FedH2L: A Federated Learning Approach with Model and Statistical Heterogeneity

1st Yiying Li*
Artificial Intelligence Research Center
DII, Academy of Military Sciences
Beijing, China
liyiying10@nudt.edu.cn

2nd Wei Zhou*
China Xi'an Satellite Control Center
Xi'an, China
zhouwei14@nudt.edu.cn

3rd Haibo Mi†
College of Information and Communication
National University of Defense Technology
Wuhan, China
haibo_mihb@126.com

4th Yijie Wang
College of Computer
National University of Defense Technology
Changsha, China
wwyyjj1971@vip.sina.com

5th Huaimin Wang
College of Computer
National University of Defense Technology
Changsha, China
whm_w@163.com

***Abstract*—Federated learning (FL) enables distributed participants to collectively learn a strong global model without sacrificing their individual data privacy. Mainstream FL approaches require each participant to share a common network architecture and further assume that data are sampled IID across participants. However, in real-world deployments, participants may require heterogeneous network architectures; and the data distribution is almost non-uniform. To address these issues we introduce FedH2L, which is agnostic to the model architecture and robust to different data distributions across participants. In contrast to approaches sharing parameters or gradients, FedH2L relies on mutual distillation, exchanging only posteriors on a shared seed set between participants in a decentralized manner. This makes it extremely bandwidth efficient, model agnostic, and crucially produces models capable of performing well on the whole data distribution when learning from heterogeneous silos.**

***Index Terms*—Federated Learning, Model heterogeneity, Statistical heterogeneity, Domain shift, Mutual learning**

I. Introduction

Today, artificial intelligence (AI) is showing its strengths in almost every walk of life. To fully realize AI's benefits, we wish to learn models across as much data as possible, but this data is often held privately across diverse users or organizations. To enable collective benefit from AI while maintaining data privacy, Federated Learning (FL) [1]–[3] algorithms aim to train a global model based on the efforts of distributed participants' data and resources.

There are a number of actively researched challenges however to achieving this vision [4], including system/model heterogeneity, statistical heterogeneity, bandwidth requirements, and residual privacy concerns. Different FL methods provide different trade-offs in their requirements on these axes along in the accuracy they ultimately provide [4]. We propose a novel FL method FedH2L, which primarily aims to support significant statistical and model heterogeneity across participants to achieve the personalized federated learning [5], and also provides benefits for bandwidth and privacy.

System heterogeneity usually refers to different computation and bandwidth resources among participants leading to different update rates, and mainstream research aims to alleviate the impact of stragglers in FL setting [6]. However, participants more generally may require fundamentally different model architectures [7]. This can occur in edge or device-based FL due to devices' different resource constraints, or in B2B FL due to each organization wishing to keep their particular optimised model architecture private. Statistical heterogeneity refers to the diversity in each user's data distribution [6], [8]. We aim to learn a strong federated system capable of performing on the global data distribution, although learning takes place locally in each user's private data silo.

Mainstream FL typically proceed by sharing parameters or gradients at each iteration. This means they are bandwidth-constrained, as contemporary models can have millions of parameters. Furthermore, most FL methods require a centralized server to aggregate results from each participant, no matter for the parameter-based methods [2], [9], [10] or for the recent prototype-based methods [11]. This requires a globally trusted authority, and provides a single point of failure. We instead present a decentralized peer-to-peer approach that is robust and extremely communication efficient. Moreover, parameter and gradient sharing strategies can also incur a residual privacy risk due to attack vulnerability [12], [13]. Our method shares no parameters, thus eliminating such vulnerability.

In this paper, we present a novel FL algorithm FedH2L, which significantly advances the practical applicability of FL by enabling simultaneous system and statistical heterogeneity across participants. Instead of exchanging gradients/parameters, FedH2L exchanges predictions on small shared seed set distributed to participants in advance [7],

*Equal contribution.
†Corresponding author.

and performs decentralized global optimization by mutual learning [14], thus enabling model-agnostic FL. This strategy also eliminates privacy concerns of parameter/gradient sharing, and requires orders of magnitude lower communication cost than sharing models/gradients. Moreover, we also pay attention to the issue of managing statistical heterogeneity across participants [4], [15], [16]. In FedH2L, each participant optimizes a multi-task objective of fitting its local data, and distillation on the seed set for knowledge sharing across peers. This multi-task optimization is challenging when there is the significant distribution shift, which can lead to gradient conflict [17] and poor solutions. To this end we introduce a new optimization strategy to find the best non-conflicting gradient for simultaneously fitting local data and incorporating feedback from peers. Our contributions are:

- We introduce FedH2L, which uniquely provides simultaneous support for a challenging set of real world conditions including *heterogeneous models* across peers, robust *decentralized* learning, *privacy preserving* parameter/gradient-free communication, while being desired to maximise performance under *heterogeneous data statistics* across peers. See Table I for comparison.
- To provide best performance under conditions of heterogeneous data statistics across peers, we introduce a new optimization strategy to find the gradient update that does not conflict between local and global update cues.
- We conduct extensive experiments on several multi-domain datasets: Rotated MNIST [18], PACS [19], and Office-Home [20]. Compared to the baselines, we improve the model performance across all domains, demonstrating the effectiveness of FedH2L.

II. Related Work

Personalized Federated Learning Recently, the personalized federated learning (PFL) [5], [26]–[28] is proposed to address the fundamental challenges of FL on the heterogeneity. Most of the PFL researches only focus on the data Non-IID heterogeneity using common methods like "FL training + local adaptation" [29], or focus on the clients' model heterogeneity by introducing tricks on network layer architectures [9] or model similarities [5], [30]. Our work also belongs to the research of PFL, and we further consider the data statistical heterogeneity, model system heterogeneity, bandwidth efficiency, privacy requirements and decentralization at the same time. We will give the detailed analysis of these multiple aspects as follows.

System and Statistical Heterogeneity FL aims to train models over remote devices, while keeping data localized. FL faces many challenges [4], and the important one is the heterogeneity on the system and statistical aspects. Participants may vary on hardware, compute and bandwidth resources. These system characteristics make issues such as stragglers prevalent. Existing studies mainly focus on the active sampling [31], [32]. However, a more severe challenge in system heterogeneity is the model heterogeneity of different architectures among participants. FedMD [7] introduce model heterogeneity based on knowledge distillation but with a centralized communication server. FML [21] trains extra heterogeneous models by learning from participants' distributed homogeneous models. FedGKT [22] trains small CNNs on edges and periodically transfer their knowledge (e.g., extracted features) instead of data by knowledge distillation to a server-side large CNN. FedProto [11] uses the prototypes for global aggregation on the server from different clients, and sends global prototypes back for local regularization. In addition, for methods like FedPer [9] and FedRep [10], although clients have their final personalized layers, they are asked to hold the same feature extractor model for parameter sharing, and thus cannot support system heterogeneity technically.

In almost every substantive use case of FL (e.g., medical data across hospitals, industrial data across corporations) participants generate and collect data in a Non-IID distributed manner, leading to statistical shift among them. To tackle such statistical heterogeneity, FedProx [6] provides convergence guarantees based on FedAvg [2] over Non-IID data. FedAgnostic [8] learns a centralized model that is optimized for any target distribution formed by a mixture of participants' distributions. FedSEM [23] and FedCluster [24] are classical methods using models clustering on the server with Non-IID data among clients. FML [21], FedGKT [22], FedMD [7], FedProto [11], FedPer [9] and FedRep [10] also have the opportunities to cope with the Non-IID data because they have individualized models or some layers for each user but still controlled by a central server. We aim to handle both model and statistical heterogeneity in a decentralized manner without the need of a centralized model or extra local models.

Bandwidth and Privacy Requirements Communication is a critical bottleneck in FL. Current communication-efficient methods mainly consider: (1) Reducing the total number of communication rounds; (2) Reducing the size of transmitted messages at each round. But such methods [2], [6], [8], [21] still proceed by sharing millions of parameters/gradients as the communicated messages, which means the best case bandwidth requirement is still orders of magnitude worse than FedH2L. Additionally, sharing parameters create attack vulnerability [12], [13], increasing the privacy risk. The aggregation of parameters/gradients also usually asks for a centralized trusted authority [2], [6] which may lead to the single point of failure. FedH2L is a communication-efficient decentralized peer-to-peer method without sharing any high-overhead and privacy compromising model parameters/gradients.

Multi-task Optimization Instead of learning a single global model, we simultaneously learn distinct local models with a multi-task objective based on local and remote teaching signals. A similar federated work in multi-task setting is MOCHA [25], but each local model only focuses on the performance on its own task, instead of the multi-task objective. A key challenge in multi-task learning [17], [33] is the conflicting gradients, especially when there is statistical heterogeneity across tasks/participants. Yu. et al [17] propose a gradient

TABLE I
COMPARISON OF FL FRAMEWORKS.

Method	Hetero. Models	Decentr.	ParamFree	BW	Hetero. Data
FedAvg [2]	✗	✗	✗	-	-
FedProx [6]	✗	✗	✗	-	+
FML [21]	✗/✓	✗	✗	-	+
FedGKT [22]	✗/✓	✗	✓	+	+
FedMD [7]	✓	✗	✓	+	+
FedAgnostic [8]	✗	✗	✗	-	+
FedSEM [23]	✗	✗	✗	+	+
FedCluster [24]	✗	✗	✗	-	+
FedPer [9]/FedRep [10]	✗	✗	✗	-	+
FedProto [11]	✓	✗	✓	+	+
MOCHA [25]	✗/✓	✗	✗	+	+
FedH2L(Ours)	✓	✓	✓	+	+

surgery to train a single model for multiple tasks by projecting each task gradient onto normal plane of the other. In contrast, we propose a novel optimization strategy to get non-conflicting gradients for each participant's model so as to fit local data and learn from other peers reliably and simultaneously.

III. METHODOLOGY

Here we introduce the details of FedH2L. Assume there are N nodes in the FL network, holding data with potentially distinct distributions $\mathcal{D} = \{D_1, D_2, \ldots, D_N\}$. There is also a public dataset D_{pub} in the same label space that everyone can access. The data on each node contains a set of data-label pairs, i.e., $D_i = \{X_i, Y_i\}$. We also split D_i into its private data which must only be kept locally, validation data and test data, i.e., $D_i = \{D_i^{\text{loc}}, D_i^{\text{val}}, D_i^{\text{test}}\}$. We aim to learn a federated system that aggregates knowledge from all nodes, but without sacrificing each node's data privacy, and without assuming a common model architecture.

We consider the homogeneous multi-domain setting [19], where all nodes share the same label set Y_i covering the same M classes, but have different data distributions. For example, one can consider medical images of the same set of diseases, but collected by different machines in different hospitals.

Each node i uses a network parameterized by θ_i, which can be uniquely customized and private to each node. No centralized model is used in FedH2L. But the goal is that after learning, each node's model θ_i should incorporate the knowledge of all nodes' datasets, and be able to perform well on any node's data distribution. The workflow is divided into two iterative phases: local and global optimization.

A. Local Optimization

Local optimization for a node follows the conventional supervised learning paradigm using locally available data. Denoting i-th node's network as f_{θ_i}, we optimize the cross-entropy (CE) loss to obtain gradient g_i^{loc}:

$$\text{minimize}_{\theta_i} \ell^{(\text{CE})}(f_{\theta_i}(\mathbf{x}_i^{\text{loc}}), \mathbf{y}_i^{\text{loc}}), \tag{1}$$

$$g_i^{\text{loc}} = \nabla_{\theta_i} \ell^{(\text{CE})}(f_{\theta_i}(\mathbf{x}_i^{\text{loc}}), \mathbf{y}_i^{\text{loc}}). \tag{2}$$

Note that $f_{\theta_i}(\mathbf{x}_i^{\text{loc}})$ provides soft labels $\mathbf{p}_i^{\text{loc}}$ corresponding to the output of the final softmax layer of the network, which are compared against the ground truth one-hot labels.

B. Global Mutual Optimization

The next step is for each node to learn from its peers. To achieve this in a decentralized manner and under conditions of heterogeneous model architecture, we exploit model distillation. Different from the conventional distillation [34] where a strong teacher trains multiple students, the federated network in FedH2L acts as an ensemble of students that all teach each other.

Preparation for mutual learning We randomly sample a batch $d_i^{\text{pub}} = (\mathbf{x}_i^{\text{pub}}, \mathbf{y}_i^{\text{pub}})$ from D_{pub} on each domain/node and compute the soft labels $\mathbf{p}_i^{\text{pub(i)}}$. Note that the superscript i denotes the data is domain i's sampled public data, and the subscript i denotes the network f_{θ_i} making the prediction. To assess the quality of predictions, we also get the accuracy Acc_i over the batch public data in each domain. Each node i broadcasts $[\mathbf{p}_i^{\text{pub(i)}}, Acc_i]$ as its teaching signal and associated teaching confidence, to others in the cohort. Note that the predictions in the teaching signal $\mathbf{p}_i^{\text{pub(i)}}$ are with respect to public data $\mathbf{x}_i^{\text{pub}}$, but contain knowledge from the local private data due to being made with the locally optimized network f_{θ_i}. The quantities $[\mathbf{p}_i^{\text{pub(i)}}, Acc_i]$ are the only parameters exchanged during the federated global mutual optimization step. So this approach is highly communication efficient.

Mutual Learning Each node i will act both as a student and a teacher, so there are $(N-1)$ teachers for each student f_{θ_i}. To improve each student node i's model based on teacher node j's data, it is trained to mimic the teacher's soft predictions on the batch public data on teacher. Specifically, each student i uses the Kullback Leibler (KL) Divergence loss $\ell_i^{(\text{KL})}$ as

$$\ell_i^{(\text{KL})} = \frac{1}{N-1} \sum_{j=1, j\neq i}^{N} Acc_j * D_{KL}(\mathbf{p}_j^{\text{pub(j)}} || \mathbf{p}_i^{\text{pub(j)}}), \tag{3}$$

where each teacher's contribution is weighted by its teaching confidence Acc_j, and where

$$D_{KL}(\mathbf{p}_j^{\text{pub(j)}}||\mathbf{p}_i^{\text{pub(j)}}) = \mathbb{E}_{\mathbf{p}_j}[\log \mathbf{p}_j^{\text{pub(j)}} - \log \mathbf{p}_i^{\text{pub(j)}}]. \quad (4)$$

In addition, besides the KL mimicry loss, we can also take advantage of the conventional supervised loss (CE loss):

$$\ell_i^{(\text{CE})} = \frac{1}{N-1} \sum_{j=1, j\neq i}^{N} \ell^{(\text{CE})}(f_{\theta_i}(\mathbf{x}_j^{\text{pub}}), \mathbf{y}_j^{\text{pub}}), \quad (5)$$

Thus we obtain the total mutual learning gradient for node i learning from the other nodes in the cohort:

$$g_i^{\text{pub}} = \nabla_{\theta_i}(\ell_i^{(\text{KL})} + \ell_i^{(\text{CE})}). \quad (6)$$

Summary In summary, each node trains using $g_i^{\text{loc}} = \nabla_{\theta_i}\ell^{(\text{CE})}$ on local data, and $g_i^{\text{pub}} = \nabla_{\theta_i}(\ell_i^{(\text{KL})} + \ell_i^{(\text{CE})})$ using public dataset across all domains.

C. Dealing with Statistical Heterogeneity

Our algorithm described so far enables decentralized FL of heterogeneous *models*. However, a key challenge is to best support the practically ubiquitous situation of *statistical* heterogeneity across domains. We hope that the local gradient g_i^{loc} can help to improve the performance on other domain's data (Cross-Domain Performance, CDP), and the remote teacher gradient g_i^{pub} can help to improve the performance on the local data (Within-Domain Performance, WDP). However this is challenging to achieve from a multi-task learning perspective, because the local learning gradient and peer learning gradient may conflict [17], [35], [36] under significant statistical shift.

Mutual Learning robust to statistical shift To perform student-teacher learning that is robust to distribution-shift across nodes, we propose to enforce the constraint:

$$\left\langle g_i^{\text{loc}}, g_i^{\text{pub}} \right\rangle \geq 0. \quad (7)$$

If this constraint is satisfied, then the remote teaching signal g_i^{pub} is unlikely to increase $\ell^{(\text{CE})}$ on each domain's local data, and we can safely use g_i^{pub} to directly update θ_i without risking negative within-domain performance. Thus we check if the constraint is violated, and project g_i^{pub} to the closest gradient $\tilde{g}_i$ (in the ℓ_2 norm sense) satisfying constraint (7). After projection $\tilde{g}_i$ is unlikely to increase $\ell^{(\text{CE})}$ or $\ell^{(\text{KL})}$. We perform:

$$\begin{aligned} \text{minimize}_{\tilde{g}_i} \quad & \frac{1}{2}\|g_i^{\text{pub}} - \tilde{g}_i\|_2^2 \\ \text{subject to} \quad & \langle \tilde{g}_i, g_i^{\text{loc}} \rangle \geq 0, \text{ for all } i \in N. \end{aligned} \quad (8)$$

Computation of $\tilde{g}_i$ We set $\tilde{g}_i \leftarrow project(g_i^{\text{pub}}, g_i^{\text{loc}})$. Here $project$ is the optimization of dual problem of Quadratic Program (QP). To solve (8) efficiently, inspired by the gradient processing in continual learning [35], we recall the primal of a QP [37] with inequality constraints:

$$\begin{aligned} \text{minimize}_z \quad & \frac{1}{2} z^\top C z + w^\top z \\ \text{subject to} \quad & Az \leq b, \end{aligned} \quad (9)$$

where $C \in \mathbb{R}^{p\times p}$ is a real symmetric matrix, $w \in \mathbb{R}^p$ is a real-valued vector , $A^\top \in \mathbb{R}^p$ is a real matrix, and $b \in \mathbb{R}$, p is the dimension of gradient vector.

The solution to the dual problem provides a lower bound to the primal QP problem. The Lagrangian dual of a QP is also a QP. Because original problem has constraint conditions, these can be built into the function. We write the Lagrangian function [38] as:

$$L(z, v) = \frac{1}{2} z^\top C z + w^\top z + v^\top (Az - b). \quad (10)$$

Defining the (Lagrangian) dual function as $g(v) = \inf_z L(z, v)$, we find an infimum of L, which occurs where the gradient is equal to zero, using $\nabla_z L(z, v) = 0$ and positive-definiteness of Q:

$$z^* = -C^{-1}(A^\top v + w). \quad (11)$$

So, the dual problem of (9) is:

$$\begin{aligned} \text{minimize}_v \quad & \frac{1}{2} v^\top A C^{-1} A^\top v + (w^\top C^{-1} A^\top + b^\top) v \\ \text{subject to} \quad & v \geq 0. \end{aligned} \quad (12)$$

With these notations, we write the primal QP (8) as:

$$\begin{aligned} \text{minimize}_z \quad & \frac{1}{2} z^\top z - {g_i^{\text{pub}}}^\top z + \frac{1}{2} {g_i^{\text{pub}}}^\top g_i^{\text{pub}} \\ \text{subject to} \quad & -{g_i^{\text{loc}}}^\top z \leq 0. \end{aligned} \quad (13)$$

According to the conversion formula above, We can pose the dual of the FedH2L QP as:

$$\begin{aligned} \text{minimize}_v \quad & \frac{1}{2} v^\top {g_i^{\text{loc}}}^\top g_i^{\text{loc}} v + {g_i^{\text{pub}}}^\top g_i^{\text{loc}} v \\ \text{subject to} \quad & v \geq 0. \end{aligned} \quad (14)$$

After (14) is solved for $v^\star$ which is specifically a real number here, we reset the projected gradient as $\tilde{g}_i = v^\star g_i^{\text{loc}} + g_i^{\text{pub}}$, and use $\tilde{g}_i$ to update θ_i for the global mutual optimization.

D. Summary

Bringing all components together, we have the full algorithm in Algo. 1. To summarize, (1) in each domain/node we first perform a local update with g_i^{loc} using $\ell^{(\text{CE})}$ on the locally preserved data and then broadcast its teaching signal $[\mathbf{p}_i^{\text{pub(i)}}, Acc_i]$ on public data. (2) In the global mutual optimization, FedH2L introduces distillation mimicry loss $\ell^{(\text{KL})}$ in addition to the conventional $\ell^{(\text{CE})}$ in order for each node to learn from its peers' teaching signals. (3) To manage potential conflicting gradients across nodes between g_i^{loc} and g_i^{pub}, we calculate the projected gradient $\tilde{g}_i$ as the final global gradient to update each f_{θ_i}. This ensures that each node in the cohort

Algorithm 1 FedH2L

Input: N domains $\mathcal{D} = \{D_1, D_2, \ldots, D_N\}$, D_{pub}, $D_i = \{D_i^{\text{loc}}, D_i^{\text{val}}, D_i^{\text{test}}\}$. Initialized N networks $\{f_{\theta_1}, f_{\theta_2}, \ldots, f_{\theta_N}\}$, learning rate β, η.

Output: Optimized networks $\{f_{\theta_1}, f_{\theta_2}, \ldots, f_{\theta_N}\}$ **begin**

- **while** *not converge or reach max steps* **do**
  - **for** $i \in [1, 2, \cdots, N]$ **do**
    - Sample local batch d_i^{loc} and public batch d_i^{pub}
    - Compute $g_i^{\text{loc}} \leftarrow$ Eq. (2) using $\ell^{(\text{CE})}$
    - Update $\theta_i \leftarrow \theta_i - \beta \cdot g_i^{\text{loc}}$
    - Compute $\mathbf{p}_i^{\text{pub(i)}}$ and Acc_i on d_i^{pub}
    - Broadcast $[\mathbf{p}_i^{\text{pub(i)}}, Acc_i]$
  - **for** $i \in [1, 2, \cdots, N]$ **do**
    - **for** $j \in [1, 2, \cdots, N]$ & $j \neq i$ **do**
      - Compute $\mathbf{p}_i^{\text{pub(j)}}$ using f_{θ_i}
    - Compute $g_i^{\text{pub}} \leftarrow$ Eq. (6) using $\ell^{(\text{KL})} + \ell^{(\text{CE})}$
    - **if** *Eq. (7) is satisfied* **then**
      - $\tilde{g}_i \leftarrow g_i^{\text{pub}}$
    - **else**
      - $\tilde{g}_i \leftarrow project(g_i^{\text{pub}}, g_i^{\text{loc}})$
    - Update $\theta_i \leftarrow \theta_i - \eta \cdot \tilde{g}_i$

achieves both CDP and WDP, improving performance on its own data, as well as strengthening its model to perform well on the private statistically heterogeneous distributions held by other nodes. This is the first work to consider both model and statistical heterogeneity across nodes in FL.

IV. Experiments

We evaluate on digit classification (Rotated MNIST) and image recognition (PACS, Office-Home) tasks. These datasets contain multiple sub-domains with statistical shift. We use the distributed framework Ray [39] to implement distributed applications. We compare FedH2L to the alternatives:

- *Independent (IND):* Node only uses the data of its own domain for conventional training (SGD on CE).
- *Aggregation (AGG):* Node aggregates its private domain data and the whole shared public data for conventional training. AGG is usually a strong baseline to beat in multi-domain learning [40].
- *FedMD [7]:* A state of the art centralized approach to model-heterogenity in FL.
- *FedAvg [2]:* The classic FL method that uses a central server to aggregate gradients and distribute parameters.
- *FedProx [6]:* A FedAvg-based approach that provides local regularization of convergence guarantees for learning over statistical heterogeneity.

Metrics In our decentralized approach, each node has its own model, and our goal is all models should outperform that of a centralized competitor. So we report the average test performance across all nodes' models. Considering the statistical heterogeneity, we report the following three metrics, where F evaluates test accuracy.

Within-Domain Performance: $WDP_i = F_i(D_i^{\text{test}})$. WDP is the performance of f_{θ_i} on the node i's test data. Higher WDP values indicate the learning experience from other nodes improve the performance on the current node. This is not guaranteed by a simple FL algorithm as other nodes' gradients can potentially cause conflict or forgetting [17], [41]. FedH2L aims to improve WDP by projecting away conflicting gradients.

Cross-Domain Performance: $CDP_i = F_i(\sum_{n=1, n\neq i}^{N} D_n^{\text{test}})$. CDP is the performance of f_{θ_i} on all other nodes' test data. If FL nodes do not learn from their peers then CDP will be low due to statistical shift.

Average accuracy: $ACC_i = F_i(\sum_{n=1}^{N} D_n^{\text{test}})$. ACC is the all-domain performance of f_{θ_i} on all nodes' test data.

A. Evaluation on Rotated MNIST

Dataset and settings Rotated MNIST [18] contains different domains with each one corresponding to a degree of roll rotation in MNIST dataset. The basic view (M0) is formed by randomly choosing 100 images each of ten classes from MNIST dataset, and we create 3 rotating domains from M0 with 20° rotation each in clockwise direction, denoted M20, M40, M60.

We first experiment by easily deploying homogeneous networks (e.g. LeNet [42]). We train using AMSGrad [43] optimizer (lr=1e-3, weight decay=1e-4) for 10,000 rounds and set batch_size=32. We explore performance considering several factors: (1) α, the proportion of D_{pub} compared with all data ($\mathcal{D}$ and D_{pub}). Note that the performance of IND, FedAvg and FedProx is independent of α. In this experiment, α of these rotated data can be split as the public data. (2) In FedH2L, E is the ratio between global and local update rounds. Local optimization is carried out each round, and global optimization every E rounds. So when calculating the global update $\tilde{g}_i$, g_i^{loc} is actually $(g_{i\_E}^{\text{loc}} - g_{i\_0}^{\text{loc}})$ over E rounds. Here we set default $E = 1$, and then ablate the hyperparameter sensitivity on E. (3) We explore both homogeneous and heterogeneous architectures. Note that even in the homogeneous architecture case, decentralized FedH2L nodes have independent parameters.

Results Table II shows the results including varying α of FedH2L. We evaluate using the validation data every 50 rounds and keep the model with the maximal ACC for the final test on three metrics. Max value on each metric is bold. We draw the following conclusions: (1) FedH2L generally outperforms competitors for a range of α. (2) FedH2L generally performs better with increased public data proportion α. (3) FedH2L outperforms the AGG and IND baselines at every α operating point. (4) Compared to state of the art competitors, FedH2L outperforms FedMD at every operating point. The poor performance of FedMD compared to FedH2L and AGG shows that it is vulnerable to distribution shift between domains. The vanilla centralized FedAVG/FedProx require over 1000× the communication bandwidth of FedH2L, and we now restrict their bandwidth to match that used by

TABLE II
TEST RESULT (%) ON THREE METRICS ON ROTATED MNIST.

Method	M0-LeNet			M20-LeNet			M40-LeNet			M60-LeNet			Avg.		
	ACC	WDP	CDP	ACC	WDP	CDP	ACC	WDP	CDP	ACC	WDP	CDP	ACC	WDP	CDP
FedH2L (α=5%)	**86.17**	88.67	**85.33**	86.33	**93.33**	85.11	**87.50**	93.33	**85.78**	**87.17**	**96.00**	**84.22**	**86.79**	**92.83**	**85.11**
AGG (α=5%)	85.50	**92.67**	83.11	**87.50**	**93.33**	**85.56**	83.67	90.00	81.56	83.83	93.33	80.67	85.13	92.33	82.73
FedMD (α=5%)	84.17	87.33	83.11	85.33	91.33	83.33	86.67	**96.00**	83.56	84.17	91.33	81.78	85.09	91.50	82.95
FedH2L (α=10%)	90.17	**93.33**	89.11	**91.67**	**96.00**	**90.22**	86.50	90.67	**85.11**	**88.17**	**93.33**	**86.44**	**89.13**	**93.33**	**87.72**
AGG (α=10%)	86.50	90.00	85.33	87.17	92.67	85.33	**86.67**	**94.00**	84.22	80.67	91.33	77.11	85.25	92.00	83.00
FedMD (α=10%)	85.00	88.67	83.78	87.67	95.33	85.11	82.00	90.67	79.11	85.67	90.00	84.22	85.09	91.17	83.06
FedH2L (asynchronous)	**90.66**	**93.33**	**89.78**	90.00	94.00	88.67	85.50	90.67	83.78	86.67	90.67	85.33	88.21	92.17	86.89
FedH2L (α=15%)	**89.67**	91.33	89.11	**90.00**	92.67	**89.11**	**90.50**	**94.00**	**89.33**	**88.33**	**92.67**	86.89	**89.63**	**92.67**	**88.61**
AGG (α=15%)	87.83	**92.00**	86.44	89.67	92.10	88.44	87.83	**94.00**	85.78	86.00	91.33	84.22	87.83	92.36	86.22
FedMD (α=15%)	88.67	89.33	88.44	89.00	93.33	87.56	85.00	90.00	83.33	84.33	**92.67**	81.56	86.75	91.33	85.22
IND	66.39	91.33	58.08	78.11	**94.00**	72.82	72.39	93.11	65.48	56.89	91.78	45.48	68.45	92.56	60.47
FedAvg	86.50	77.33	**89.56**	86.50	86.67	86.44	86.50	92.67	84.44	86.50	89.33	85.56	86.50	86.50	86.50
FedProx	86.67	80.00	88.89	86.67	90.00	85.56	86.67	91.33	85.11	86.67	85.33	**87.11**	86.67	86.67	86.67

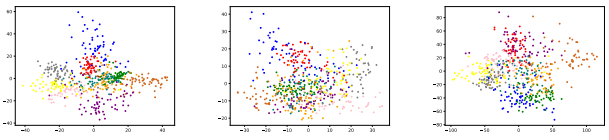

Fig. 1. PCA projections of features on all domains' test data using domain M0's model of Rotated MNIST for example. Left: FedH2L. Middle: IND. Right: AGG. (Dot: Image. Color: Digit label.)

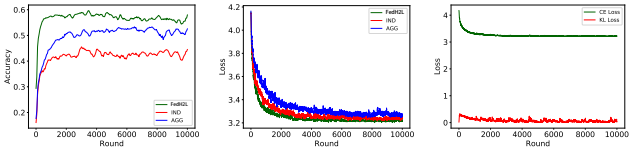

Fig. 2. Learning and loss curves on Office-Home in domain Product. Left: ACC on validation data. Middle: Loss in local optimization. Right: CE and KL losses in global optimization of FedH2L.

FedH2L and get the results in Table II. FedH2L outperforms FedAvg/FedProx clearly at $\alpha = 15\%$.

Qualitative Results We perform PCA projections of the features on all domains' test data in Figure 1. FedH2L provides the improved overall separability on all domains' data.

B. Evaluation on PACS dataset

Dataset and settings PACS [19] is a multi-domain object recognition benchmark with 9991 images of 7 categories across 4 different domains. The original PACS dataset has a fixed split for train, validation and test. We separate out 10% of its test part as the public seed data in our experiment, use the rest 90% of its test part as our test data, and directly use the train part as our private data. Here we mainly consider the heterogeneous model case where we deploy ResNet18, ResNet34, AlexNet and VGG11 in the experiment. The homogenous model case where all nodes use a ResNet18 is reported in the Further Studies, and it also shows the benefits of FedH2L. We use AMSGrad (lr=1e-4, weight decay=1e-5) to train 10,000 rounds and set batch_size=32.

Results We can see from Table III: (i) In the heterogeneous case, FedAvg and FedProx are inherently inapplicable and FedH2L surpasses the other alternatives. (ii) We observe that although VGG11 does not perform well in the sketch domain (see IND/AGG WDP), when used with FedH2L, it still benefits rather than harms the other nodes' performance thanks in part due to the teaching confidence signal (Eq. (3)).

C. Evaluation on Office-Home dataset

Dataset and settings The Office-Home [20] dataset is initially proposed to evaluate domain adaptation. It consists 4 different domains with each containing images of 65 object categories. We split each domains data into $\{D_i^{\text{pri}}, D_i^{\text{pub}}, D_i^{\text{val}}, D_i^{\text{test}}\}$ according to the default $[65\%, 10\%, 10\%, 15\%]$. We apply ResNet34, MobileNet, AlexNet and ResNet50 as their heterogeneous models and use the same hyperparameters as in the PACS experiment. The homogeneous model case is also reported in the Further Studies where FedH2L shows consistent benefits.

Results In Table IV, FedH2L gives a clear boost to overall accuracy, within-domain and cross-domain performance.

D. Further Analysis

Optimization and loss analysis Figure 2(left) shows ACC on the validation data. FedH2L exhibits faster convergence to the higher performance. Figure 2(right) shows the consistent utility of KL loss during the first 1000 rounds for convergence and performance benefits as shown on ACC. Figure 2(middle) shows the loss during local optimization, which benefits FedH2L locally with the help of the global mutual learning.

Discussion on design components of global mutual optimization In global optimization, our contributions are: KL mimicry loss Eq. (3), and the *project* operation for the calculation of $\tilde{g}_i$ to achieve stable multi-domain learning Eq. (8). We ablate them in Table V on Rotated MNIST ($\alpha = 10\%$).

TABLE III
TEST RESULT (%) ON THREE METRICS ON PACS WITH HETEROGENEOUS MODELS.

Method	Photo-ResNet18			Art_painting-ResNet34			Cartoon-AlexNet			Sketch-VGG11			Avg.		
	ACC	WDP	CDP	ACC	WDP	CDP	ACC	WDP	CDP	ACC	WDP	CDP	ACC	WDP	CDP
FedH2L	83.86	99.80	80.66	**90.91**	99.95	**88.57**	**81.68**	**99.67**	**76.16**	**52.87**	**80.33**	**37.26**	**77.33**	**94.94**	**70.66**
IND	51.08	99.57	41.29	77.72	99.30	72.15	68.52	99.39	59.05	44.79	78.75	22.83	60.53	94.25	48.83
AGG	**84.90**	**100.00**	**81.90**	89.50	**100.00**	86.85	80.80	98.77	75.28	52.81	78.01	36.51	77.00	94.20	70.14
FedMD	80.05	**100.00**	76.05	86.90	99.08	83.75	78.07	95.65	72.67	51.40	75.47	35.83	74.11	92.55	67.08
FedAvg/FedProx	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

TABLE IV
TEST RESULT (%) ON THREE METRICS ON OFFICE-HOME WITH HETEROGENEOUS MODELS.

Method	Art-ResNet34			Clipart-MobileNet			Product-AlexNet			Real_world-ResNet50			Avg.		
	ACC	WDP	CDP	ACC	WDP	CDP	ACC	WDP	CDP	ACC	WDP	CDP	ACC	WDP	CDP
FedH2L	**65.52**	**58.70**	**66.70**	**73.55**	76.52	**72.40**	**59.64**	**80.82**	**51.00**	**60.97**	**70.29**	**57.30**	**64.92**	**71.58**	**61.85**
IND	41.00	57.14	38.20	55.14	**78.49**	46.08	46.60	79.40	33.23	47.61	63.31	41.42	47.59	69.59	39.73
AGG	57.34	51.86	58.30	70.61	**78.49**	67.56	54.32	77.02	45.05	54.68	64.94	50.64	59.24	68.08	55.39
FedMD	55.46	55.59	55.44	67.49	77.50	63.61	53.17	75.59	44.02	51.74	59.42	48.72	56.97	67.03	52.95
FedAvg/FedProx	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

TABLE V
COMPONENTS STUDY IN GLOBAL MUTUAL OPTIMIZATION (AVG).

Method	ACC	WDP	CDP
FedH2L	**89.13**	**93.33**	**87.72**
FedH2L (no KL)	86.79	91.67	84.50
FedH2L (no $project$)	88.46	92.67	87.45
FedH2L (PCGrad)	88.34	92.67	86.89

TABLE VI
HYPERPARAMETER SENSITIVITY OF E IN FEDH2L (AVG).

Method	ACC	WDP	CDP
FedH2L ($E = 1$)	**89.13**	**93.33**	**87.72**
FedH2L ($E = 5$)	88.04	92.17	86.67
FedH2L ($E = 10$)	87.25	93.17	85.28

KL loss plays an important role in both CDP and WDP. The robustness benefit of mutual learning by KL loss to find a *wider* minimum in the single domain has been analyzed in DML [14]. Similarly, under our multi-domain setting, the matching with teachers' posterior predictions increases the model's generalization (CDP) to other domains. Meanwhile, the soft labels (for KL loss) help to alleviate the domain shift interference of the domain's hard true labels (for CE loss). Thus KL loss benefits optimization stability (WDP) during the global mutual optimization.

If we remove the $project$ operation, then θ_i will be updated by directly using g_i^{pub}. The results confirm that WDP gets worse without the constrained $\tilde{g}_i$. Moreover, we compare with an alternative gradient projection PCGrad [17] which deals with conflicting gradients in a handcrafted way. But PCGrad shows unsatisfactory performance even slightly worse than without the project operation.

Hyperparameter sensitivity We ablate the hyperparameter of E in FedH2L in Table VI on Rotated MNIST ($\alpha = 10\%$). FedH2L generally performs better with lower update interval E. Performance degrades smoothly with larger E which lowers communication cost proportionally.

Limitations A limitation of FedH2L is while our comms cost is $\approx 10e6\times$ lower than FedAvg at small scale (4 nodes), this advantage will be eroded if scaled to many participants. This could be alleviated by communicating between a subset of randomly chosen pairs at each global round, which preliminary experiments of such asynchronous distributed learning in Table II show lead to similar performance.

E. Further Studies

1) Results when using homogeneous models on PACS and Office-Home: We assemble the ResNet18 model for each node. For PACS, Table VII shows: FedH2L generally provides a consistent improvement over others in the homogeneous case. The original communication bandwidth of FedAvg/FedProx is $\approx 10e6\times$ that of ours. Even if we control FedAvg's and FedProx's communication to $\approx 100\times$ to ours by controlling E and its participating fraction [2], they are still outperformed by FedH2L. Similarly, for Office-home, FedH2L gives a clear boost to overall accuracy, backward and forward transfer performance in the homogeneous model case (see Table VIII).

2) Extension to the unlabeled public data: FedH2L can extend to the situation where the public data is available but unlabeled. The difference from the labeled public data case is: (i) only node's own data is available for local optimization without the use of public data from other domains; (ii) during global mutual optimization only KL loss is used to g_i^{pub} in Eq.(6) in the main paper. Table VII, VIII also report the FedH2L results with unlabeled public data. These results are still reasonable despite the absence of labels. Note that the AGG baseline is not applicable given that the public data from other domains is not available for supervised learning.

TABLE VII
TEST RESULT (%) ON THREE METRICS ON PACS WHEN USING HOMOGENEOUS MODELS (RESNET18).

Method	Photo-ResNet18			Art_painting-ResNet18			Cartoon-ResNet18			Sketch-ResNet18			Avg.		
	ACC	WDP	CDP	ACC	WDP	CDP	ACC	WDP	CDP	ACC	WDP	CDP	ACC	WDP	CDP
FedH2L(E=1)	84.31	99.93	81.22	**88.08**	99.89	85.04	87.10	99.57	83.27	**91.37**	99.58	**86.23**	**87.72**	99.74	83.94
FedH2L(E=5)	84.54	**100.00**	81.44	87.68	**100.00**	84.51	**87.20**	99.86	83.31	90.69	99.72	84.85	87.53	99.90	83.53
FedH2L(E=10)	83.88	**100.00**	80.64	87.53	**100.00**	84.31	86.20	99.62	82.08	90.38	99.69	84.36	87.00	99.83	82.85
IND	51.45	**100.00**	41.70	70.53	99.95	62.94	73.48	99.95	65.36	62.95	**99.89**	39.05	64.60	**99.95**	52.26
AGG	84.52	99.93	81.42	86.30	99.89	82.79	85.46	**100.00**	81.00	89.35	**99.89**	82.53	86.41	99.93	81.94
FedMD	82.39	99.87	78.88	85.75	99.62	82.17	83.93	99.91	79.03	88.52	98.56	82.02	85.15	99.49	80.53
FedAvg	84.93	95.62	82.78	84.93	72.06	**88.25**	84.93	72.23	**88.82**	84.93	94.68	78.62	84.93	83.65	84.62
FedProx	**85.80**	83.00	**86.36**	85.80	77.80	87.86	85.80	95.88	82.70	85.80	85.13	**86.23**	85.80	85.45	**85.79**
FedH2L(unlabeled)	81.13	99.47	77.45	85.60	99.89	81.91	82.68	99.39	77.55	87.38	98.45	80.23	84.20	99.30	79.29

TABLE VIII
TEST RESULT (%) ON THREE METRICS ON OFFICE-HOME WHEN USING HOMOGENEOUS MODELS (RESNET18).

Method	Art-ResNet18			Clipart-ResNet18			Product-ResNet18			Real_world-ResNet18			Avg.		
	ACC	WDP	CDP	ACC	WDP	CDP	ACC	WDP	CDP	ACC	WDP	CDP	ACC	WDP	CDP
FedH2L(E=1)	58.26	50.62	59.59	**61.25**	74.38	56.15	**62.21**	82.57	53.91	**62.12**	69.32	**59.28**	**60.96**	69.22	57.23
FedH2L(E=5)	56.89	49.38	58.19	59.32	70.61	54.94	59.46	78.76	51.58	59.00	66.88	55.89	58.67	66.41	55.15
FedH2L(E=10)	57.62	**53.73**	58.30	59.69	75.04	53.73	60.42	**84.00**	50.81	60.56	**69.81**	56.91	59.57	**70.65**	54.94
IND	35.58	48.13	33.41	44.90	**76.03**	32.83	47.29	80.03	33.94	51.92	68.34	45.45	44.92	68.13	36.41
AGG	54.41	50.00	55.17	57.58	72.91	51.63	60.15	82.88	50.87	58.45	68.18	54.61	57.65	68.49	53.07
FedMD	52.30	46.89	53.23	53.99	70.94	47.42	56.57	78.61	47.58	56.29	64.45	53.07	54.79	65.22	50.33
FedAvg	**59.96**	53.11	**61.15**	59.96	40.56	**67.50**	59.96	74.17	**54.17**	59.96	68.18	56.72	59.96	59.01	**59.89**
FedProx	53.90	44.10	55.60	53.90	46.80	56.66	53.90	64.66	49.52	53.90	55.03	53.46	53.90	52.65	53.81
FedH2L(unlabeled)	51.56	45.96	52.53	54.82	66.34	50.35	53.54	67.04	48.03	54.45	61.69	51.60	53.59	60.26	50.63

V. CONCLUSION

We proposed FedH2L for FL with heterogeneous models and data statistics. Each node in the cohort acts as both student and teacher, providing effective communication efficient federated learning. FedH2L supports heterogeneous architectures, which is crucial for FL across diverse hardware platforms, and with institutions' proprietary models; and is robust to heterogeneous data statistics, which – while not widely studied academically – is ubiquitous in practical FL.

VI. ACKNOWLEDGEMENT

This work was supported by the National Key R&D Program of China (Grant No. 2022ZD0115302) and National Natural Science Foundation of China (Grant No. 62206307). We thank Professor Timothy Hospedales from the University of Edinburgh for his constructive suggestions.

REFERENCES

[1] 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 *CCS*, 2017.

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

[3] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, 2016.

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

[5] A. Z. Tan, H. Yu, L. Cui, and Q. Yang, "Towards personalized federated learning," *TNNLS*, 2022.

[6] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *MLSys*, vol. 2, pp. 429–450, 2020.

[7] D. Li and J. Wang, "Fedmd: Heterogenous federated learning via model distillation," *arXiv preprint arXiv:1910.03581*, 2019.

[8] M. Mohri, G. Sivek, and A. T. Suresh, "Agnostic federated learning," in *ICML*. PMLR, 2019, pp. 4615–4625.

[9] M. G. Arivazhagan, V. Aggarwal, A. K. Singh, and S. Choudhary, "Federated learning with personalization layers," *arXiv preprint arXiv:1912.00818*, 2019.

[10] L. Collins, H. Hassani, A. Mokhtari, and S. Shakkottai, "Exploiting shared representations for personalized federated learning," in *ICML*. PMLR, 2021, pp. 2089–2099.

[11] Y. Tan, G. Long, L. Liu, T. Zhou, Q. Lu, J. Jiang, and C. Zhang, "Fedproto: Federated prototype learning across heterogeneous clients," in *AAAI*, vol. 36, no. 8, 2022, pp. 8432–8440.

[12] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," *NeurIPS*, vol. 32, 2019.

[13] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Inference attacks against collaborative learning," *arXiv preprint arXiv:1805.04049*, vol. 13, 2018.

[14] Y. Zhang, T. Xiang, T. M. Hospedales, and H. Lu, "Deep mutual learning," in *CVPR*, 2018, pp. 4320–4328.

[15] X. Peng, Z. Huang, Y. Zhu, and K. Saenko, "Federated adversarial domain adaptation," in *ICLR*, 2020.

[16] J. Quiñonero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, "Dataset shift in machine learning," in *The MIT Press*, 2009.

[17] T. Yu, S. Kumar, A. Gupta, S. Levine, K. Hausman, and C. Finn, "Gradient surgery for multi-task learning," *NeurIPS*, vol. 33, pp. 5824–5836, 2020.

[18] M. Ghifary, W. B. Kleijn, M. Zhang, and D. Balduzzi, "Domain generalization for object recognition with multi-task autoencoders," in *ICCV*, 2015, pp. 2551–2559.

[19] D. Li, Y. Yang, Y.-Z. Song, and T. M. Hospedales, "Deeper, broader and artier domain generalization," in *ICCV*, 2017, pp. 5542–5550.

[20] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, "Deep hashing network for unsupervised domain adaptation," in *CVPR*, 2017, pp. 5018–5027.

[21] T. Shen, J. Zhang, X. Jia, F. Zhang, G. Huang, P. Zhou, K. Kuang, F. Wu, and C. Wu, "Federated mutual learning," *arXiv preprint arXiv:2006.16765*, 2020.

[22] C. He, M. Annavaram, and S. Avestimehr, "Group knowledge transfer: Federated learning of large cnns at the edge," *NeurIPS*, vol. 33, pp. 14 068–14 080, 2020.

[23] G. Long, M. Xie, T. Shen, T. Zhou, X. Wang, and J. Jiang, "Multi-center federated learning: clients clustering for better personalization," *World Wide Web*, vol. 26, no. 1, pp. 481–500, 2023.

[24] F. Sattler, K.-R. Müller, and W. Samek, "Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints," *TNNLS*, vol. 32, no. 8, pp. 3710–3722, 2020.

[25] V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar, "Federated multi-task learning," *NeurIPS*, vol. 30, 2017.

[26] Y. Deng, M. M. Kamani, and M. Mahdavi, "Adaptive personalized federated learning," *arXiv preprint arXiv:2003.13461*, 2020.

[27] A. Fallah, A. Mokhtari, and A. Ozdaglar, "Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach," *NeurIPS*, vol. 33, pp. 3557–3568, 2020.

[28] V. Kulkarni, M. Kulkarni, and A. Pant, "Survey of personalization techniques for federated learning," in *WorldS4*, 2020, pp. 794–797.

[29] 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.

[30] O. Gupta and R. Raskar, "Distributed learning of deep neural network over multiple agents," *Journal of Network and Computer Applications*, vol. 116, pp. 1–8, 2018.

[31] J. Kang, Z. Xiong, D. Niyato, H. Yu, Y.-C. Liang, and D. I. Kim, "Incentive design for efficient federated learning in mobile networks: A contract theory approach," in *APWCS*. IEEE, 2019, pp. 1–5.

[32] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," in *ICC*, 2019.

[33] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *CVPR*, 2018, pp. 7482–7491.

[34] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[35] D. Lopez-Paz and M. Ranzato, "Gradient episodic memory for continual learning," *NeurIPS*, vol. 30, 2017.

[36] W. Zhou and Y. Li, "A fixed version of quadratic program in gradient episodic memory," *arXiv preprint arXiv:2107.07384*, 2021.

[37] J. Nocedal and S. J. Wright, *Numerical optimization*. Springer, 2006.

[38] R. I. Bot, S.-M. Grad, and G. Wanka, *Duality in vector optimization*. Springer Science & Business Media, 2009.

[39] P. Moritz, R. Nishihara, S. Wang, A. Tumanov, R. Liaw, E. Liang, M. Elibol, Z. Yang, W. Paul, M. I. Jordan *et al.*, "Ray: A distributed framework for emerging ai applications," in *OSDI*, 2018, pp. 561–577.

[40] Y. Li, Y. Yang, W. Zhou, and T. Hospedales, "Feature-critic networks for heterogeneous domain generalization," in *ICML*. PMLR, 2019, pp. 3915–3924.

[41] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," *Psychology of learning and motivation*, vol. 24, pp. 109–165, 1989.

[42] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[43] S. J. Reddi, S. Kale, and S. Kumar, "On the convergence of adam and beyond," in *ICLR*, 2018.