAINING BASED ON **NON-SENSITIVE** DATA; GOALS OF WORKS: **E**FFICIENCY **H**ETEROGENEITY **P**RIVACY.

FL Studies	Combination	Amalgamation	Distillation	Generation	Process	Goals
FedAvg [8]	n/a	Model Avg	n/a	n/a	[/A]*N	EH
FedAD [162]	n/a	n/a	KD **Attention, Logits**	n/a	/[D]*1	HP
FedKD$_1$ [166]	n/a	n/a	KD **Weighted Logits**	n/a	/[D]*1	EHP
FedED [139]	n/a	n/a	KD **Logits Avg on Global Validation Data**	n/a	[,D]*N	EHP
FedIris [180]	n/a	n/a	KD *Hidden*	n/a	[D]*N	H
FedProto [155]	n/a	n/a	KD *Per-Class Hidden Avg*	n/a	[/D]*N	EH
FedMD [161]	n/a	n/a	KD **Logits Avg**	n/a	[D/]*N	H
FedMD-NFDP [163]	n/a	n/a	KD **Logits/Softmax/Argmax Avg**	n/a	/[D/]*N	HP
DS-FL [169]	n/a	n/a	KD **Entropy Reduced Logits Avg**	n/a	[/D]*N	EH
RHFL [165]	n/a	n/a	KD **Weighted Logits**	n/a	/[D]*N	EH
KT-pFL [168]	n/a	n/a	KD **Learned Weighted Softmax**	n/a	[/D]*N	EH
Cronus [167]	n/a	n/a	KD **Robust Mean Estimation of Softmax**	n/a	/[/D]*N	EHP
FedDISC [202]	n/a	n/a	n/a	*Synthetic Data*	[G]*1,	EH
FRD [200]	n/a	n/a	n/a	*Mixup Data*	[G]*1,	E
One-Shot FL [141]	Output Avg	n/a	KD **Softmax**	n/a	/[CD]*1	EP
FedDF [140]	n/a	Model Avg	KD **Logits Avg**	n/a	[/AD]*N	HP
PerAda [143]	n/a	Adapter Avg	KD **Logits Avg**	n/a	[//AD]*N	EHP
FedFiMa [208]	n/a	Rep. Layer Avg	KD **Hidden Avg**	n/a	[/AD]*N	EH
FedFusion [145]	n/a	Model Avg	KD *Hidden*, *Hidden*	n/a	[DA]*N	E
FedNTD [147]	n/a	Model Avg	KD *Not-True Classes Softmax*	n/a	[DA]*N	H
FedKC [174]	n/a	Model Avg	KD *Clustered Hidden Avg*	n/a	[DA]*N	HP
FedDistill [156]	n/a	Model Avg	KD *Softmax of Latest Local Model*	n/a	[/DA]*N	H
pFedSD [158]	n/a	Model Avg	KD *Softmax of Pervious Local Model*	n/a	[/DA]*N	H
FedMLB [148]	n/a	Model Avg	KD *Softmax, Scaled Softmax*	n/a	[/DA]*N	EH
FedAlign$_1$ [150]	n/a	Model Avg	KD *Lipschitz Constants* [209]	n/a	[/DA]*N	EH
MOON [12]	n/a	Model Avg	Contrastive Learning *Hidden*, *Hidden*	n/a	[/DA]*N	EH
FedCAD [149]	n/a	Model Avg	KD *Class-Wise Softmax*	n/a	[/DA]*N	H
FedGKT [210]	n/a	n/a	KD *Logits* KD *Logits, Hidden, Argmax*	n/a	[/D/D]*N	EH
FCCL [157]	n/a	n/a	Contrastive Learning **Logits Avg** Continual Learning *Logits*	n/a	[D/D]*N	H
GFL [198]	n/a	Model Avg	n/a	**Synthetic Data**	/G[,A,]*N	HP
FOSTER [201]	n/a	Model Avg	n/a	**Synthetic Outliers**	,[G/A]*N	EH
FedDG [203]	n/a	Model Avg	n/a	**Interpolated Data**	[G/A]*N	H
SemiFL [206]	n/a	Model Avg	n/a	*Augmented and Mixup Data*	[,G/A]*N	H
NeighGen [197] FedSage [197]		Gradient Avg Model Avg	n/a	*Synthetic Node*	[G/A]*N [/A]*N	H
Fed-ZDAC [199] Fed-ZDAS [199]	n/a n/a	Model Avg n/a	n/a n/a	**Zero-shot Synthetic Data** **Zero-shot Synthetic Data**	[G/A]*N [/G,A]*N	HP
FedMix [196]	n/a	Model Avg	n/a	*Mixup Data Mixup Data*	[GA,]*NG,	HP
FedDyn [131]	n/a	n/a	KD *Hidden, Logits Avg*	**Synthetic Data**	[/GD]*N	E
FD+FAug [130]	n/a	n/a	KD *Per-Class Logits Avg*	**Synthetic Data**	[/G/D]*N	EP
FedCAVE-Ens [171] FedCAVE-KD [171]	Collection n/a	n/a n/a	n/a KD **Softmax**	**Synthetic Data**	/C[G,]*1 /C[GDG,]*1	H H
Fed-ET [144]	n/a	Rep. Layer Avg Model Avg	KD **Argmax, Weighted Logits**	n/a	[/ADA]*N	EH
FedBE [142]	n/a	Model Avg	KD **Softmax Avg**	**Synthetic Model**	[/AGD]*1	H
DynaFed [204]	n/a	Model Avg	n/a	**Synthetic Data**	[/A]*NG[A,]*N	EH
FedAUX [164]	n/a	Model Avg	Contrastive Learning **Hidden**, *Hidden* KD **Weighted Logits**	n/a	,D[A/D]*N ,D[A/D]*1	HP
FedKD$_2$ [146]	n/a	Gradient Avg	KD *Hidden, Attention, Logits* KD *Hidden, Attention, Logits*	n/a	[/DAD]*N	EH
FedGen [172]	n/a	Model Avg	KD **Softmax** KD **Logits Avg**	**Augmented Data**	D[/G,AD]*N	EHP
SDA-FL [205]	n/a	Model Avg	KD **Argmax**	**Synthetic Data** *Augmented Data*	/G[G/AD]*N	HP
FedKT [160]	Voting Voting	n/a	KD **Argmax** KD **Argmax**	n/a	[/CDCD]*1	HP
FedMatch [152]	Voting	Model Avg	KD *Argmax*	n/a	[C/DAA]*N [,CDA]*N	EH
FedFTG [173]	n/a	Model Avg	KD **Softmax**, **Softmax** KD **Softmax**	**Synthetic Data**	[/AGDD]*N	EH
RSCFed [159] (Unlabeled Case)	n/a	Model EMA Model Avg	KD *Softmax*	n/a	[DAAA]*N	H
FedAlign$_2$ [151]	n/a	Model Avg	Contrastive Learning *Argmax, Argmin*	n/a	[DDAA]*N	EH
FedSSL [175]	n/a	n/a	KD **Softmax** KD **Softmax**, *Interpolated Softmax*	**Synthetic Data**	[/GD/GD]*N	HP
DENSE [170]	Collection	n/a	KD **Logits Avg**, *Batch-Wise Statistics*, **Softmax** KD **Softmax**, **Softmax**	**Synthetic Data**	/C[G,DGD]*1	HP
FedZKT [207]	n/a	n/a	KD **Softmax**, **Softmax Avg** KD **Softmax Avg** KD **Softmax**	**Synthetic Data**	[/GDGDGD]*N	H

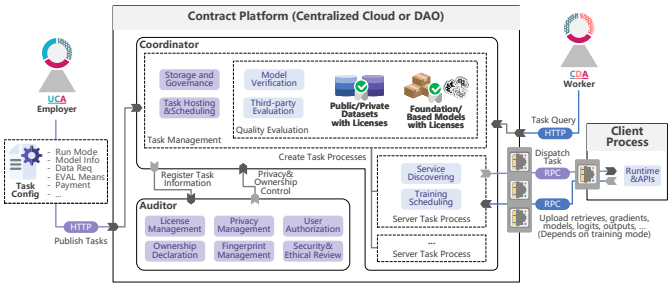


Fig. 6. An overview of contract-based FL systems. (U: model User, C: Coordinator, D: Data owner, A: Auditor)

Fig. 6, the major difference between traditional FL (ref Fig. 3) and contract-based FL is that we involve a trustworthy third-party platform called Contract Platform to host and coordinate ML tasks for platform users. Rather than directly pushing ML tasks from servers to clients, employers need to publish the tasks to the contract platform and wait for acceptance from workers, which involves a mutual choice procedure. The conditions and payment for participation are also appliable, and the final model will be audited and evaluated by the platform for fairness. In addition, privacy-enhanced technologies [42] and fingerprint management [211] can also be implemented by the platform to protect the IP of platform users. The properties of contract-based FL are: 1) **Opt-in**, as workers reserve the right to join or quit from training networks; 2) **Contractual**, enabling employers to define payment, organization mode, model quality criteria, rehire rules, etc. through contracts; 3) **Market-based**, where contracts are open and task pricing is influenced and determined by the market.

In short, contract-based FL retains all functionalities related to model training at the platform and only provides task publishing services to employers, who play the role of model training in traditional FL. Furthermore, similar to model-reusing mechanisms (ref. §V) that can be applied to query-based FL, our contract-based FL is not limited to an amalgamation-based collaborative training paradigm like FedAvg. In fact, the contract platform is designed as a crowdsourcing platform among data miners for ML cooperation tasks, including distributed training, fine-tuning, and ensembling based on scratch or Foundation Models (FMs) [214]. However, most current crowdsourcing platforms like Amazon Mechanical Turk and Appen[4] primarily deal with human intelligence tasks such as data collection and annotation (aka. microtasks) rather than ML modeling tasks. It remains unexplored how to design and publish an ML task to seek crowd labor. On the other hand, many studies [215]–[219] and products [58], [220], [221] emphasize monetizing ML activities through blockchain-based techniques and Decentralized Autonomous Organizations (DAOs). However, both monetization and decentralization are non-essential functions for our contract platforms[‡VI].

B. How to Design ML Microtasks

When we come to the scenario of contract-based FL, we first need to define how to design ML microtasks for crowd workers. Unfortunately, previous collaborative ML studies [25], [33] rarely discuss this question because they default to assuming that all employers or workers have the same resource type to fit into their modeling frameworks. On the contrary, in contract-based FL, we leave the freedom of design microtasks and choosing modeling method to the employers.

The answer depends on what resources the employers have and what tasks the workers can perform. For example, in horizontal FL, the server should provide an initial model, and clients should have training data under the same feature space and provide computational power. However, in vertical FL, the clients should have training data under the same sample space (i.e., ID space). This difference determines which modeling framework employers can adopt. Following this idea, we provide comparisons of typical FL [8], distributed ML [224], and blockchain-based [221] modeling frameworks in TABLE VI. For simplicity, we introduce the **Platform** between Employer and Worker in some frameworks to make it adaptable for contract-based FL. You can quickly restore the original structure by merging Employer and Platform columns. To represent the transmission of resources, we mark resources provided by each group of entities with different colors. The computational resource used for training is marked with ★. By this way, we can conveniently find the feasible and privacy-preserving modeling frameworks for collaborative learning. Here, we briefly introduce these frameworks group by scenarios[5].

1) Train from scratch with workers' private data: There are four frameworks that support this scenario. The first is **Centralized ML**, which requires workers to upload their data to a cloud platform for modeling. However, this approach is not feasible for FL scenarios due to privacy concerns. The second is **FedAvg** [8], which requires workers to upload computed gradients upon the scratch model and their private data (horizontally split). Computing resources need to be provided by workers as well. The third is **SplitNN** [183], where each worker trains the cut layers of the model using their vertically split data. The hidden representations from these layers are uploaded to the platform for the training of the remaining layers. The last and distinctive is a Web3 system named **Ocean** [221]. It enables workers to upload only the metadata of their local dataset to a blockchain-based platform to attract buyers. Interested buyers then need to purchase datatokens to access the local dataset and deploy their modeling algorithm. To ensure the privacy and IP security of workers, only trusted algorithms are allowed, and only model predictions will be sent to buyers.

2) Train from FMs: FMs, trained on larger-scale data with robust generalization abilities across various downstream tasks, serve as a solid foundation for collaborative training. Yuan *et al.* proposed **DT-FM** [214] to establish an effective geo-distributed learning system across 8 regions for training the language model GPT-3. Similarly, Borzunov *et al.* [223] recruited volunteer nodes to train a transformer model over the Internet. To enhance system robustness, **Moshpit SGD** [224] divides worker nodes into small, independent groups to ensure that all-reduce is not affected by the failure of a single

[4]AMT: https://www.mturk.com/; Appen: https://appen.com/crowd-2/

[5]From the viewpoint of Employers.

TABLE VI
COMPARISONS OF TYPICAL MODEL TRAINING FRAMEWORKS WITH THE INTERVENTION OF A TRUSTWORTHY PLATFORM. *, *, * INDICATE RESOURCES FROM EMPLOYER, PLATFORM, AND WORKER, RESPECTIVELY. RESOURCES IN DIFFERENT COLOR COLUMNS (E.G., DATA IN COLUMN PLATFORM) REPRESENT TRANSMISSION. ★: LOCATION OF MODEL TRAINING. *Resources* RECEIVED FROM OTHER WORKERS.

	Employer	Platform	Worker
Centralized ML	Model	Model Data ★	Data
FedAvg [8]	Model	Model Gradient	Model Data ★
DMoE [222]	Gating NN Data	Gating NN Data Output ★	Model Data ★
DeDES [190]	Model Subset	Model	Model Data ★
Borzunov *et al.* [223]	Model	Model Data Gradient	Model Data-Stream ★
Moshpit SGD [224]	Model Data	Model Data Gradient	Model Data-Loader ★
VC-ASGD [225]	Model Data	Model Data Gradient	Sub-model H-Data ★
SplitNN [183]	Model ID-Label	Model ID-Label Hidden ★	Sub-model V-Data Gradient ★
DT-FM [214]	FM	FM Data	Sub-FM Micro-Batch *Gradient* ★
FS-LLM [226]	FM	FM Adapter	FM Data ★
FedKSeed [227]	FM	FM Seed Scaler Gradient	FM Seed Data ★
Berdoz *et al.* [228]	Class	Class NN Structure Hidden	NN Structure Data Hidden AVG ★
CCL [153]	Class	Class	Data *Model* *Hidden AVG* ★
FedProto [154], [155]	Class	Class Hidden	Data Model Hidden AVG ★
Ocean [221]	Model Output	Metadata	Metadata Model Data ★

participant. Additionally, **VC-ASGD** [225] divides the deep learning training job into asynchronous parameter update subtasks to improve scalability. In these cases, employers are required to supply the initial model and training data needed to launch training subtasks. This training paradigm is particularly suitable for worker nodes that possess computational and communication resources but lack their own training data.

Instead of initiating training from scratch and consuming substantial computational resources, we can also collaboratively adapt these FMs to specific local tasks based on relatively restricted hardware and data size through fine-tuning. Recently, Woisetschläger *et al.* [229] have explored the opportunity to fine-tune large language models, such as FLAN-T5, by edge devices. Simultaneously, **FS-LLM** [226] employs Parameter-Efficient Fine-Tuning (PEFT) methods for federated fine-tuning of LLaMA-7B, minimizing communication and computation costs. **FATE-LLM** [230] offers a solution called FedHeteroLLM, leveraging KD to train a mentee model from its local pre-trained LLM for federated aggregation. To alleviate the computational burden associated with backpropagation-based optimization methods, **FedKSeed** [227] employs zeroth-order optimization (ZOO). Only seed and scalar gradients need to be transmitted for federated fine-tuning.

3) Ensemble workers' private model: As we presented in §V, ensembling is an effective method to integrate knowledge, and the distributed version is proposed by **DMoE** [222]. It employs a Distributed Hash Table (DHT) to store metadata and worker statuses, constructing a decentralized expert network. To make inferences using this network, each model user must train a local gating network to select a subset of experts tailored to their input. No additional training required, Wang *et al.* purposed **DeDES** [190], which selects a diverse subsect of weak models from population and make inference by voting. The unique advantage of ensembling lies in its inherent support for model heterogeneity, and its integration is extensible. This capability enables the establishment of a cooperative network while maintaining flexibility and availability.

4) Aggregated workers's private representations: Another popular scheme for organizing ML microtasks involves sharing the learned representations of workers. For instance, the use of class-conditional average of last hidden layer activations (aka. Prototypes) can enhance class discrimination across different clients (e.g., **FedProto** [154], [155] and [228]). With no central server required, **CCL** [153] presents a decentralized learning approach based on cross-workers prototypes sharing. To configure the microtasks, employers only need to set the target class (no scratch model reqiured), and the platform can then collect and distribute per-class prototypes among workers.

C. Limitations of Contract-based FL

In contract-based FL systems, employers can publish their personalized ML crowdsourcing tasks, considering both their local resources and the resources of the target clients. However, to meet the flexibility requirements of widely supporting training methods, the contract platform should offer highly compatible APIs that can integrate various distributed training frameworks. Additionally, it needs to provide comprehensive audit against malicious users and plagiarism, significantly increasing the complexity compared to traditional FL and query-based FL. Meanwhile, to establish confidence among users, a fair and consensual third-party model evaluation mechanism should be established, which, however, is almost unexplored in the current academic studies.

VIII. CONCLUSION

Traditional federated learning systems with a server-dominated cooperation framework suppress the enthusiasm of participants and limit the further extension of such collaboration. To explore the opportunity to establish a more open and reciprocal cooperation platform, we investigate current progress in federated learning, decentralized machine learning, and model reusing systems. In this way, we depict two rough sketches of open federated learning platforms: query-based federated learning paltform and contract-based federated

learning paltform. Based on these two proposed platforms, we survey their possible supported techniques and their related legal issues, including ML licensing and copyrightability. We believe this survey can encourage a rethinking of current collaborative ML systems design and lead to the pervasive availability of AI for everyone.

ACKNOWLEDGMENTS

This research is supported by the National Research Foundation Singapore and DSO National Laboratories under the AI Singapore Programme (AISG Award No: AISG2-RP-2020-018). Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not reflect the views of National Research Foundation, Singapore.

REFERENCES

[1] S. E. Whang, Y. Roh, H. Song, and J.-G. Lee, "Data collection and quality challenges in deep learning: A data-centric ai perspective," *The VLDB Journal*, pp. 1–23, 2023.

[2] O. Tene, "Privacy: The new generations," *International data privacy law*, vol. 1, no. 1, pp. 15–27, 2011.

[3] P. Voigt and A. Von dem Bussche, "The EU general data protection regulation (GDPR): A practical guide," *Springer International Publishing*, 2017.

[4] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine (SPM)*, vol. 37, no. 3, pp. 50–60, 2020.

[5] N. Truong, K. Sun, S. Wang, F. Guitton, and Y. Guo, "Privacy preservation in federated learning: An insightful survey from the gdpr perspective," *Computers & Security*, vol. 110, p. 102402, 2021.

[6] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konecny, S. Mazzocchi, H. B. McMahan *et al.*, "Towards federated learning at scale: System design," in *Proceedings of the 2nd SysML Conference*, 2019.

[7] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," in *NIPS 2016 workshop on Private Multi-Party Machine Learning*, 2016.

[8] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017, pp. 1273–1282.

[9] C. Xu, Y. Qu, Y. Xiang, and L. Gao, "Asynchronous federated learning on heterogeneous devices: A survey," *Computer Science Review*, vol. 50, p. 100595, 2023.

[10] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proceedings of the 3rd SysML Conference*, 2020.

[11] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "SCAFFOLD: Stochastic controlled averaging for federated learning," in *Proceedings of the 37th International Conference on Machine Learning (ICML)*. PMLR, 2020, pp. 5132–5143.

[12] Q. Li, B. He, and D. Song, "Model-contrastive federated learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 10 713–10 722.

[13] M. Duan, D. Liu, X. Chen, R. Liu, Y. Tan, and L. Liang, "Self-balancing federated learning with global imbalanced data in mobile systems," *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, vol. 32, no. 1, pp. 59–71, 2020.

[14] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Robust and communication-efficient federated learning from non-iid data," *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, vol. 31, no. 9, pp. 3400–3413, 2019.

[15] L. Li, D. Liu, M. Duan, Y. Zhang, A. Ren, X. Chen, Y. Tan, and C. Wang, "Federated learning with workload-aware client scheduling in heterogeneous systems," *Neural Networks*, vol. 154, pp. 560–573, 2022.

[16] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 2017, pp. 1175–1191.

[17] R. C. Geyer, T. Klein, and M. Nabi, "Differentially private federated learning: A client level perspective," *arXiv preprint arXiv:1712.07557*, 2017.

[18] K. Cheng, T. Fan, Y. Jin, Y. Liu, T. Chen, D. Papadopoulos, and Q. Yang, "SecureBoost: A lossless federated learning framework," *IEEE Intelligent Systems*, vol. 36, no. 6, pp. 87–98, 2021.

[19] I. C. Society, "Ieee guide for architectural framework and application of federated machine learning," *IEEE Std 3652.1-2020*, pp. 1–69, 2021.

[20] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.

[21] Q. Yang, L. Fan, R. Tong, and A. Lv, "Ieee federated machine learning," *IEEE Federated Machine Learning - White Paper*, pp. 1–18, 2021.

[22] M. Aledhari, R. Razzak, R. M. Parizi, and F. Saeed, "Federated learning: A survey on enabling technologies, protocols, and applications," *IEEE Access*, vol. 8, pp. 140 699–140 725, 2020.

[23] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[24] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, and Y. Gao, "A survey on federated learning," *Knowledge-Based Systems (KBS)*, vol. 216, p. 106775, 2021.

[25] Q. Li, Z. Wen, Z. Wu, S. Hu, N. Wang, Y. Li, X. Liu, and B. He, "A survey on federated learning systems: vision, hype and reality for data privacy and protection," *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 2021.

[26] V. Kulkarni, M. Kulkarni, and A. Pant, "Survey of personalization techniques for federated learning," in *2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*. IEEE, 2020, pp. 794–797.

[27] A. Z. Tan, H. Yu, L. Cui, and Q. Yang, "Towards personalized federated learning," *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, pp. 1–17, 2022.

[28] X. Ma, J. Zhu, Z. Lin, S. Chen, and Y. Qin, "A state-of-the-art survey on solving non-iid data in federated learning," *Future Generation Computer Systems (FGCS)*, vol. 135, pp. 244–258, 2022.

[29] Z. Liu, J. Guo, W. Yang, J. Fan, K.-Y. Lam, and J. Zhao, "Privacy-preserving aggregation in federated learning: A survey," *IEEE Transactions on Big Data (TBD)*, pp. 1–20, 2022.

[30] A. El Ouadrhiri and A. Abdelhadi, "Differential privacy for deep and federated learning: A survey," *IEEE Access*, vol. 10, pp. 22 359–22 380, 2022.

[31] X. Yin, Y. Zhu, and J. Hu, "A comprehensive survey of privacy-preserving federated learning: A taxonomy, review, and future directions," *ACM Computing Surveys (CSUR)*, vol. 54, no. 6, pp. 1–36, 2021.

[32] L. Lyu, H. Yu, and Q. Yang, "Threats to federated learning: A survey," *arXiv preprint arXiv:2003.02133*, 2020.

[33] D. C. Nguyen, M. Ding, Q.-V. Pham, P. N. Pathirana, L. B. Le, A. Seneviratne, J. Li, D. Niyato, and H. V. Poor, "Federated learning meets blockchain in edge computing: Opportunities and challenges," *IEEE Internet of Things Journal (IoT-J)*, vol. 8, no. 16, pp. 12 806–12 825, 2021.

[34] Y. Qu, M. P. Uddin, C. Gan, Y. Xiang, L. Gao, and J. Yearwood, "Blockchain-enabled federated learning: A survey," *ACM Computing Surveys (CSUR)*, vol. 55, no. 4, pp. 1–35, 2022.

[35] J. Zhu, J. Cao, D. Saxena, S. Jiang, and H. Ferradi, "Blockchain-empowered federated learning: Challenges, solutions, and future directions," *ACM Computing Surveys (CSUR)*, 2022.

[36] C. Niu, F. Wu, S. Tang, L. Hua, R. Jia, C. Lv, Z. Wu, and G. Chen, "Billion-scale federated learning on mobile clients: A submodel design with tunable privacy," in *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2020, pp. 1–14.

[37] S. AbdulRahman, H. Tout, A. Mourad, and C. Talhi, "Fedmccs: multicriteria client selection model for optimal iot federated learning," *IEEE Internet of Things Journal (IoT-J)*, vol. 8, no. 6, pp. 4723–4735, 2020.

[38] B. G. Tekgul, Y. Xia, S. Marchal, and N. Asokan, "WAFFLE: Watermarking in federated learning," in *Proceedings of the 40th International*

Symposium on Reliable Distributed Systems (SRDS). IEEE, 2021, pp. 310–320.

[39] S. Shao, W. Yang, H. Gu, J. Lou, Z. Qin, L. Fan, Q. Yang, and K. Ren, "Fedtracker: Furnishing ownership verification and traceability for federated learning model," *arXiv preprint arXiv:2211.07160*, 2022.

[40] H. Yu, Z. Liu, Y. Liu, T. Chen, M. Cong, X. Weng, D. Niyato, and Q. Yang, "A fairness-aware incentive scheme for federated learning," in *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society (AIES)*, 2020, pp. 393–399.

[41] L. Hanzlik, Y. Zhang, K. Grosse, A. Salem, M. Augustin, M. Backes, and M. Fritz, "MLCapsule: Guarded offline deployment of machine learning as a service," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, 2021, pp. 3300–3309.

[42] E. Hesamifard, H. Takabi, M. Ghasemi, and R. N. Wright, "Privacy-preserving machine learning as a service," *Proceedings on Privacy Enhancing Technologies*, vol. 2018, no. 3, pp. 123–142, 2018.

[43] F. Sattler, K.-R. Müller, T. Wiegand, and W. Samek, "On the byzantine robustness of clustered federated learning," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 8861–8865.

[44] A. Act, "Health insurance portability and accountability act of 1996," *Public law*, vol. 104, p. 191, 1996.

[45] C. Dwork, "Differential privacy," in *Automata, Languages and Programming: 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II 33*. Springer, 2006, pp. 1–12.

[46] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang, and H. Qi, "Beyond inferring class representatives: User-level privacy leakage from federated learning," in *Proceedings of the 2019 IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 2019, pp. 2512–2520.

[47] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 32. Curran Associates, Inc., 2019.

[48] X. Jin, P.-Y. Chen, C.-Y. Hsu, C.-M. Yu, and T. Chen, "Cafe: Catastrophic data leakage in vertical federated learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 34, 2021, pp. 994–1006.

[49] W. Wei, L. Liu, Y. Wut, G. Su, and A. Iyengar, "Gradient-leakage resilient federated learning," in *2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2021, pp. 797–807.

[50] Z. Li, J. Zhang, L. Liu, and J. Liu, "Auditing privacy defenses in federated learning via generative gradient leakage," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 10 132–10 142.

[51] Q. Zhang, T. Wu, P. Zhou, S. Zhou, Y. Yang, and X. Jin, "Felicitas: Federated learning in distributed cross device collaborative frameworks," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 4502–4509.

[52] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning," in *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, 2016, pp. 265–283.

[53] Y. Liu, T. Fan, T. Chen, Q. Xu, and Q. Yang, "FATE: An industrial grade platform for collaborative learning with data protection," *The Journal of Machine Learning Research (JMLR)*, vol. 22, no. 1, pp. 10 320–10 325, 2021.

[54] D. J. Beutel, T. Topal, A. Mathur, X. Qiu, T. Parcollet, P. P. de Gusmão, and N. D. Lane, "Flower: A friendly federated learning research framework," *arXiv preprint arXiv:2007.14390*, 2020.

[55] C. He, S. Li, J. So, X. Zeng, M. Zhang, H. Wang, X. Wang, P. Vepakomma, A. Singh, H. Qiu *et al.*, "FedML: A research library and benchmark for federated machine learning," in *NeurIPS 2020 Workshop on Scalability, Privacy, and Security in Federated Learning*, 2020.

[56] P. Foley, M. J. Sheller, B. Edwards, S. Pati, W. Riviera, M. Sharma, P. N. Moorthy, S.-h. Wang, J. Martin, P. Mirhaji *et al.*, "OpenFL: the open federated learning library," *Physics in Medicine & Biology*, vol. 67, no. 21, p. 214001, 2022.

[57] H. R. Roth, Y. Cheng, Y. Wen, I. Yang, Z. Xu, Y.-T. Hsieh, K. Kersten, A. Harouni, C. Zhao, K. Lu *et al.*, "Nvidia flare: Federated learning from simulation to real-world," in *NeurIPS 2022 Workshop on Federated Learning: Recent Advances and New Challenges*, 2022.

[58] A. Ziller, A. Trask, A. Lopardo, B. Szymkow, B. Wagner, E. Bluemke, J.-M. Nounahon, J. Passerat-Palmbach, K. Prakash, N. Rose *et al.*, "PySyft: A library for easy federated learning," *Federated Learning Systems: Towards Next-Generation AI*, pp. 111–139, 2021.

[59] T. Zhang, L. Gao, C. He, M. Zhang, B. Krishnamachari, and A. S. Avestimehr, "Federated learning for the internet of things: applications, challenges, and opportunities," *IEEE Internet of Things Magazine (IoTM)*, vol. 5, no. 1, pp. 24–29, 2022.

[60] M. Alazab, S. P. RM, M. Parimala, P. K. R. Maddikunta, T. R. Gadekallu, and Q.-V. Pham, "Federated learning for cybersecurity: concepts, challenges, and future directions," *IEEE Transactions on Industrial Informatics (TII)*, vol. 18, no. 5, pp. 3501–3509, 2021.

[61] D. Zeng, S. Liang, X. Hu, H. Wang, and Z. Xu, "FedLab: A flexible federated learning framework," *Journal of Machine Learning Research*, vol. 24, no. 100, pp. 1–7, 2023.

[62] S. Caldas, P. Wu, T. Li, J. Konečnỳ, H. B. McMahan, V. Smith, and A. Talwalkar, "LEAF: A benchmark for federated settings," *arXiv preprint arXiv:1812.01097*, 2018.

[63] Y. Chen, B. Zheng, Z. Zhang, Q. Wang, C. Shen, and Q. Zhang, "Deep learning on mobile and embedded devices: State-of-the-art, challenges, and future directions," *ACM Computing Surveys (CSUR)*, vol. 53, no. 4, pp. 1–37, 2020.

[64] L. Li, D. Li, T. F. Bissyandé, J. Klein, Y. Le Traon, D. Lo, and L. Cavallaro, "Understanding android app piggybacking: A systematic study of malicious code grafting," *IEEE Transactions on Information Forensics and Security (TIFS)*, vol. 12, no. 6, pp. 1269–1284, 2017.

[65] N. Bouacida and P. Mohapatra, "Vulnerabilities in federated learning," *IEEE Access*, vol. 9, pp. 63 229–63 249, 2021.

[66] J. Park, D.-J. Han, M. Choi, and J. Moon, "Sageflow: Robust federated learning against both stragglers and adversaries," in *Advances in neural information processing systems (NeurIPS)*, vol. 34, 2021, pp. 840–851.

[67] M. Fang, X. Cao, J. Jia, and N. Z. Gong, "Local model poisoning attacks to byzantine-robust federated learning," in *Proceedings of the 29th USENIX Conference on Security Symposium*, 2020, pp. 1623–1640.

[68] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *International Conference on Artificial Intelligence and Statistics (AISTATS)*. PMLR, 2020, pp. 2938–2948.

[69] H. Wang, K. Sreenivasan, S. Rajput, H. Vishwakarma, S. Agarwal, J.-y. Sohn, K. Lee, and D. Papailiopoulos, "Attack of the tails: Yes, you really can backdoor federated learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020, pp. 16 070–16 084.

[70] A. Reisizadeh, H. Taheri, A. Mokhtari, H. Hassani, and R. Pedarsani, "Robust and communication-efficient collaborative learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 32, 2019.

[71] H. Ludwig, N. Baracaldo, G. Thomas, Y. Zhou, A. Anwar, S. Rajamoni, Y. Ong, J. Radhakrishnan, A. Verma, M. Sinn *et al.*, "Ibm federated learning: an enterprise framework white paper v0. 1," *arXiv preprint arXiv:2007.10987*, 2020.

[72] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 17th Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2019, pp. 4171–4186.

[73] T. L. Scao, A. Fan, C. Akiki, E. Pavlick, S. Ilić, D. Hesslow, R. Castagné, A. S. Luccioni, F. Yvon, M. Gallé *et al.*, "BLOOM: A 176b-parameter open-access multilingual language model," *arXiv preprint arXiv:2211.05100*, 2022.

[74] S. You, C. Xu, F. Wang, and C. Zhang, "Workshop on model mining," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 4177–4178.

[75] P. Brereton, D. Budgen, K. Bennnett, M. Munro, P. Layzell, L. MaCaulay, D. Griffiths, and C. Stannett, "The future of software," *Communications of the ACM*, vol. 42, no. 12, pp. 78–84, 1999.

[76] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," *Neural computation*, vol. 3, no. 1, pp. 79–87, 1991.

[77] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *NIPS Deep Learning and Representation Learning Workshop*, 2014.

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

[79] Y. LeCun, C. Cortes, and C. Burges, "Mnist handwritten digit database," *ATT Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, vol. 2, 2010.

[80] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering (TKDE)*, vol. 22, no. 10, pp. 1345–1359, 2009.

[81] Z.-H. Zhou, *Ensemble methods: foundations and algorithms*. CRC press, 2012.

[82] M. Wang and W. Deng, "Deep visual domain adaptation: A survey," *Neurocomputing*, vol. 312, pp. 135–153, 2018.

[83] L. Wang and K.-J. Yoon, "Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 44, no. 6, pp. 3048–3068, 2021.

[84] H. Cao, C. Tan, Z. Gao, G. Chen, P.-A. Heng, and S. Z. Li, "A survey on generative diffusion model," *arXiv preprint arXiv:2209.02646*, 2022.

[85] S. Ji, T. Saravirta, S. Pan, G. Long, and A. Walid, "Emerging trends in federated learning: From model fusion to federated x learning," *arXiv preprint arXiv:2102.12920*, 2021.

[86] Y. Mirsky and W. Lee, "The creation and detection of deepfakes: A survey," *ACM Computing Surveys (CSUR)*, vol. 54, no. 1, pp. 1–41, 2021.

[87] D. Licari and G. Comandè, "ITALIAN-LEGAL-BERT: A Pre-trained Transformer Language Model for Italian Law," in *Companion Proceedings of the 23rd International Conference on Knowledge Engineering and Knowledge Management*, ser. CEUR Workshop Proceedings, vol. 3256. Bozen-Bolzano, Italy: CEUR, Sep. 2022.

[88] F. Tang, L. Zeng, F. Wang, and J. Zhou, "Persona authentication through generative dialogue," *arXiv preprint arXiv:2110.12949*, 2021.

[89] E. Nijkamp, B. Pang, H. Hayashi, L. Tu, H. Wang, Y. Zhou, S. Savarese, and C. Xiong, "CodeGen: An open large language model for code with multi-turn program synthesis," in *Proceedings of the 11th International Conference on Learning Representations (ICLR)*, 2023.

[90] L. Zhang, A. Rao, and M. Agrawala, "Adding conditional control to text-to-image diffusion models," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 3836–3847.

[91] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 10 684–10 695.

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

[93] R. Li, L. B. Allal, Y. Zi, N. Muennighoff, D. Kocetkov, C. Mou, M. Marone, C. Akiki, J. Li, J. Chim *et al.*, "StarCoder: may the source be with you!" *arXiv preprint arXiv:2305.06161*, 2023.

[94] S. Zhang, S. Roller, N. Goyal, M. Artetxe, M. Chen, S. Chen, C. Dewan, M. Diab, X. Li, X. V. Lin *et al.*, "OPT: Open pre-trained transformer language models," *arXiv preprint arXiv:2205.01068*, 2022.

[95] P. Goyal, Q. Duval, I. Seessel, M. Caron, M. Singh, I. Misra, L. Sagun, A. Joulin, and P. Bojanowski, "Vision models are more robust and fair when pretrained on uncurated images without supervision," *arXiv preprint arXiv:2202.08360*, 2022.

[96] R. Taylor, M. Kardas, G. Cucurull, T. Scialom, A. Hartshorn, E. Saravia, A. Poulton, V. Kerkez, and R. Stojnic, "GALACTICA: A large language model for science," *arXiv preprint arXiv:2211.09085*, 2022.

[97] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuAD: 100,000+ questions for machine comprehension of text," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2016, pp. 2383–2392.

[98] Y. Huang, T. Lv, L. Cui, Y. Lu, and F. Wei, "LayoutLMv3: Pre-training for document ai with unified text and image masking," in *Proceedings of the 30th ACM International Conference on Multimedia (MM)*, 2022, pp. 4083–4091.

[99] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, and I. Androutsopoulos, "LEGAL-BERT: The muppets straight out of law school," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020, pp. 2898–2904.

[100] Y. Peng, S. Yan, and Z. Lu, "Transfer learning in biomedical natural language processing: An evaluation of bert and elmo on ten benchmarking datasets," in *Proceedings of the 18th BioNLP Workshop and Shared Task*, 2019, pp. 58–65.

[101] L. Rosen, *Open Source Licensing: Software Freedom and Intellectual Property Law*. New Jersey: Prentice Hall Professional Technical Reference, 2005.

[102] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.

[103] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[104] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[105] A. Jobin, M. Ienca, and E. Vayena, "The global landscape of ai ethics guidelines," *Nature Machine Intelligence*, vol. 1, no. 9, pp. 389–399, 2019.

[106] E. Awad, S. Dsouza, R. Kim, J. Schulz, J. Henrich, A. Shariff, J.-F. Bonnefon, and I. Rahwan, "The moral machine experiment," *Nature*, vol. 563, no. 7729, pp. 59–64, 2018.

[107] R. Yuste, S. Goering, B. A. y. Arcas, G. Bi, J. M. Carmena, A. Carter, J. J. Fins, P. Friesen, J. Gallant, J. E. Huggins *et al.*, "Four ethical priorities for neurotechnologies and ai," *Nature*, vol. 551, no. 7679, pp. 159–163, 2017.

[108] D. Contractor, D. McDuff, J. K. Haines, J. Lee, C. Hines, B. Hecht, N. Vincent, and H. Li, "Behavioral use licensing for responsible ai," in *2022 ACM Conference on Fairness, Accountability, and Transparency (FAccT)*, 2022, pp. 778–788.

[109] L. Chen, P. Koutris, and A. Kumar, "Towards model-based pricing for machine learning in a data marketplace," in *Proceedings of the 2019 International Conference on Management of Data (COMAD)*, 2019, pp. 1535–1552.

[110] R. W. Gomulkiewicz, "Open source license proliferation: Helpful diversity or hopeless confusion?" *Washington University Journal of Law & Policy*, vol. 30, no. 1, 2009.

[111] H. R. Reddy, "Jacobsen v. katzer: The federal circuit weighs in on the enforceability of free and open source software licenses ii. copyright - note," *Berkeley Technology Law Journal*, vol. 24, no. 1, pp. 299–320, 2009.

[112] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, no. 2, pp. 241–259, 1992.

[113] M. P. Perrone and L. N. Cooper, "When networks disagree: Ensemble methods for hybrid neural networks," in *How We Learn; How We Remember: Toward An Understanding Of Brain And Neural Systems: Selected Papers of Leon N Cooper*. World Scientific, 1995, pp. 342–358.

[114] B. Clarke, "Comparing bayes model averaging and stacking when model approximation error cannot be ignored," *Journal of Machine Learning Research (JMLR)*, vol. 4, no. Oct, pp. 683–712, 2003.

[115] Z. Wu, Q. Li, and B. He, "Practical vertical federated learning with unsupervised representation learning," *IEEE Transactions on Big Data (TBD)*, 2022.

[116] L. Gao, X. Ma, J. Lin, and J. Callan, "Precise zero-shot dense retrieval without relevance labels," *arXiv preprint arXiv:2212.10496*, 2022.

[117] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray *et al.*, "Training language models to follow instructions with human feedback," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, 2022, pp. 27 730–27 744.

[118] G. Izacard, M. Caron, L. Hosseini, S. Riedel, P. Bojanowski, A. Joulin, and E. Grave, "Unsupervised dense information retrieval with contrastive learning," *Transactions on Machine Learning Research (TMLR)*, 2022.

[119] A. Madani, B. Krause, E. R. Greene, S. Subramanian, B. P. Mohr, J. M. Holton, J. L. Olmos Jr, C. Xiong, Z. Z. Sun, R. Socher *et al.*, "Large language models generate functional protein sequences across diverse families," *Nature Biotechnology*, pp. 1–8, 2023.

[120] R. Maclin, J. W. Shavlik *et al.*, "Combining the predictions of multiple classifiers: Using competitive learning to initialize neural networks," in *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI)*, vol. 95, 1995, pp. 524–531.

[121] D. Opitz and J. Shavlik, "Generating accurate and diverse members of a neural-network ensemble," *Advances in Neural Information Processing Systems (NIPS)*, vol. 8, 1995.

[122] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," *Advances in Neural Information Processing Systems (NIPS)*, vol. 30, 2017.

[123] K. Pillutla, S. M. Kakade, and Z. Harchaoui, "Robust aggregation for federated learning," *IEEE Transactions on Signal Processing (TSP)*, vol. 70, pp. 1142–1154, 2022.

[124] Z. Sun, P. Kairouz, A. T. Suresh, and H. B. McMahan, "Can you really backdoor federated learning?" in *NeurIPS 2019 Workshop on Federated Learning for User Privacy and Data Confidentiality*, 2019.

[125] A. Reisizadeh, A. Mokhtari, H. Hassani, A. Jadbabaie, and R. Pedarsani, "FedQAP: A communication-efficient federated learning method with periodic averaging and quantization," in *International*

Conference on Artificial Intelligence and Statistics (AISTATS). PMLR, 2020, pp. 2021–2031.

[126] H. Wang, M. Yurochkin, Y. Sun, D. Papailiopoulos, and Y. Khazaeni, "Federated learning with matched averaging," in *Proceedings of the 8th International Conference on Learning Representations (ICLR)*, 2020.

[127] F. Yu, W. Zhang, Z. Qin, Z. Xu, D. Wang, C. Liu, Z. Tian, and X. Chen, "Fed2: Feature-aligned federated learning," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 2066–2074.

[128] G. K. Gudur, B. S. Balaji, and S. K. Perepu, "Resource-constrained federated learning with heterogeneous labels and models," in *KDD 2020 Workshop on Artificial Intelligence of Things*, 2020.

[129] Z. Qu, X. Li, R. Duan, Y. Liu, B. Tang, and Z. Lu, "Generalized federated learning via sharpness aware minimization," in *Proceedings of the 39th International Conference on Machine Learning (ICML)*. PMLR, 2022, pp. 18 250–18 280.

[130] E. Jeong, S. Oh, H. Kim, J. Park, M. Bennis, and S.-L. Kim, "Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data," in *NeurIPS 2018 Workshop on Machine Learning on the Phone and other Consumer Devices*, 2018.

[131] C. Jin, X. Chen, Y. Gu, and Q. Li, "FedDyn: A dynamic and efficient federated distillation approach on recommender system," in *2022 IEEE 28th International Conference on Parallel and Distributed Systems (ICPADS)*. IEEE, 2023, pp. 786–793.

[132] M. Yurochkin, M. Agarwal, S. Ghosh, K. Greenewald, N. Hoang, and Y. Khazaeni, "Bayesian nonparametric federated learning of neural networks," in *Proceedings of the 36th International Conference on Machine Learning (ICML)*. PMLR, 2019, pp. 7252–7261.

[133] M. Yurochkin, M. Agarwal, S. Ghosh, K. Greenewald, and N. Hoang, "Statistical model aggregation via parameter matching," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 32, 2019.

[134] T. C. Lam, N. Hoang, B. K. H. Low, and P. Jaillet, "Model fusion for personalized learning," in *Proceedings of the 38th International Conference on Machine Learning (ICML)*. PMLR, 2021, pp. 5948–5958.

[135] S. Su, B. Li, and X. Xue, "One-shot federated learning without server-side training," *Neural Networks*, vol. 164, pp. 203–215, 2023.

[136] S. Wang, X. Li, J. Sun, and Z. Xu, "Training networks in null space of feature covariance for continual learning," in *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 184–193.

[137] Y. Kong, L. Liu, Z. Wang, and D. Tao, "Balancing stability and plasticity through advanced null space in continual learning," in *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXVI*. Springer, 2022, pp. 219–236.

[138] C.-M. Feng, B. Li, X. Xu, Y. Liu, H. Fu, and W. Zuo, "Learning federated visual prompt in null space for mri reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[139] D. Sui, Y. Chen, J. Zhao, Y. Jia, Y. Xie, and W. Sun, "FedED: Federated learning via ensemble distillation for medical relation extraction," in *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, 2020, pp. 2118–2128.

[140] T. Lin, L. Kong, S. U. Stich, and M. Jaggi, "Ensemble distillation for robust model fusion in federated learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020, pp. 2351–2363.

[141] N. Guha, A. Talwalkar, and V. Smith, "One-shot federated learning," in *NeurIPS 2018 Workshop on Machine Learning on the Phone and other Consumer Devices*, 2018.

[142] H. Chen and W. Chao, "FedBE: Making bayesian model ensemble applicable to federated learning," in *Proceedings of the 9th International Conference on Learning Representations (ICLR)*, 2021.

[143] C. Xie, D.-A. Huang, W. Chu, D. Xu, C. Xiao, B. Li, and A. Anandkumar, "PerAda: Parameter-efficient and generalizable federated learning personalization with guarantees," *arXiv preprint arXiv:2302.06637*, 2023.

[144] Y. J. Cho, A. Manoel, G. Joshi, R. Sim, and D. Dimitriadis, "Heterogeneous ensemble knowledge transfer for training large models in federated learning," in *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI)*, 2022, pp. 2881–2887.

[145] X. Yao, T. Huang, C. Wu, R. Zhang, and L. Sun, "Towards faster and better federated learning: A feature fusion approach," in *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 175–179.

[146] C. Wu, F. Wu, L. Lyu, Y. Huang, and X. Xie, "Communication-efficient federated learning via knowledge distillation," *Nature communications*, vol. 13, no. 1, p. 2032, 2022.

[147] G. Lee, M. Jeong, Y. Shin, S. Bae, and S.-Y. Yun, "Preservation of the global knowledge by not-true distillation in federated learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[148] J. Kim, G. Kim, and B. Han, "Multi-level branched regularization for federated learning," in *Proceedings of the 39th International Conference on Machine Learning (ICML)*. PMLR, 2022, pp. 11 058–11 073.

[149] Y. He, Y. Chen, X. Yang, Y. Zhang, and B. Zeng, "Class-wise adaptive self distillation for heterogeneous federated learning," in *Proceedings of the 36th AAAI Conference on Artificial Intelligence*, vol. 22, 2022, pp. 12 967–12 968.

[150] M. Mendieta, T. Yang, P. Wang, M. Lee, Z. Ding, and C. Chen, "Local learning matters: Rethinking data heterogeneity in federated learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 8397–8406.

[151] J. Zhang, X. Zhang, X. Zhang, D. Hong, R. K. Gupta, and J. Shang, "Navigating alignment for non-identical client class sets: A label name-anchored federated learning framework," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023, pp. 3297–3308.

[152] W. Jeong, J. Yoon, E. Yang, and S. J. Hwang, "Federated semi-supervised learning with inter-client consistency & disjoint learning," in *Proceedings of the 9th International Conference on Learning Representations (ICLR)*, 2021.

[153] S. A. Aketi and K. Roy, "Cross-feature contrastive loss for decentralized deep learning on heterogeneous data," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2024, pp. 12–21.

[154] U. Michieli and M. Ozay, "Prototype guided federated learning of visual feature representations," *arXiv preprint arXiv:2105.08982*, 2021.

[155] Y. Tan, G. Long, L. Liu, T. Zhou, Q. Lu, J. Jiang, and C. Zhang, "FedProto: Federated prototype learning across heterogeneous clients," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 8, 2022, pp. 8432–8440.

[156] D. Jiang, C. Shan, and Z. Zhang, "Federated learning algorithm based on knowledge distillation," in *2020 International Conference on Artificial Intelligence and Computer Engineering (ICAICE)*. IEEE, 2020, pp. 163–167.

[157] W. Huang, M. Ye, and B. Du, "Learn from others and be yourself in heterogeneous federated learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 10 143–10 153.

[158] H. Jin, D. Bai, D. Yao, Y. Dai, L. Gu, C. Yu, and L. Sun, "Personalized edge intelligence via federated self-knowledge distillation," *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, vol. 34, no. 2, pp. 567–580, 2022.

[159] X. Liang, Y. Lin, H. Fu, L. Zhu, and X. Li, "RSCFed: random sampling consensus federated semi-supervised learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 10 154–10 163.

[160] Q. Li, B. He, and D. Song, "Practical one-shot federated learning for cross-silo setting," in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI)*. International Joint Conferences on Artificial Intelligence Organization, 2021, pp. 1484–1490.

[161] D. Li and J. Wang, "FedMD: Heterogenous federated learning via model distillation," in *NeurIPS 2019 Workshop on Federated Learning for User Privacy and Data Confidentiality*, 2019.

[162] X. Gong, A. Sharma, S. Karanam, Z. Wu, T. Chen, D. Doermann, and A. Innanje, "Ensemble attention distillation for privacy-preserving federated learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 15 076–15 086.

[163] L. Sun and L. Lyu, "Federated model distillation with noise-free differential privacy," in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI)*. International Joint Conferences on Artificial Intelligence Organization, 8 2021, pp. 1563–1570.

[164] F. Sattler, T. Korjakow, R. Rischke, and W. Samek, "FedAUX: Leveraging unlabeled auxiliary data in federated learning," *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, 2021.

[165] X. Fang and M. Ye, "Robust federated learning with noisy and heterogeneous clients," in *Proceedings of the IEEE/CVF Conference*

on Computer Vision and Pattern Recognition (CVPR), June 2022, pp. 10 072–10 081.

[166] X. Gong, A. Sharma, S. Karanam, Z. Wu, T. Chen, D. Doermann, and A. Innanje, "Preserving privacy in federated learning with ensemble cross-domain knowledge distillation," in *Proceedings of the 36th AAAI Conference on Artificial Intelligence*, vol. 36, no. 11, 2022, pp. 11 891–11 899.

[167] H. Chang, V. Shejwalkar, R. Shokri, and A. Houmansadr, "Cronus: Robust and heterogeneous collaborative learning with black-box knowledge transfer," in *NeurIPS 2021 Workshop on New Frontiers in Federated Learning: Privacy, Fairness, Robustness, Personalization and Data Ownership*, 2021.

[168] J. Zhang, S. Guo, X. Ma, H. Wang, W. Xu, and F. Wu, "Parameterized knowledge transfer for personalized federated learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 34, 2021, pp. 10 092–10 104.

[169] S. Itahara, T. Nishio, Y. Koda, M. Morikura, and K. Yamamoto, "Distillation-based semi-supervised federated learning for communication-efficient collaborative training with non-iid private data," *IEEE Transactions on Mobile Computing (TMC)*, vol. 22, no. 1, pp. 191–205, 2021.

[170] J. Zhang, C. Chen, B. Li, L. Lyu, S. Wu, S. Ding, C. Shen, and C. Wu, "DENSE: Data-free one-shot federated learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[171] C. E. Heinbaugh, E. Luz-Ricca, and H. Shao, "Data-free one-shot federated learning under very high statistical heterogeneity," in *Proceedings of the 11th International Conference on Learning Representations (ICLR)*, 2023.

[172] Z. Zhu, J. Hong, and J. Zhou, "Data-free knowledge distillation for heterogeneous federated learning," in *Proceedings of the 38th International Conference on Machine Learning (ICML)*. PMLR, 2021, pp. 12 878–12 889.

[173] L. Zhang, L. Shen, L. Ding, D. Tao, and L.-Y. Duan, "Fine-tuning global model via data-free knowledge distillation for non-iid federated learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 10 174–10 183.

[174] H. Wang, H. Zhao, Y. Wang, T. Yu, J. Gu, and J. Gao, "FedKC: Federated knowledge composition for multilingual natural language understanding," in *Proceedings of the ACM Web Conference 2022*, 2022, pp. 1839–1850.

[175] C. Fan, J. Hu, and J. Huang, "Private semi-supervised federated learning," in *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI)*, 2022, pp. 2009–2015.

[176] J. Dean, G. S. Corrado, R. Monga, K. Chen, M. Devin, Q. V. Le, M. Z. Mao, M. Ranzato, A. Senior, P. Tucker *et al.*, "Large scale distributed deep networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems (NeurIPS)*, 2012, pp. 1223–1231.

[177] Q. Li, Y. Diao, Q. Chen, and B. He, "Federated learning on non-iid data silos: An experimental study," in *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE, 2022, pp. 965–978.

[178] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "FitNets: Hints for thin deep nets," in *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2015.

[179] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials (COMST)*, vol. 22, no. 3, pp. 2031–2063, 2020.

[180] Z. Luo, Y. Wang, Z. Wang, Z. Sun, and T. Tan, "FedIris: Towards more accurate and privacy-preserving iris recognition via federated template communication," in *CVPR 2022 Workshop on Federated Learning for Computer Vision*, 2022, pp. 3357–3366.

[181] T. Furlanello, Z. Lipton, M. Tschannen, L. Itti, and A. Anandkumar, "Born again neural networks," in *Proceedings of the 35th International Conference on Machine Learning (ICML)*. PMLR, 2018, pp. 1607–1616.

[182] N. Dvornik, C. Schmid, and J. Mairal, "Diversity with cooperation: Ensemble methods for few-shot classification," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 3723–3731.

[183] P. Vepakomma, O. Gupta, T. Swedish, and R. Raskar, "Split learning for health: Distributed deep learning without sharing raw patient data," in *ICLR 2019 Workshop on AI for Social Good*, 2019.

[184] C. Thapa, P. C. M. Arachchige, S. Camtepe, and L. Sun, "Splitfed: When federated learning meets split learning," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, vol. 36, no. 8, 2022, pp. 8485–8493.

[185] J. Kim, S. Shin, Y. Yu, J. Lee, and K. Lee, "Multiple classification with split learning," in *The 9th International Conference on Smart Media and Applications*, 2020, pp. 358–363.

[186] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2015, pp. 1322–1333.

[187] R. G. Lopes, S. Fenu, and T. Starner, "Data-free knowledge distillation for deep neural networks," *NIPS 2017 Workshop on Learning with Limited Labeled Data: Weak Supervision and Beyond*, 2017.

[188] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *Proceedings of the 2nd International Conference on Learning Representations (ICLR)*, 2014.

[189] N. Shi, F. Lai, R. A. Kontar, and M. Chowdhury, "Fed-ensemble: Ensemble models in federated learning for improved generalization and uncertainty quantification," *IEEE Transactions on Automation Science and Engineering (T-ASE)*, 2023.

[190] N. Wang, W. Feng, M. Duan, F. Liu, S.-K. Ng *et al.*, "Data-free diversity-based ensemble selection for one-shot federated learning," *Transactions on Machine Learning Research*, 2023.

[191] C. J. Geyer, "Practical markov chain monte carlo," *Statistical Science*, vol. 7, no. 4, pp. 473–483, 1992.

[192] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[193] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *Journal of artificial intelligence research (JAIR)*, vol. 16, no. 1, pp. 321–357, 2002.

[194] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *Proceedings of the 6th International Conference on Learning Representations (ICLR)*, 2018.

[195] S. C. Wong, A. Gatt, V. Stamatescu, and M. D. McDonnell, "Understanding data augmentation for classification: when to warp?" in *Proceedings of the 2016 international conference on digital image computing: techniques and applications (DICTA)*. IEEE, 2016, pp. 1–6.

[196] T. Yoon, S. Shin, S. J. Hwang, and E. Yang, "FedMix: Approximation of mixup under mean augmented federated learning," in *Proceedings of the 9th International Conference on Learning Representations (ICLR)*, 2021.

[197] K. Zhang, C. Yang, X. Li, L. Sun, and S. M. Yiu, "Subgraph federated learning with missing neighbor generation," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 34, 2021, pp. 6671–6682.

[198] Y. Cheng, L. Zhang, and A. Li, "GFL: Federated learning on non-iid data via privacy-preserving synthetic data," in *2023 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 2023, pp. 61–70.

[199] W. Hao, M. El-Khamy, J. Lee, J. Zhang, K. J. Liang, C. Chen, and L. C. Duke, "Towards fair federated learning with zero-shot data augmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2021, pp. 3310–3319.

[200] H. Cha, J. Park, H. Kim, S.-L. Kim, and M. Bennis, "Federated reinforcement distillation with proxy experience memory," in *IJCAI 2019 Workshop on Federated Learning for User Privacy and Data Confidentiality*, 2019.

[201] S. Yu, J. Hong, H. Wang, Z. Wang, and J. Zhou, "Turning the curse of heterogeneity in federated learning into a blessing for out-of-distribution detection," in *Proceedings of the 11th International Conference on Learning Representations (ICLR)*, 2023.

[202] M. Yang, S. Su, B. Li, and X. Xue, "Exploring one-shot semi-supervised federated learning with a pre-trained diffusion model," *arXiv preprint arXiv:2305.04063*, 2023.

[203] Q. Liu, C. Chen, J. Qin, Q. Dou, and P.-A. Heng, "FedDG: Federated domain generalization on medical image segmentation via episodic learning in continuous frequency space," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 1013–1023.

[204] R. Pi, W. Zhang, Y. Xie, J. Gao, X. Wang, S. Kim, and Q. Chen, "DynaFed: Tackling client data heterogeneity with global dynamics," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 12 177–12 186.

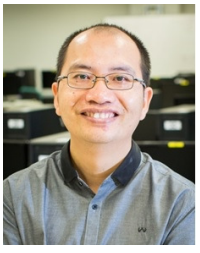
[205] Z. Li, J. Shao, Y. Mao, J. H. Wang, and J. Zhang, "Federated learning with GAN-based data synthesis for non-IID clients," in *IJCAI 2022*

Workshop on Trustworthy Federated Learning. Springer, 2022, pp. 17–32.
[206] E. Diao, J. Ding, and V. Tarokh, "SemiFL: Semi-supervised federated learning for unlabeled clients with alternate training," vol. 35, 2022, pp. 17 871–17 884.
[207] L. Zhang, D. Wu, and X. Yuan, "FedZKT: Zero-shot knowledge transfer towards resource-constrained federated learning with heterogeneous on-device models," in *Proceedings of the IEEE 42nd International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2022, pp. 928–938.
[208] T. Che, Z. Zhang, Y. Zhou, X. Zhao, J. Liu, Z. Jiang, D. Yan, R. Jin, and D. Dou, "Federated fingerprint learning with heterogeneous architectures," in *2022 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2022, pp. 31–40.
[209] Y. Shang, B. Duan, Z. Zong, L. Nie, and Y. Yan, "Lipschitz continuity guided knowledge distillation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 10 675–10 684.
[210] C. He, M. Annavaram, and S. Avestimehr, "Group knowledge transfer: Federated learning of large cnns at the edge," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33. Curran Associates, Inc., 2020, pp. 14 068–14 080.
[211] J. Chen, J. Wang, T. Peng, Y. Sun, P. Cheng, S. Ji, X. Ma, B. Li, and D. Song, "Copy, right? a testing framework for copyright protection of deep learning models," in *IEEE Symposium on Security and Privacy (SP)*. IEEE, 2022, pp. 824–841.
[212] H. Jia, H. Chen, J. Guan, A. S. Shamsabadi, and N. Papernot, "A zest of LIME: Towards architecture-independent model distances," in *Proceedings of the 10th International Conference on Learning Representations (ICLR)*, 2022.
[213] J. W. Vaughan, "Making better use of the crowd: How crowdsourcing can advance machine learning research," *The Journal of Machine Learning Research (JMLR)*, vol. 18, no. 1, pp. 7026–7071, 2018.
[214] B. Yuan, Y. He, J. Davis, T. Zhang, T. Dao, B. Chen, P. S. Liang, C. Re, and C. Zhang, "Decentralized training of foundation models in heterogeneous environments," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, 2022, pp. 25 464–25 477.
[215] H. Dias and N. Meratnia, "Blocklearning: A modular framework for blockchain-based vertical federated learning," in *Inernational Conference on Ubiquitous Security (UbiSec)*. Springer, 2022, pp. 319–333.
[216] R. Blythman, M. Arshath, S. Vivona, J. Smékal, and H. Shaji, "Opportunities for decentralized technologies within ai hubs," in *NeurIPS 2022 Workshop on Decentralization and Trustworthy Machine Learning in Web3: Methodologies, Platforms, and Applications*, 2022.
[217] Y. Deng, T. Han, and N. Zhang, "FLeX: Trading edge computing resources for federated learning via blockchain," in *IEEE Conference on Computer Communications Workshops*. IEEE, 2021, pp. 1–2.
[218] S. Guo, F. Zhang, S. Guo, S. Xu, and F. Qi, "Blockchain-assisted privacy-preserving data computing architecture for web3," *IEEE Communications Magazine*, vol. 61, no. 8, pp. 28–34, 2023.
[219] Z. Batool, K. Zhang, and M. Toews, "FL-MAB: client selection and monetization for blockchain-based federated learning," in *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing (SAC)*, 2022, pp. 299–307.
[220] J. Steeves, A. Shaabana, Y. Hu, F. Luus, S. T. Liu, and J. D. Tasker-Steeves, "Incentivizing intelligence: The bittensor approach," in *NeurIPS 2022 Workshop on Decentralization and Trustworthy Machine Learning in Web3: Methodologies, Platforms, and Applications*, 2022.
[221] T. McConaghy, "Ocean protocol: Tools for the web3 data economy," in *Handbook on Blockchain*. Springer, 2022, pp. 505–539.
[222] M. Ryabinin and A. Gusev, "Towards crowdsourced training of large neural networks using decentralized mixture-of-experts," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020, pp. 3659–3672.
[223] A. Borzunov, M. Ryabinin, T. Dettmers, Q. Lhoest, L. Saulnier, M. Diskin, Y. Jernite, and T. Wolf, "Training transformers together," in *Proceedings of the NeurIPS 2021 Competitions and Demonstrations Track*. PMLR, 2022, pp. 335–342.
[224] M. Ryabinin, E. Gorbunov, V. Plokhotnyuk, and G. Pekhimenko, "Moshpit SGD: Communication-efficient decentralized training on heterogeneous unreliable devices," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 34, 2021, pp. 18 195–18 211.
[225] M. Atre, B. Jha, and A. Rao, "Distributed deep learning using volunteer computing-like paradigm," in *2021 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*. IEEE, 2021, pp. 933–942.
[226] W. Kuang, B. Qian, Z. Li, D. Chen, D. Gao, X. Pan, Y. Xie, Y. Li, B. Ding, and J. Zhou, "FederatedScope-LLM: A comprehensive package for fine-tuning large language models in federated learning," *arXiv preprint arXiv:2309.00363*, 2023.
[227] Z. Qin, D. Chen, B. Qian, B. Ding, Y. Li, and S. Deng, "Federated full-parameter tuning of billion-sized language models with communication cost under 18 kilobytes," *arXiv preprint arXiv:2312.06353*, 2023.
[228] F. Berdoz, A. Singh, M. Jaggi, and R. Raskar, "Scalable collaborative learning via representation sharing," in *NeurIPS 2022 Workshop on Decentralization and Trustworthy Machine Learning in Web3: Methodologies, Platforms, and Applications*, 2022.
[229] H. Woisetschläger, A. Isenko, S. Wang, R. Mayer, and H.-A. Jacobsen, "Federated fine-tuning of llms on the very edge: The good, the bad, the ugly," *arXiv preprint arXiv:2310.03150*, 2023.
[230] T. Fan, Y. Kang, G. Ma, W. Chen, W. Wei, L. Fan, and Q. Yang, "FATE-LLM: A industrial grade federated learning framework for large language models," *arXiv preprint arXiv:2310.10049*, 2023.

Moming Duan is a Research Fellow at Institute of Data Science, National University of Singapore. He received PhD degree in computer science from Chongqing University (2017-2022). He has published several papers in prestigious conferences and journals including TPDS, WWW, ICCD, Neural Networks, TMLR. His research interests include Federated Learning, Collaborative Machine Learning, and AI Licensing.

Qinbin Li is a postdoc at UC Berkeley. He received his PhD degree from National University of Singapore. He is a recipient of Google PhD Fellowship 2021. His research interests include federated learning, trustworthy machine learning, and systems.

Linshan Jiang is currently a Research Fellow at Institute of Data Science, National University of Singapore. He obtained his Ph.D. degree in computer science and engineering from Nanyang Technological University, Singapore in 2022. He has published several papers on the top conference and journals in CPS-IoT. His research interests focus on the privacy and security in the distributed AI system, including federated/collaborative learning, blockchain-enabled AI and resilient AIoT system.



Bingsheng He received the bachelor degree in computer science from Shanghai Jiao Tong University (1999-2003), and the PhD degree in computer science in Hong Kong University of Science and Technology (2003-2008). He is a Professor in School of Computing, National University of Singapore. His research interests are high performance computing, distributed and parallel systems, and database systems.