Multi-Source to Multi-Target Decentralized Federated Domain Adaptation

Su Wang, Seyyedali Hosseinalipour, *Member, IEEE,* Christopher G. Brinton, *Senior Member, IEEE*

***Abstract*—Heterogeneity across devices in federated learning (FL) typically refers to statistical (e.g., non-i.i.d. data distributions) and resource (e.g., communication bandwidth) dimensions. In this paper, we focus on another important dimension that has received less attention: varying quantities/distributions of *labeled* and *unlabeled* data across devices. In order to leverage all data, we develop a decentralized federated domain adaptation methodology which considers the transfer of ML models from devices with high quality labeled data (called sources) to devices with low quality or unlabeled data (called targets). Our methodology, Source-Target Determination and Link Formation (ST-LF), optimizes both (i) classification of devices into sources and targets and (ii) source-target link formation, in a manner that considers the trade-off between ML model accuracy and communication energy efficiency. To obtain a concrete objective function, we derive a measurable generalization error bound that accounts for estimates of source-target hypothesis deviations and divergences between data distributions. The resulting optimization problem is a mixed-integer signomial program, a class of NP-hard problems, for which we develop an algorithm based on successive convex approximations to solve it tractably. Subsequent numerical evaluations of ST-LF demonstrate that it improves classification accuracy and energy efficiency over state-of-the-art baselines.**

***Index Terms*—Federated learning, federated domain adaptation, link formation, decentralized federated learning, network optimization.**

I. Introduction

Federated learning (FL) [1], [2] is a class of distributed machine learning (ML) methods designed for training ML models across non-independent and identically distributed (non-i.i.d.) datasets at edge devices. FL has been actively studied in two different scenarios: (i) centralized FL [3], [4], which relies on a server to aggregate ML parameters across devices, and (ii) decentralized FL [5], [6], [7], in which devices exchange ML parameters through peer-to-peer communications. Existing work in both scenarios has made great strides in addressing the fact that devices often exhibit heterogeneity with respect to their local data distributions and/or local communication/computation capabilities for processing model updates. However, many of these works have implicitly assumed that each device has a sufficient amount of labeled data available to train local models in the first place [8], [9]. This tends to be unrealistic, particularly when the measurements are originating from unfamiliar environments.

In training supervised ML models through FL, the resulting global model will naturally be biased to favor devices/distributions with more labeled data. For example, the standard FL algorithm [3] and its weighted averaging rule will yield a global model that favors data categories that are labeled and more populous. Since labeled and unlabeled data are both distributed across real-world networks in a non-i.i.d. manner, the global model may not capture the unique statistical properties at devices with mostly unlabeled data, and thus lead to poor performance for those devices.

Consequently, there is a need for FL methodologies that address heterogeneity in the local quantity and distribution of unlabeled data. To address this, consider the possibility of using both labeled and unlabeled data at network devices to calculate statistical divergence measures between devices. In decentralized FL, we could then optimize the transfer of unique weighted combinations of ML models from (a) devices with high quality labeled data, called sources or source domains, to (b) devices with poorly labeled and/or unlabeled data, called targets or target domains. Such source-to-target model transmissions are commonly called domain adaptation [10], [11]. However, literature in this area [8], [12], [13] typically assumes that the sources and targets are known apriori. Since devices are heterogeneous with respect to both local *quantity* and *distribution* of unlabeled data in federated settings, optimal source-target determination is non-trivial, e.g., it could yield targets with some labeled data and sources with some unlabeled data. Moreover, empirical data divergence computations should not require the transmission of raw data.

In this paper, we investigate domain adaptation in decentralized FL, with the above considerations and additional network factors such as energy efficiency. Our proposed methodology, *Source-Target Determination and Link Formation* (ST-LF), is among the first to jointly (i) determine device source/target classifications, (ii) analyze the link formation problem from sources to targets, and (iii) minimize the total network communication resource consumption. Individually, (i), (ii), and (iii) pose several challenges: for example, optimal source/target classification may entail considering targets with labeled data. The difficulty is further exacerbated as (i), (ii), and (iii) are intertwined together, i.e., the result of source/target selection directly influences possible link formation regimes and communication resource consumption.

ST-LF aims to classify devices as either sources or targets by estimating post-training ML error based on the local quantity

S. Wang and C. G. Brinton are with Purdue University, IN, USA e-mail: {wang2506,cgb}@purdue.edu.

S. Hosseinalipour is with University at Buffalo–SUNY, NY, USA email: alipour@buffalo.edu.

This project was supported in part by the Office of Naval Research (ONR) under grants N000142212305 and N00014-23-C-1016, and by the National Science Foundation (NSF) under grants CNS-2146171 and CNS-2212565.

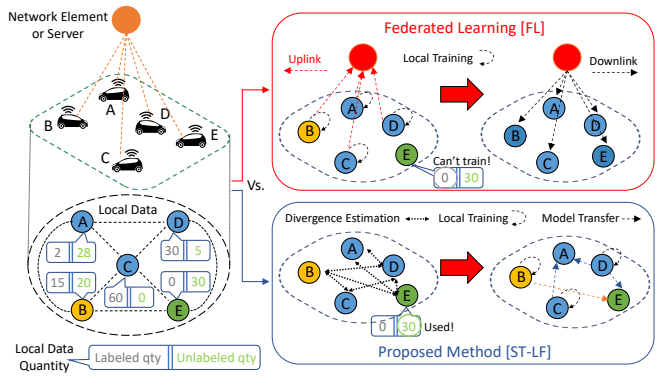


FIGURE 1: Small motivating example of a network of 5 smart cars. Only cars B, C, and D have meaningful amounts of labeled data, with cars A and E containing very few or no labeled data. Using a server, FL combines ML models from devices with labeled data, yielding a global model heavily biased for the "blue" domain. Meanwhile, ST-LF uses unlabeled data to estimate pair-wise divergences, then determines source/target selection and source-to-target link formation, leading to individualized ML models without a server.

and quality of data. To facilitate this, we develop theoretical bounds for expected post-training ML error and generalization bounds to capture the contribution of source-to-target model transmissions. We also develop an algorithm to compute empirical divergence on labeled and unlabeled data, which offers similar data privacy advantages found in FL by eliminating the need to transmit devices' raw data over the network. Then, we leverage theoretical bounds and divergence measures to optimize link formation from multiple sources to multiple targets. ST-LF for the first time reveals how communication efficiency requirements influence link formation from sources to targets, and optimizes network communication resource consumption jointly with link formation.

A. Motivating Toy Example

Consider Fig. 1, where a network of five smart cars is aiming to collaboratively construct object detection classifiers. Each car gathers data from its local environment, which may contain variations on the same type of object (e.g., images taken in rainy vs sunny environments) and unique sets of objects altogether (e.g., trees vs cacti in the landscape). We use the phrase "data domain" to describe this data-related heterogeneity across network devices, with a colored circle on the left hand side of Fig. 1 indicating similarity (in a statistical sense) between local data domains. For example, cars A, C, and D all have blue circles, meaning they share very similar data domains. Furthermore, each car has a unique local ratio of labeled and unlabeled data.

With traditional FL methods, only cars with labeled data will contribute to ML model training. As a result, cars A through D will train ML models while all cars, A through E, will receive the global ML model. Since there are more data and devices from the "blue" domain, the global ML model is biased to favor the "blue" domain, to the detriment of "yellow" (15 labeled data) and "green" (0 labeled data) domains.

With our proposed method (i.e., ST-LF), information in *both labeled and unlabeled* data is employed to estimate divergence among device pairs. Using this together with information on resource availability, ST-LF subsequently optimizes the set of source and target cars. In the example of Fig. 1, cars B, C, and D become the source cars, and cars A and E are the target cars. Car A's labeled dataset is small, but it has a similar "blue" data domain with cars C and D, which have much larger labeled datasets. Therefore, car A is likely well-represented by the local models constructed on similar datasets at cars C and D. These source cars will locally perform the ML training. Then, sources and targets will be matched together based on ST-LF's link formation output. In this example, the network can combine the ML models at cars C and D to yield an ML model for car A, as all three cars belong to the "blue" domain. Similarly, combining the ML models at cars B and D can yield a specially-tailored ML model for car E, as combining "yellow" and "blue" domains yields a "green" domain. Our optimization will further consider the tradeoff between wireless energy efficiency and ML quality improvement in determining link formation for the network.

B. Outline and Summary of Contributions

Structurally, we first review relevant work in Sec. II, and present some important theoretical preliminaries in Sec. III-B. Then, we develop our ST-LF methodology in Sec. IV, and experimentally characterize our formulation itself and its performance relative to several baselines both in Sec. V. We summarize our key contributions as follows:

- **Formulation of ST-LF** (Sec. IV): We study a novel yet natural problem in decentralized FL – only a subset of devices have high quality labeled data. We develop a methodology, ST-LF (overview in Fig. 2), to address source/target classification, link formation, and communication efficiency requirements simultaneously.
- **Development of measurable theoretical bounds** (Sec. IV-A): We develop a measurable multi-source generalization error bound that captures the impact of combining source hypotheses at targets, which we use to formulate model transfer between multiple sources to multiple targets. Subsequently, we propose an algorithm to estimate source-to-target distribution divergences based on hypothesis comparisons, which offers similar data privacy advantages to those found in FL by eliminating the need to transmit devices' raw data over the network.
- **Development of optimization methodology for ST-LF** (Sec. IV-B): As a part of ST-LF, we formulate a novel optimization problem which jointly optimizes our generalization error bound, source-to-target model transfer, and communication efficiency. We show that source/target classification and source-to-target link formation in multi-source to multi-target model transfer scenarios can be classified as a mixed-integer signomial program, a class of NP-hard problems. We subsequently propose a tractable solution using posynomial approximation-based techniques. The proposed optimization transformation techniques, given their generality and versatility, have a broader range of applicability to federated domain adaptation problems.
- **Demonstrate ML performance and energy improvements from ST-LF** (Sec. V): We experimentally demonstrate the superior performance of ST-FL's source/target

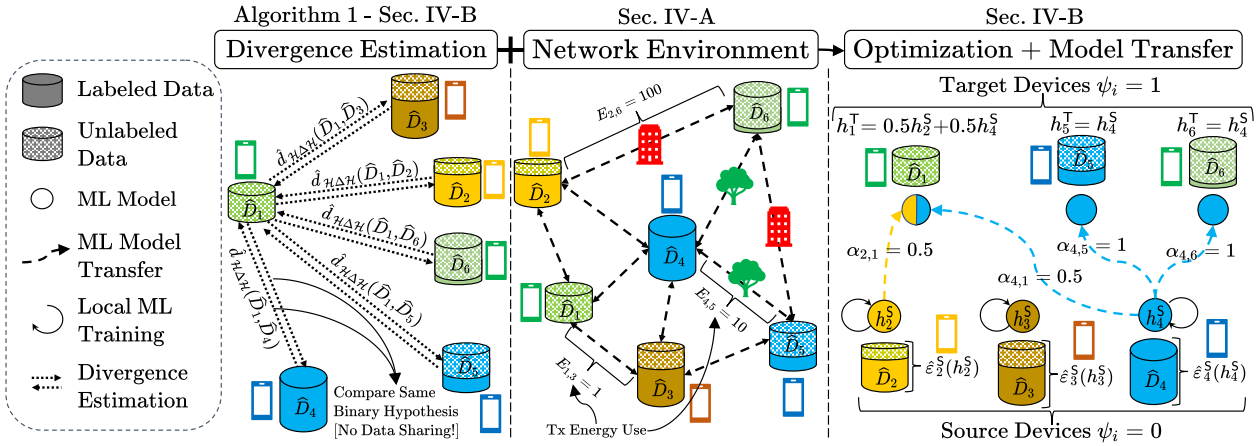


FIGURE 2: Overview of our ST-LF methodology, where each color represents a different domain. On the left, ST-LF first determines empirical distribution divergences among device pairs through comparison of a binary domain hypothesis/classifier (visualized for device 1). In the middle, ST-LF extracts information about the network environment such as communication energy costs indicted by "Tx Energy Use". Finally, on the right, ST-LF uses these measurements in an optimization problem to determine optimal source/target classification $\boldsymbol{\psi}$ and combination weights $\boldsymbol{\alpha}$ with respect to both expected ML model performance and network energy consumption.

classification and link formation relative to baselines from literature. Our experiments reveal the importance of jointly optimizing learning, model transfer, and communication.

II. Related Work

Federated Learning. Many methods in FL have been proposed to reduce the impact of non-i.i.d. gathered data on ML model training [14], [15]. In particular, multi-task FL [16] learns multiple models, personalized FL [17], [18], [19] adjusts a global model to fit individual devices, and works such as [7], [20] propose reducing statistical divergence by sharing data via social agreements. However, these works rely on all devices having fully labeled data, which poses difficulties in applications where networks of devices only have partially labeled data. In the extreme case of fully unlabeled data, methods for unsupervised FL have been proposed, e.g., [21] using surrogate labels or [22] which proposes federated contrastive learning. We study the "intermediate" problem where devices have varying levels of labeled data in FL, and develop novel methods to optimize the transfer of models trained at devices with high quality labeled data, called sources, to those devices without, called targets.

In our ST-LF methodology, the weighted contributions of source domain ML models are combined at the target domains. This makes our methodology a form of decentralized FL [5], [6], which exploits the proliferation of device-to-device wireless communications at the edge [23], [24]. In this regard, existing works have considered jointly optimizing expected ML model performance and the communication resource consumption that is required over these distributed links [25], [26], [27], [28]. However, these works address centralized settings where the objective is to minimize server-to-device communications. We instead consider how expected communication costs among D2D links can influence both source/target device selection and subsequent source-to-target selection in decentralized networks with unlabeled data.

Domain Adaptation. Domain adaptation [29], [30], [31] is concerned with the transferal of an ML model from a source to a target domain. Our problem is a form of unsupervised domain adaptation [32], [33] where the target domain is entirely reliant on the source domain. Whereas most unsupervised domain adaptation methods [10], [32], [34] assume the labeled and unlabeled data are in a single location, we consider the federated setting where data is distributed across different devices. To this end, [8] develops an adversarial-based method for distributed domain adaption where a single target and a set of sources is predetermined. In particular, the adversarial learning approach of [8] has inspired several extensions [35], [36] all involving generative adversarial networks and other adversarial methods to perform federated domain adaptation. Another recent method [37] entrusts a central server with data from the edge devices in a federated network, which, while effective, has some difficulty lending itself to decentralized FL settings and does lead to some minor data privacy concerns. Existing literature [8], [37], [34], [35] generally assumes that the source and target devices are known in the network apriori, and has yet to consider the source/target device classification problem by itself when devices may vary with respect to their local ratio and quantity of labeled/unlabeled data. Our methodology, enabled by our measurable multi-source generalization error bound, is the first to jointly consider source/target determination, link formation optimization, and communication efficiency for multi-source to multi-target domain adaptation.

III. Preliminaries

A. Multi-Source/Multi-Target Federated Domain Adaptation

We consider a set $\mathcal{N}$ of devices in a decentralized FL setting (see Fig. 2). We aim to partition $\mathcal{N}$ into *source* devices $\mathcal{S}$ and *target* devices $\mathcal{T}$. Each source $s \in \mathcal{S}$ trains a hypothesis $h_s^{\mathsf{S}} : \mathcal{X} \to \mathcal{Y}$, $h_s^{\mathsf{S}} \in \mathcal{H}$, where $\mathcal{H}$ denotes the hypotheses space, $\mathcal{X}$ is the input space of the data, and $\mathcal{Y}$ is the classification output. We consider $\mathcal{H}$ as a binary hypothesis space for our theoretical analysis, i.e., $\mathcal{Y} \in \{0, 1\}$, as in [10], [34]. Instead of locally training, each target device $t \in \mathcal{T}$ has a hypothesis h_t^{T} formed based on a weighted combination of source hypotheses, i.e.,

$h_t^{\mathsf{T}} = \sum_{s\in\mathcal{S}} \alpha_{s,t} h_s^{\mathsf{S}}$, where $\alpha_{s,t} \geq 0$ is the combination weight from source s to target t, and $\sum_{s\in\mathcal{S}} \alpha_{s,t} = 1$.

We define the *domain* of source s as $\boldsymbol{\mathcal{D}}_s^{\mathsf{S}} = \langle \mathcal{D}_s^{\mathsf{S}}, f_s^{\mathsf{S}} \rangle$ and target t as $\boldsymbol{\mathcal{D}}_t^{\mathsf{T}} = \langle \mathcal{D}_t^{\mathsf{T}}, f_t^{\mathsf{T}} \rangle$, where $\mathcal{D}_s^{\mathsf{S}}$, $\mathcal{D}_t^{\mathsf{T}}$ are data distributions and f_s^{S}, $f_t^{\mathsf{T}} : \mathcal{X} \to \mathcal{Y}$ are ground-truth labeling functions at s and t, respectively. In general, we expect the $\mathcal{D}_s^{\mathsf{S}}$, $\mathcal{D}_t^{\mathsf{T}}$ to be non-i.i.d. We denote the true error induced by a source hypothesis as

$$\varepsilon_s^{\mathsf{S}}(h_s^{\mathsf{S}}) = \mathbb{E}_{x\sim\mathcal{D}_s^{\mathsf{S}}}[|h_s^{\mathsf{S}}(x) - f_s^{\mathsf{S}}(x)|], \tag{1}$$

which is the expected deviation of source hypothesis h_s^{S} from f_s^{S} over $\mathcal{D}_s^{\mathsf{S}}$. Similarly, target devices $t \in \mathcal{T}$ have true error:

$$\varepsilon_t^{\mathsf{T}}(h_t^{\mathsf{T}}) = \mathbb{E}_{x\sim\mathcal{D}_t^{\mathsf{T}}}[|h_t^{\mathsf{T}}(x) - f_t^{\mathsf{T}}(x)|]. \tag{2}$$

We also use $\varepsilon_s^{\mathsf{S}}(\cdot,\cdot)$ and $\varepsilon_t^{\mathsf{T}}(\cdot,\cdot)$ to compare two hypotheses, e.g., $\varepsilon_s^{\mathsf{S}}(h_1, h_2)$ denotes the true hypothesis divergence error for $h_1, h_2 \in \mathcal{H}$ over $\mathcal{D}_s^{\mathsf{S}}$.

Multi-source to multi-target federated domain adaptation is the process of transferring hypotheses trained on source domains to target domains to minimize the target domains' true errors. However, the true error measures $\varepsilon_s^{\mathsf{S}}(\cdot), \varepsilon_t^{\mathsf{T}}(\cdot)$ depend on the underlying data distributions, which are unknown. We thus use the source empirical error:

$$\widehat{\varepsilon}_s^{\mathsf{S}}(h_s^{\mathsf{S}}) = \sum_{x\in\widehat{\mathcal{D}}_s^{\mathsf{S}}} \frac{|h_s^{\mathsf{S}}(x) - f_s^{\mathsf{S}}(x)|}{\widehat{D}_s^{\mathsf{S}}} \tag{3}$$

for sources $s \in \mathcal{S}$, where $\widehat{\mathcal{D}}_s^{\mathsf{S}}$ is the empirical dataset at s. In general, sets will be denoted with caligraphic font, e.g., $\mathcal{X}$, and non-caligraphic gives their cardinality, e.g., $X = |\mathcal{X}|$. The $\wedge$ symbol above a variable denotes an empirical result (i.e., the quantity is evaluated over an empirical dataset). While the exact ground-truth labeling function f_s^{S} for $s \in \mathcal{S}$ is unknown, we can determine $f_s^{\mathsf{S}}(x)$ when x is some labeled datum. When x is unlabeled, we treat $|h_s^{\mathsf{S}}(x) - f_s^{\mathsf{S}}(x)|$ as 1.[1] In this manner, the empirical error at a partially labeled device s can be adjusted to also consider the unlabeled data at s. While target devices $t \in \mathcal{T}$ may also have partially labeled datasets, for mathematical analysis, once classified as targets, target devices are assumed to have no labeled data. Consequently, their empirical target error $\widehat{\varepsilon}_t^{\mathsf{T}}(h_t^{\mathsf{T}})$ cannot be measured as $f_t^{\mathsf{T}}(x)$ is unknown $\forall x$. Instead, empirical hypothesis difference errors are considered at the targets, which we define formally for each target t as:

$$\widehat{\varepsilon}_t^{\mathsf{T}}(h_1, h_2) = \sum_{x\in\widehat{\mathcal{D}}_t^{\mathsf{T}}} \frac{|h_1(x) - h_2(x)|}{\widehat{D}_t^{T}}, \tag{4}$$

which can be computed given hypotheses $h_1, h_2 \in \mathcal{H}$. We can similarly calculate $\widehat{\varepsilon}_s^{\mathsf{S}}(h_1, h_2)$ at sources $s \in \mathcal{S}$.

In our problem setting, devices are heterogeneous with respect to the fractions of labeled data that they possess. Classifying devices with limited quantities of labeled data as sources could produce hypotheses that generalize poorly to local unlabeled data as well as to any potential target domains. We thus desire a new analytical framework for classifying devices in multi-source to multi-target federated domain adaptation, which has to-date remained elusive in the literature. To this end, we first review some relevant theory for single-source to single-target domain adaptation in the next section. We then develop our framework in Sec. IV-A.

[1]While the expected error on an unlabeled datum without prior information would be 0.5, this may not hold as an upper bound for empirical source errors.

B. Theoretical Background

Works in single-source to single-target domain adaptation [10], [34] have proposed analyzing the generalization error of a source hypothesis to a target domain in terms of a *domain divergence* measure:

Definition 1. *($\mathcal{H}$-divergence) Let $\mathcal{H}$ be the hypothesis class on input space $\mathcal{X}$, and $\mathcal{A}_{\mathcal{H}}$ denote the collection of subsets of $\mathcal{X}$ that form the support of a particular hypothesis in $\mathcal{H}$. The $\mathcal{H}$-divergence between two data distributions $\mathcal{D}$ and $\mathcal{D}'$ is defined as $d_{\mathcal{H}}(\mathcal{D}, \mathcal{D}') = 2\sup_{A\in\mathcal{A}_{\mathcal{H}}} |Pr_{\mathcal{D}}(A) - Pr_{\mathcal{D}'}(A)|$, where* sup *is the supremum over the subsets and $Pr_{\mathcal{D}}(A)$ is the probability that subset A is in $\mathcal{D}$.*

A hypothesis class $\mathcal{H}$ induces its own symmetric difference hypothesis space $\mathcal{H}\Delta\mathcal{H} := \{h(x) \oplus h'(x) | h, h' \in \mathcal{H}\}$, where $\oplus$ is the XOR operation. The $\mathcal{H}\Delta\mathcal{H}$ space has an associated divergence $d_{\mathcal{H}\Delta\mathcal{H}}$ defined as in Definition 1 but with $\mathcal{A} \in \mathcal{A}_{\mathcal{H}\Delta\mathcal{H}}$ instead. Denoting the optimal joint hypothesis for a single source s and target t as $h_{s,t}^* := \arg\min_{h\in\mathcal{H}}\{\varepsilon_s^{\mathsf{S}}(h) + \varepsilon_t^{\mathsf{T}}(h)\}$ and the corresponding minimum joint error as $\lambda_{s,t}^* := \varepsilon_s^{\mathsf{S}}(h^*) + \varepsilon_t^{\mathsf{T}}(h^*)$, [10] proved a generalization bound on the target error $\varepsilon_t^{\mathsf{T}}(h)$ in terms of the empirical source error and the empirical $\widehat{d}_{\mathcal{H}\Delta\mathcal{H}}$ divergence:

Theorem 1. *Let $\mathcal{H}$ be a hypothesis space of Vapnik–Chervonenkis (VC) dimension d, s be a source domain, t be a target domain, and $\widehat{\mathcal{D}}_s^{\mathsf{S}}$, $\widehat{\mathcal{D}}_t^{\mathsf{T}}$ be the empirical distributions induced by samples of size n drawn from $\mathcal{D}_s^{\mathsf{S}}$ and $\mathcal{D}_t^{\mathsf{T}}$. Then, with probability of at least $1-\delta$ over the choice of samples,*

$$\begin{aligned}\varepsilon_t^{\mathsf{T}}(h) \leq \widehat{\varepsilon}_s^{\mathsf{S}}(h) + \frac{\widehat{d}_{\mathcal{H}\Delta\mathcal{H}}(\widehat{\mathcal{D}}_s^{\mathsf{S}}, \widehat{\mathcal{D}}_t^{\mathsf{T}})}{2} \\ + 4\sqrt{\frac{2d\log(2n) + \log(4/\delta)}{n}} + \lambda_{s,t}^*, \forall h \in \mathcal{H}.\end{aligned} \tag{5}$$

Existing literature [8], [12], which has extended Theorem 1 to multi-source domain adaptation, has relied on identifying an optimal hypothesis $h_{s,t}^*$ for every source $s \in \mathcal{S}$ to a single target t in order to compute a combined minimum error. However, finding $h_{s,t}^*$ and $\lambda_{s,t}^*$ requires optimizing the error over an entire hypothesis space $\mathcal{H}$ for both source and target domains, which is impractical in our setting given the inability to centralize data in many FL applications. Furthermore, while these bounds consider the worst-case where the errors across sources are independent, in practice they may exhibit dependencies which can be exploited to obtain tighter bounds. Motivated by this, in Sec. IV-A, we develop a multi-source generalization error bound which considers the *joint* effect of sources, eliminating the need for $\lambda_{s,t}^*$ terms from (5). Subsequently, we formulate our link formation optimization problem that jointly minimizes source and target errors.

IV. Source-Target Determination and Link Formation in FL (ST-LF)

A. Generalization Error Characterization

One of our central contributions is developing a multi-source generalization error bound for the true target error $\varepsilon_t^{\mathsf{T}}(h_t^{\mathsf{T}})$ that consists of optimizable/controllable terms. In the process to obtain such a bound, we first bound the true target error $\varepsilon_t^{\mathsf{T}}(h_t^{\mathsf{T}})$ as a combination of terms that compare the domain at target t to the domains at sources $s \in \mathcal{S}$. This makes the bound a function of controllable combination weights $\alpha_{s,t}$ and reveals their impact on the performance.

Theorem 2. *(Upper Bound for True Target Error) Given a set of sources $\mathcal{S}$ and a weighted target hypothesis $h_t^{\mathsf{T}} = \sum_{s\in\mathcal{S}} \alpha_{s,t} h_s^{\mathsf{S}}$, where $0 \leq \alpha_{s,t} \leq 1\ \forall s,t$, and $\sum_{s\in\mathcal{S}} \alpha_{s,t} = 1$, for a target $t \in \mathcal{T}$, the true target error is bounded as follows:*

$$\varepsilon_t^{\mathsf{T}}(h_t^{\mathsf{T}}) \leq \sum_{s\in\mathcal{S}} \alpha_{s,t} \Bigg[\underbrace{\varepsilon_s^{\mathsf{S}}(h_s^{\mathsf{S}})}_{(i)} + \underbrace{\varepsilon_t^{\mathsf{T}}(f_s^{\mathsf{S}})}_{(ii)} + \underbrace{\frac{d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_t^{\mathsf{T}}, \mathcal{D}_s^{\mathsf{S}})}{2}}_{(iii)} + \underbrace{\varepsilon_t^{\mathsf{T}}(h_t^{\mathsf{T}}, h_s^{\mathsf{S}})}_{(iv)} \Bigg]. \quad (6)$$

The proof of Theorem 2 is given in Appendix A; the proofs of all subsequent results are in their appropriate appendices.

Theorem 2 elucidates a set of theoretical criteria for efficient model transfer: (i) a source hypothesis should have small error $\varepsilon_s^{\mathsf{S}}(h_s^{\mathsf{S}})$, (ii) the ground-truth labeling functions at the source and target should be similar as measured by $\varepsilon_t^{\mathsf{T}}(f_s^{\mathsf{S}}, f_t^{\mathsf{T}})$, (iii) the underlying data distributions at the sources and target should be similar as measured by $\frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_t^{\mathsf{T}}, \mathcal{D}_s^{\mathsf{S}})$, and (iv) a source hypothesis should lead to a small hypothesis combination noise $\varepsilon_t^{\mathsf{T}}(h_t^{\mathsf{T}}, h_s^{\mathsf{S}})$. Given a target t, the bound in (6) favors a larger combination weight $\alpha_{s,t}$ between a source-target pair (s,t) (i.e., model transfer between the two) if the hypothesis/domain at a source s satisfies the four aforementioned criteria.

The bound in (6) contains terms defined based on the underlying data distributions, e.g., $\mathcal{D}_t^{\mathsf{T}}$, that cannot be measured in practice. We thus aim to transform terms (i), (iii), and (iv) in (6) to measurable quantities that can be computed in practical systems.[2] To this end, we exploit the empirical Rademacher complexity and transform *true* error measures (e.g., $\varepsilon_t^{\mathsf{T}}$) defined on unknown data distributions to *empirical* error measures (e.g., $\widehat{\varepsilon}_T^{\mathsf{T}}$), which are computable.

Definition 2. *(Empirical Rademacher Complexity). Let $\mathcal{H}$ be a family of functions mapping from input domain $\mathcal{X}$ to an interval $[a,b]$ and $\boldsymbol{Q} = \{x_i\}_{i=1}^n$ be a fixed sample of size n with elements in $\mathcal{X}$. Then, the empirical Rademacher complexity of $\mathcal{H}$ with respect to the sample $\boldsymbol{Q}$ is defined as [38]:*

$$Rad_{\boldsymbol{Q}}(\mathcal{H}) = \mathbb{E}_\sigma \left[\sup_{h\in\mathcal{H}} \frac{1}{n} \sum_{i=1}^n \sigma_i h(x_i) \right], \quad (7)$$

where $\boldsymbol{\sigma} = \{\sigma_i\}_{i=1}^n$ and σ_i are i.i.d. random variables taking values in $\{+1, -1\}$.

[2]Term (ii) in (6) is based on ground-truth labeling functions and thus is a constant.

Empirical Rademacher complexity is a measure of the complexity of a hypothesis space $\mathcal{H}$ given a set of data $\boldsymbol{Q}$. Utilizing this measure, we show that true hypothesis noise (i.e., term (iv) in (6)) can be approximated via empirical errors.

Lemma 1. *(Upper Bound for Hypothesis Combination Noise) Let $\mathcal{H}$ be a binary hypothesis space. For arbitrary $0 < \delta < 1$, the following bound holds $\forall h \in \mathcal{H}$ with probability of at least $1-\delta$:*

$$\varepsilon_t^{\mathsf{T}}(h_t^{\mathsf{T}}, h_s^{\mathsf{S}}) \leq \widehat{\varepsilon}_t^{\mathsf{T}}(h_t^{\mathsf{T}}, h_s^{\mathsf{S}}) + 4 Rad_{\widehat{\mathcal{D}}_t^{\mathsf{T}}}(\mathcal{H}) + 3\sqrt{\frac{\log(2/\delta)}{2\widehat{D}_t^{\mathsf{T}}}}. \quad (8)$$

Lemma 1 obtains the true hypothesis comparison error $\varepsilon_t^{\mathsf{T}}(h_t^{\mathsf{T}}, h_s^{\mathsf{S}})$ between hypotheses h_t^{T} and h_s^{S} with respect to the data distribution $\mathcal{D}_t^{\mathsf{T}}$ at target t. Similar to the above procedure, for term (i) in (6), we can obtain

$$\varepsilon_s^{\mathsf{S}}(h_s^{\mathsf{S}}) \leq \widehat{\varepsilon}_s^{\mathsf{S}}(h_s^{\mathsf{S}}) + 2 Rad_{\widehat{\mathcal{D}}_s^{\mathsf{S}}}(\mathcal{H}) + 3\sqrt{\log(2/\delta)/(2\widehat{D}_s^{S})} \quad (9)$$

which resembles the results in [12].

The terms in (8) are measurable quantities. Specifically, the empirical Rademacher complexity $Rad_{\widehat{\mathcal{D}}_t^{\mathsf{T}}}(\mathcal{H})$ can be bounded as $Rad_{\widehat{\mathcal{D}}_t^{\mathsf{T}}}(\mathcal{H}) \leq \sqrt{2\log(2)}$ via Massart's Lemma (Lemma 3 in Appendix D). This is a general bound for binary hypothesis spaces $\mathcal{H}$ as it accounts for the worst-case, in which $\mathcal{H}$ shatters $\widehat{\mathcal{D}}_t^{\mathsf{T}}$. We next revisit Theorem 2 to obtain a measurable bound on the true target error:

Corollary 1. *(Measurable Error Bound for True Target Error) Let $\mathcal{H}$ be a hypothesis class, and $\{\widehat{\mathcal{D}}_s^{\mathsf{S}}\}_{s\in\mathcal{S}}$ and $\widehat{\mathcal{D}}_t^{\mathsf{T}}$ be empirical distributions of sources and target domains. For arbitrary $0 < \delta < 1$, the following bound holds $\forall h \in \mathcal{H}$ with probability of at least $1-\delta$:*

$$\varepsilon_t^{\mathsf{T}}(h_t^{\mathsf{T}}) \leq \sum_{s\in\mathcal{S}} \alpha_{s,t} \Bigg[\underbrace{\widehat{\varepsilon}_s^{\mathsf{S}}(h_s^{\mathsf{S}}) + \frac{\widehat{d}_{\mathcal{H}\Delta\mathcal{H}}(\widehat{\mathcal{D}}_t^{\mathsf{T}}, \widehat{\mathcal{D}}_s^{\mathsf{S}})}{2} + \varepsilon_t^{\mathsf{T}}(f_s^{\mathsf{S}}) + \widehat{\varepsilon}_t^{\mathsf{T}}(h_t^{\mathsf{T}}, h_s^{\mathsf{S}})}_{(a)} \quad (10)$$

$$+ 10\sqrt{2\log(2)} + \underbrace{6\left(\left(\frac{\log(2/\delta)}{2\widehat{D}_t^{\mathsf{T}}}\right)^{\frac{1}{2}} + \left(\frac{\log(2/\delta)}{2\widehat{D}_s^{\mathsf{S}}}\right)^{\frac{1}{2}}\right)}_{(b)} \Bigg].$$

We group the terms in (10) into two categories. The first, (a), captures the impact of source hypotheses, data distributions, and ground-truth labeling functions (i.e., the empirical source error, the empirical divergence, the ground-truth labeling function difference, and the hypothesis combination noise). The second, (b), considers the impact of data quantity at the sources and targets on the error.

Unlike existing multi-source generalization error bounds [8], [12], our result in Corollary 1 does not require pairwise hypothesis optimization for $h_{s,t}^*$ to minimize pairwise error $\lambda_{s,t}^*$, which, as discussed in Sec. III-B, enables us to consider the joint impacts of sources $s \in \mathcal{S}$ used in model transfer to a target t. Using this important property, we next formulate the multi-source to multi-target federated domain adaptation problem as a link formation optimization.

B. ST-LF Optimization Formulation and Solver

ST-LF aims to transfer source-trained ML models to targets in order to maximize the ML performance across all devices while efficiently using network communication resources. To this end, we should first determine the source-target classification across devices $\boldsymbol{\psi} = \{\psi_i\}_{i\in\mathcal{N}}$, with $\psi_i = 0$ if device i is a source and $\psi_i = 1$ otherwise, since that is not known *a priori*. Then, the combination weights (interpreted as link/edge weights) between different source-target pairs $\boldsymbol{\alpha} = \{\alpha_{i,j}\}_{i,j\in\mathcal{N}}$ should be designed, as visualized in Fig. 2. Formally, we pose ST-LF as the following optimization problem:

$$(\boldsymbol{\mathcal{P}}): \ \underset{\boldsymbol{\alpha},\boldsymbol{\psi}}{\arg\min}\ \phi^{\mathsf{S}} \underbrace{\sum_{i\in\mathcal{N}}(1-\psi_i)S_i}_{(c)} + \phi^{\mathsf{T}} \underbrace{\sum_{j\in\mathcal{N}}\psi_j \sum_{i\in\mathcal{N}}(1-\psi_i)\alpha_{i,j}T_{i,j}}_{(d)} + \phi^{\mathsf{E}} \underbrace{\sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{N}} E_{i,j}(\alpha_{i,j})}_{(e)} \quad (11)$$

subject to

$$h_j^{\mathsf{T}} = \sum_{i\in\mathcal{N}} \alpha_{i,j}(1-\psi_i)\psi_j h_i^{\mathsf{S}}, \forall j \in \mathcal{N} \quad (12)$$

$$\sum_{i\in\mathcal{N}} \alpha_{i,j} = \psi_j, \forall j \in \mathcal{N} \quad (13)$$

$$E_{i,j}(\alpha_{i,j}) = K_{i,j}\frac{\alpha_{i,j}}{\alpha_{i,j}+\epsilon_E}, \forall i,j \in \mathcal{N} \quad (14)$$

$$0 \le \alpha_{i,j} \le 1, \forall i,j \in \mathcal{N} \quad (15)$$

$$\psi_i(t) \in \{0,1\}, \forall i,j \in \mathcal{N}. \quad (16)$$

Objective of $(\boldsymbol{\mathcal{P}})$. (11) captures the trade-off between source/target classification errors and communication energy consumption from model transfers. In particular, (c) captures the true source error obtained in Lemma 1, where

$$S_i = \widehat{\varepsilon}_i^{\mathsf{S}}(h_i^{\mathsf{S}}) + 2\sqrt{2\log(2)} + 3\sqrt{\log(2/\delta)/(2\widehat{D}_i^{\mathsf{S}})}. \quad (17)$$

Also, (d) captures the error bound at target domains deduced in Corollary 1 through

$$\begin{aligned} T_{i,j} &= \widehat{\varepsilon}_i^{\mathsf{S}}(h_i^{\mathsf{S}}) + 10\sqrt{2\log(2)} \\ &+ \varepsilon_j^{\mathsf{T}}(f_j^{\mathsf{T}}, f_i^{\mathsf{S}}) + \frac{1}{2}\widehat{d}_{\mathcal{H}\Delta\mathcal{H}}(\widehat{\mathcal{D}}_j^{\mathsf{T}}, \widehat{\mathcal{D}}_i^{\mathsf{S}}) + \widehat{\varepsilon}_j^{\mathsf{T}}(h_j^{\mathsf{T}}, h_i^{\mathsf{S}}) \\ &+ 6\left(\sqrt{\log(2/\delta)/(2\widehat{D}_i^{\mathsf{S}})} + \sqrt{\log(2/\delta)/(2\widehat{D}_j^{\mathsf{T}})}\right). \end{aligned} \quad (18)$$

Finally, (e) measures the communication energy consumption specified in (14). Minimizing (c) in $(\boldsymbol{\mathcal{P}})$ influences source/-target classification $\boldsymbol{\psi}$ as the optimization will aim to classify only those devices with high quality data as sources. Terms (d) and (e) primarily determine the combination weights $\boldsymbol{\alpha}$ because the true target errors in term (d) can be minimized by proper selection of $\boldsymbol{\alpha}$ to optimize target hypotheses h_j^{T}, and the communication energy cost in term (e) is only incurred when a source i transmits its hypothesis to a target j, i.e., only when $\alpha_{i,j} > 0$ due to (14). We can shift the relative importance of terms (c), (d), and (e) in (11) by adjusting the scaling coefficients $\phi^{\mathsf{S}}, \phi^{\mathsf{T}}, \phi^{\mathsf{E}} \ge 0$. In the extreme case, $\phi^{\mathsf{S}} = 0$ makes all devices sources ($\mathcal{S} = \mathcal{N}$), $\phi^{\mathsf{T}} = 0$ makes all devices targets ($\mathcal{T} = \mathcal{N}$), and $\phi^{\mathsf{E}} = 0$ means the optimization does not consider communication energy use.

$(\boldsymbol{\mathcal{P}})$ is the first concrete formulation of link formation in FL domain adaptation enabled via our results in Sec. IV-A. While communication energy is chosen as the cost of an active link between devices i and j (similarly to [39], [19]), the factor $K_{i,j}$ in (11) and (14) could be adjusted on a case-by-case basis to fit various other potential link costs of interest such as latency, privacy constraints, or link formation, as long as these costs do not introduce factors that violate signomial/geometric programming rules that we will employ later in this section. Different practical link formation costs can lead $(\boldsymbol{\mathcal{P}})$ to yield alternative model combinations.

More generally, we can draw parallels between $(\boldsymbol{\mathcal{P}})$ and link formation problems, which have been well studied in network science. This includes in wireless/IoT networks, where links are established for e.g., throughput maximization [40], and in social networks, where link costs correspond to e.g., privacy/trust measures [41]. However, existing link formation methodologies are not well-suited for the federated domain adaptation problem, presented in $(\boldsymbol{\mathcal{P}})$, which poses a tight coupling between ML performance and resource efficiency.

Constraints of $(\boldsymbol{\mathcal{P}})$. We replace the original definition of $h_j^{\mathsf{T}} = \sum_{i\in\mathcal{S}} \alpha_{i,j} h_i^{\mathsf{S}}$ (Sec. III-A) with (12) by adding the coefficient $(1-\psi_i)\psi_j$ inside the sum. This coefficient ensures that target hypotheses are only reliant on source hypotheses since $(1-\psi_i)\psi_j = 1$ if and only if $\psi_i = 0$ (i.e., i is a source) and $\psi_j = 1$ (i.e., j is a target). Constraint (13) guarantees that only target devices ($\psi_j = 1$) receive models (i.e., $\sum_{i\in\mathcal{N}} \alpha_{i,j} = 1$) and source devices ($\psi_j = 0$) only send models (i.e., $\sum_{i\in\mathcal{N}} \alpha_{i,j} = 0$). (14) models when communication energy is incurred between two devices i and j, i.e., only if model transfer occurs between them ($\alpha_{i,j} > 0$). In fact, $\alpha_{i,j}$ only has two states: either $\alpha_{i,j} > 0$ so source i transmits to target j or $\alpha_{i,j} = 0$, meaning there is no transmission. To capture this in a tractable manner, we approximate the desired behavior via $\frac{\alpha_{i,j}}{\alpha_{i,j}+\epsilon_E}$, where $0 < \epsilon_E \ll 1$, in (14), so that $\frac{\alpha_{i,j}}{\alpha_{i,j}+\epsilon_E} \approx 1$ when $\alpha_{i,j} > 0$, and $\frac{\alpha_{i,j}}{\alpha_{i,j}+\epsilon_E} = 0$ when $\alpha_{i,j} = 0$. $K_{i,j}$ is the energy/cost of model transfer from i to j, for which we adopt physical layer communication models presented in Sec. V. Thus, in our work, the energy consumption $K_{i,j}$ is taken as the cost of link formation. Finally, (15) and (16) are the feasibility constraints for offloading ratios and source/target classification respectively.

Computing the terms of $(\boldsymbol{\mathcal{P}})$. Before we can solve $(\boldsymbol{\mathcal{P}})$, the optimization terms must first be computed. In particular, the source error terms embedded within S_i (found in part (c) of (11)) can all be computed directly, for example, the empirical source errors, $\widehat{\varepsilon}_i^{\mathsf{S}}(h_i^{\mathsf{S}})$, can be locally determined by devices themselves. While the empirical error terms, $\widehat{\varepsilon}_j^{\mathsf{T}}$ and $\widehat{\varepsilon}_i^{\mathsf{S}}$, embedded within $T_{i,j}$ can be similarly computed, the remaining terms found in part (d) of (11) require different strategies. Firstly, the ground-truth labeling function difference, $\varepsilon_j^{\mathsf{T}}(f_j^{\mathsf{T}}, f_i^{\mathsf{S}})$ cannot be found in practice, as it requires devices to have concrete ground-truth labeling functions. If devices have concrete ground-truth labeling functions, then the network has no need for any ML, and, therefore, we can

Algorithm 1: Determination of Empirical Divergences

input : Starting Domain Classification Hypothesis $h^{'}$, Network Devices $\mathcal{N}$, Local Training Period T^d, Number of Local Aggregations τ^d

1 **for** *Device pair* (i,j), $i \in \mathcal{N}$ $j \in \mathcal{N}$, $i \neq j$ **do**
2 Initialize hypothesis at both devices i and j, i.e., $h_i = h_j = h^{'}$.
3 Label all $x_d \in \widehat{\mathcal{D}}_i$ as 0 and all $x_d \in \widehat{\mathcal{D}}_j$ as 1.
4 **for** $\tau_i = 1$ *to* τ^d **do**
5 Locally train hypotheses at i and j on local data $\widehat{\mathcal{D}}_i$ and $\widehat{\mathcal{D}}_j$ respectively for T^d iterations.
6 Transfer hypothesis h_i at device i to device j, and vice versa.
7 Take the average of h_i and h_j, i.e., $\bar{h}^{'} = \frac{h_i + h_j}{2}$.
8 Measure classification error of final hypothesis $\bar{h}^{'}$ on data at devices i and j.
9 Transfer error of final hypothesis at device i to device j, and vice versa.
10 Compute $\widehat{d}_{\mathcal{H}}(\widehat{\mathcal{D}}_j, \widehat{\mathcal{D}}_i)$ using the classification error.
11 Obtained $\widehat{d}_{\mathcal{H}}(\widehat{\mathcal{D}}_j, \widehat{\mathcal{D}}_i)$ for all potential source and target pair combinations, i.e., $\forall i \in \mathcal{N}$ and $\forall j \in \mathcal{N}$.

Algorithm 2: Optimization solver for problem $\boldsymbol{\mathcal{P}}$

input : Convergence criterion.
output : $\boldsymbol{x}^\star$, Objective of $\boldsymbol{\mathcal{P}}$ evaluated at $\boldsymbol{x}^\star$

1 Set the iteration count $\ell = 0$.
2 Choose a feasible point $\boldsymbol{x}^{[0]}$.
3 Obtain the monomial approximations (19),(20),(21),(22),(23),(24) given $\boldsymbol{x}^{[\ell]}$.
4 Replace the results in the approximation of Problem $\boldsymbol{\mathcal{P}}$ (see $\boldsymbol{\mathcal{P}}'$ in Appendix H-2).
5 With logarithmic change of variables, transform the resulting GP problem to a convex problem.
6 $\ell = \ell + 1$
7 Obtain the solution of the convex problem using current art solvers (e.g., CVXPY [42]) to determine $\boldsymbol{x}^{[\ell]}$.
8 **if** *two consecutive solutions* $\boldsymbol{x}^{[\ell-1]}$ *and* $\boldsymbol{x}^{[\ell]}$ *do not meet the specified convergence criterion* **then**
9 Go to line 3 and redo the steps using $\boldsymbol{x}^{[\ell]}$.
10 **else**
11 Set the solution of the iterations as $\boldsymbol{x}^\star = \boldsymbol{x}^{[\ell]}$.

omit the ground-truth labeling function differences from the computation of $T_{i,j}$.

Owing to practical difficulties in computing $\widehat{d}_{\mathcal{H}\Delta\mathcal{H}}$ generally, we follow a similar strategy as [10], [12] in using $\widehat{d}_{\mathcal{H}}$ as an estimate for $\widehat{d}_{\mathcal{H}\Delta\mathcal{H}}$. To compute the empirical $\frac{1}{2}\widehat{d}_{\mathcal{H}}(\widehat{\mathcal{D}}_j^{\mathsf{T}}, \widehat{\mathcal{D}}_i^{\mathsf{S}})$ divergence, we propose Algorithm 1, a decentralized peer-to-peer algorithm designed to operate pairwise between devices. Since empirical distribution divergences quantify the separability of two domains based on their local data [10], our algorithm uses a binary classifier at two devices to obtain $\frac{1}{2}\widehat{d}_{\mathcal{H}}(\widehat{\mathcal{D}}_j^{\mathsf{T}}, \widehat{\mathcal{D}}_i^{\mathsf{S}})$. Essentially, algorithm 1 has three key steps: (i) relabel data as 0 at the source and 1 at the target, (ii) separately train and combine binary classifiers at the source and target domains, and (iii) use the final binary classifier to find the empirical separability of the two domains. Since Algorithm 1 only relies on sharing the parameters of this binary domain classifier, it thus offers similar data privacy advantages found in FL by avoiding raw data sharing among devices. Further discussion of Algorithm 1 is provided in Appendix F.

Solution of $(\boldsymbol{\mathcal{P}})$. In $(\boldsymbol{\mathcal{P}})$, the optimization variables are tightly coupled together - as any choice of source/target classification ψ_i, $\forall i \in \mathcal{N}$, directly influences possible model offloading ratios $\alpha_{i,j}$, $\forall i,j \in \mathcal{N}$. This coupling manifests itself through the multiplication of optimization variables. For example, term (d) in (11) and (12) contains the multiplication of real variable $\alpha_{i,j}$ and integer variable ψ_j. Furthermore, these variable combinations often involve (i) negative optimization variables such as the $1-\psi_i$ term found in (d) and (12), (ii) equality constraints such as (13), and (iii) the division of sums involving optimization variables such as (14). As a result of these complex combinations of optimization variables, $(\boldsymbol{\mathcal{P}})$ belongs to a class of mixed-integer signomial programs, which are known to be NP-hard and non-convex [43]. Since this formulation captures the intricate and coupled process from source/target identification to model ratio offloading, this complexity is expected.

We now develop a tractable solution for $(\boldsymbol{\mathcal{P}})$ based on a set of modifications and approximations for the objective function (11) and constraints (12)-(16). These approximations can then be iteratively refined, ultimately converging to an optimal solution. While our approach in this work is refined specifically for $(\boldsymbol{\mathcal{P}})$, it can be applied for general link formation problems in decentralized federated domain adaptation, where formulations are concerned with optimizing ML performance through network control. We are among the first to leverage these flexible optimization methods for domain adaptation problems.

Our methodology exploits approximations of (11)-(16) to convert $(\boldsymbol{\mathcal{P}})$ from a mixed-integer signomial programming problem to a geometric programming problem, which, after a logarithmic change of variables, becomes a convex programming problem that can be solved using modern optimization libraries such as CVXPY [42]. Thus, to properly explain our methodology, we must discuss geometric programming (GP), which requires first defining monomials and posynomials.

Definition 3. *A **monomial** is defined as a function*[3] $f : \mathbb{R}^n_{++} \rightarrow \mathbb{R}$ *of the form* $f(\boldsymbol{y}) = z y_1^{\beta_1} y_2^{\beta_2} \cdots y_n^{\beta_n}$, *where* $z \geq 0$, $\boldsymbol{y} = [y_1, \cdots, y_n]$, *and* $\beta_j \in \mathbb{R}$, $\forall j$. *A **posynomial** g is defined as a sum of monomials, and has form* $g(\boldsymbol{y}) = \sum_{m=1}^{M} z_m y_1^{\beta_1^m} y_2^{\beta_2^m} \cdots y_n^{\beta_n^m}$, $z_m \geq 0, \forall m$.

A further discussion of GP is available in Appendix H-1, but the key point is that only standard GP with posynomial objective function subject to posynomial inequality and monomial equality constraints can enable a logarithmic change of variables and subsequent solution using modern solvers. Our formulation, $(\boldsymbol{\mathcal{P}})$, violates GP rules. Specifically, the objective function (11) is not a posynomial or a monomial due to the negative optimization variables (i.e., $(1-\psi_i)$) present in parts (c) and (d). Furthermore, both (13) and (14)

[3] $\mathbb{R}^n_{++}$ denotes the strictly positive quadrant of an n-dimensional Euclidean space.

violate GP constraints, as the former is an equality constraint on posynomials while the latter contains a division by a posynomial which is not a posynomial. To address all these violating terms, we use the method of penalty functions and auxiliary optimization variables [44] to approximate (i) the negative optimization variables, (ii) the posynomial divisors, and (iii) the posynomial equality constraints.

We first consider terms with negative optimization variables. First, we bound these terms using a unique auxiliary variable, yielding the general format: $a(x) - b(x) < \chi$, where $a(x)$ represents terms with positive variables, $b(x)$ represents terms with negative variables, and $\chi > 0$ is the auxiliary variable. Manipulating this expression yields $a(x)/(b(x) + \chi) \leq 1$, which is not a posynomial due to the division by a posynomial. One way to proceed is to approximate the posynomial denominator using a monomial, which allows the expression to become a posynomial (as the division of a posynomial by a monomial is a posynomial). To do so, we can bound a posynomial with a monomial through the arithmetic-geometric mean inequality:

Lemma 2 (Arithmetic-geometric mean inequality [45]). *Consider a posynomial function $g(\boldsymbol{y}) = \sum_{i=1}^{i'} u_i(\boldsymbol{y})$, where $u_i(\boldsymbol{y})$ is a monomial, $\forall i$. The following inequality holds:*

$$g(\boldsymbol{y}) \geq \widehat{g}(\boldsymbol{y}) \triangleq \prod_{i=1}^{i'} \left(\frac{u_i(\boldsymbol{y})}{\alpha_i(\boldsymbol{z})} \right)^{\alpha_i(\boldsymbol{z})}, \tag{25}$$

where $\alpha_i(\boldsymbol{z}) = u_i(\boldsymbol{z})/g(\boldsymbol{z})$, $\forall i$, and $\boldsymbol{z} > 0$ is a fixed point.

The arithmetic-geometric mean inequality yields an initial, monomial bound that can be rather loose, but can be iteratively refined to obtain near equality with the original posynomial. We repeat this posynomial-to-monomial approximation process for terms (c) and (d) from (11), as they contain negative optimization variables. Additionally, $T_{i,j}$ contains h_j^{T}, which contains another negative $(1-\psi_j)$ term, so we use an approximation for $T_{i,j}$ within our approximation for (d) from (11). As (14) also has a posynomial denominator, we apply the same strategy to approximate its posynominal denominator as a monomial. Finally, we can substitute the equality constraint in (13) into two constraints (i) $\sum_{i\in\mathcal{N}} \alpha_{i,j} - \psi_j \leq \chi_{i,j}^{C} + \epsilon_C$ and (ii) $\sum_{i\in\mathcal{N}} \alpha_{i,j} - \psi_j \geq \chi_{i,j}^{C} - \epsilon_C$, with auxiliary variable $\chi_{i,j}^{\mathsf{C}}$ and a very small constant $\epsilon_C > 0$. In this way, the objective function augmented with $\chi_{i,j}^{\mathsf{C}}$ will squeeze (i) and (ii), approximating equality. Both constraints (i) and (ii) involve negative optimization variables, allowing us to then approximate them using the above outlined process for (11).

To summarize, the approximations for (11)-(14) are displayed in (19)-(24). The above discussion has been a high-level overview of the derivations of our approximations. The full derivations for each approximation are rather lengthy, and are thus deferred to Appendix H-2 for conciseness. To ensure that these approximations converge independently, they are associated with a unique auxiliary variable that is added to the objective function or bounded between very small quantities using an additional pair of constraints. The resulting optimization formulation with modified objective function, approximations (19)-(24), augmented constraints, and auxiliary variables retains the same core insights as those for $(\boldsymbol{\mathcal{P}})$, and is thus similarly left to Appendix H-2 to minimize repetition.

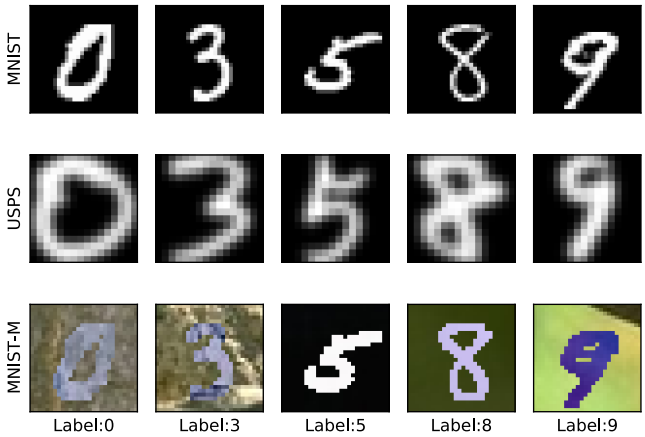


FIGURE 3: An overview of the three common domain adaptation datasets used to evaluate our method. We explain the physical differences of each dataset in Sec. V.

As a result of the approximations in (19)-(24), our optimization solver, summarized in Algorithm 2, is an iterative method that starts with an initial value for the solution $[\boldsymbol{x}]^0$, containing an initial estimate for $\boldsymbol{\alpha}$ and $\boldsymbol{\psi}$. The only requirement for the initial estimate is that it is feasible for $\boldsymbol{\alpha}$ and $\boldsymbol{\psi}$ - our solver will then converge to the optimal regardless of initial estimate. At each iteration indexed via ℓ, our solver obtains a solution $[\boldsymbol{x}]^\ell$ via transforming the problem into a solvable convex program, in which all the terms in the objective function and constraints are transformed into convex terms around the previous $[\boldsymbol{x}]^{\ell-1}$.

V. Numerical Evaluation

In this section, we experimentally demonstrate four key points of ST-LF. In Sec. V-A, we show that our optimization $(\boldsymbol{\mathcal{P}})$ (i) allocates source/target classification $\boldsymbol{\psi}$ and link formation $\boldsymbol{\alpha}$ effectively based on distribution divergence $\widehat{d}_{\mathcal{H}\Delta\mathcal{H}}$, and (ii) adjusts communication resource consumption as a result of both the energy scaling ϕ^{E} and the underlying classification task. Then, in Sec. V-B, we show that our methodology (iii) obtains significant enhancements in classification accuracy at target devices for domain adaptation tasks and (iv) significant communication resource savings compared to several baseline algorithms, validating its benefit for decentralized FL settings.

Experimental Setup. We conduct our evaluations on three image classification datasets commonly used in domain adaptation: MNIST [46], USPS [47], and MNIST-M [32]. A snapshot of the three datasets is shown in 3. In particular, while all three datasets are concerned with the problem of digit recognition, the images in each dataset are formatted differently, and, thus, transferring a trained ML model from one dataset to another encounters some difficulty. Specifically, the images in MNIST are neatly formatted with the same background, and USPS contains images similar to MNIST but taken with different resolution. Finally, as a tougher challenge, MNIST-M contains digits from MNIST blended with various photographs from BSDS500 [48], resulting in a wide-range of digit outlines and backgrounds (e.g., the sample MNIST-M

$$F_i(\boldsymbol{x}) = \psi_i + \frac{\chi_i^{\mathsf{S}}}{S_i} \geq \widehat{F}_i(\boldsymbol{x};\ell) \triangleq \left(\frac{\psi_i F_i([\boldsymbol{x}]^{\ell-1})}{[\psi_i]^{\ell-1}}\right)^{\frac{[\psi_i]^{\ell-1}}{F_i([\boldsymbol{x}]^{\ell-1})}} \left(\frac{\chi_i^{\mathsf{S}} F_i([\boldsymbol{x}]^{\ell-1})}{[\chi_i^{\mathsf{S}}]^{\ell-1}}\right)^{\frac{[\chi_i^{\mathsf{S}}/S_i]^{\ell-1}}{F_i([\boldsymbol{x}]^{\ell-1})}} \tag{19}$$

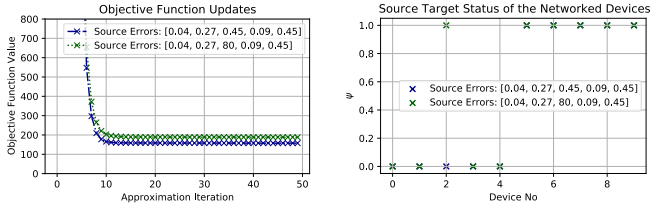
$$G_{i,j,k}(\boldsymbol{x}) = \psi_k + \frac{\widehat{\chi}^{\mathsf{T}}_{i,j,k}}{\psi_i \alpha_{k,i} \widehat{T}_{i,j,k}} \geq \widehat{G}_{i,j,k}(\boldsymbol{x};\ell) \triangleq \left(\frac{\psi_k G_{i,j,k}([\boldsymbol{x}]^{\ell-1})}{[\psi_k]^{\ell-1}}\right)^{\frac{[\psi_k]^{\ell-1}}{G_{i,j,k}([\boldsymbol{x}]^{\ell-1})}} \left(\frac{G_{i,j,k}([\boldsymbol{x}]^{\ell-1}) \widehat{\chi}^{\mathsf{T}}_{i,j,k}/(\psi_i \alpha_{k,i})}{\left[\widehat{\chi}^{\mathsf{T}}_{i,j,k}/(\psi_i \alpha_{k,i})\right]^{\ell-1}}\right)^{\frac{\left[\widehat{\chi}^{\mathsf{T}}_{i,j,k}/(\widehat{T}_{i,j,k}\psi_i\alpha_{k,i})\right]^{\ell-1}}{G_{i,j,k}([\boldsymbol{x}]^{\ell-1})}} \tag{20}$$

$$H_{i,j}(\boldsymbol{x}) = \psi_i \widehat{T}_{i,j} + \psi_i \sum_{k\in\mathcal{N}} \widehat{\chi}^{\mathsf{T}}_{i,j,k} + \frac{\chi^{\mathsf{T}}_{i,j}}{\psi_j \alpha_{i,j}} \geq \widehat{H}_{i,j}(\boldsymbol{x};\ell) \triangleq$$
$$\left(\frac{\psi_i H_{i,j}([\boldsymbol{x}]^{\ell-1})}{[\psi_i]^{\ell-1}}\right)^{\frac{[\psi_i]^{\ell-1}\widehat{T}_{i,j}}{H_{i,j}([\boldsymbol{x}]^{\ell-1})}} \left(\frac{H_{i,j}([\boldsymbol{x}]^{\ell-1})\chi^{\mathsf{T}}_{i,j}/(\psi_j\alpha_{i,j})}{\left[\chi^{\mathsf{T}}_{i,j}/(\psi_j\alpha_{i,j})\right]^{\ell-1}}\right)^{\frac{\left[\chi^{\mathsf{T}}_{i,j}/(\psi_j\alpha_{i,j})\right]^{\ell-1}}{H_{i,j}([\boldsymbol{x}]^{\ell-1})}} \prod_{k\in\mathcal{N}} \left(\frac{\psi_i \widehat{\chi}^{\mathsf{T}}_{i,j,k} H_{i,j}([\boldsymbol{x}]^{\ell-1})}{[\psi_i \widehat{\chi}^{\mathsf{T}}_{i,j,k}]^{\ell-1}}\right)^{\frac{[\psi_i\widehat{\chi}^{\mathsf{T}}_{i,j,k}]^{\ell-1}}{H_{i,j}([\boldsymbol{x}]^{\ell-1})}} \tag{21}$$

$$J_{i,j}(\boldsymbol{x}) = \alpha_{i,j} + \epsilon_E \geq \widehat{J}_{i,j}(\boldsymbol{x};\ell) \triangleq \left(\frac{\alpha_{i,j} J_{i,j}([\boldsymbol{x}]^{\ell-1})}{[\alpha_{i,j}]^{\ell-1}}\right)^{\frac{[\alpha_{i,j}]^{\ell-1}}{J_{i,j}([\boldsymbol{x}]^{\ell-1})}} \left(J_{i,j}([\boldsymbol{x}]^{\ell-1})\right)^{\frac{\epsilon_E}{J_{i,j}([\boldsymbol{x}]^{\ell-1})}}, \tag{22}$$

$$M^{+}_{i,j}(\boldsymbol{x}) = \chi^{C}_{i,j} + \epsilon_C + \psi_j \geq \widehat{M}^{+}_{i,j}(\boldsymbol{x};\ell) \triangleq \left(\frac{\chi^{C}_{i,j} M^{+}_{i,j}([\boldsymbol{x}]^{\ell-1})}{[\chi^{C}_{i,j}]^{\ell-1}}\right)^{\frac{[\chi^{C}_{i,j}]^{\ell-1}}{M^{+}_{i,j}([\boldsymbol{x}]^{\ell-1})}} \left(M^{+}_{i,j}([\boldsymbol{x}]^{\ell-1})\right)^{\frac{\epsilon_C}{M^{+}_{i,j}([\boldsymbol{x}]^{\ell-1})}} \left(\frac{\psi_j M^{+}_{i,j}([\boldsymbol{x}]^{\ell-1})}{[\psi_j]^{\ell-1}}\right)^{\frac{[\psi_j]^{\ell-1}}{M^{+}_{i,j}([\boldsymbol{x}]^{\ell-1})}} \tag{23}$$

$$M^{-}_{i,j}(\boldsymbol{x}) = \sum_{i\in\mathcal{N}} \alpha_{i,j} \geq \widehat{M}^{-}_{i,j}(\boldsymbol{x};\ell) \triangleq \prod_{i\in\mathcal{N}} \left(\frac{\alpha_{i,j} M^{-}_{i,j}([\boldsymbol{x}]^{\ell-1})}{[\alpha_{i,j}]^{\ell-1}}\right)^{\frac{[\alpha_{i,j}]^{\ell-1}}{M^{-}_{i,j}([\boldsymbol{x}]^{\ell-1})}} \tag{24}$$



A) Convergence of $(\boldsymbol{\mathcal{P}})$. B) Source/target classification ψ_i.

FIGURE 4: Convergence behavior and source/target device classification at convergence for Algorithm 2 with two different settings of source errors across devices with labeled/partially labeled data.

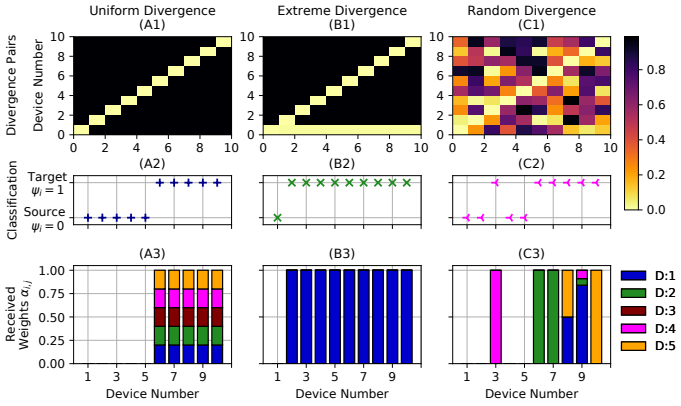
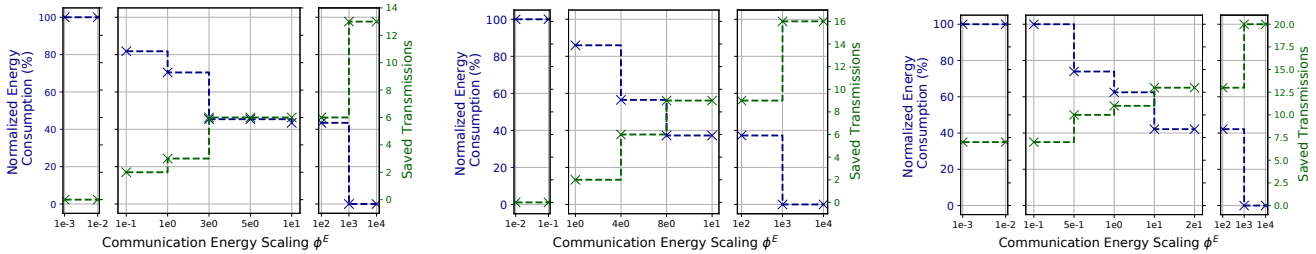


FIGURE 5: The effects of uniform, extreme, and random distribution divergence regimes on the behavior of $(\boldsymbol{\mathcal{P}})$. Each regime occupies a column, showing (i) the divergences $d_{\mathcal{H}\Delta\mathcal{H}}$ between pairs of devices, (ii) the optimized source/target classifications $\boldsymbol{\psi}$, and (iii) the optimized combination weights $\boldsymbol{\alpha}$. The third row breaks down received ML models at targets from source devices 1-5 (i.e., $D: 1, \cdots, 5$), which are proportional to the divergences.

image with label 3 in Fig. 3 is challenging even for people to decipher).

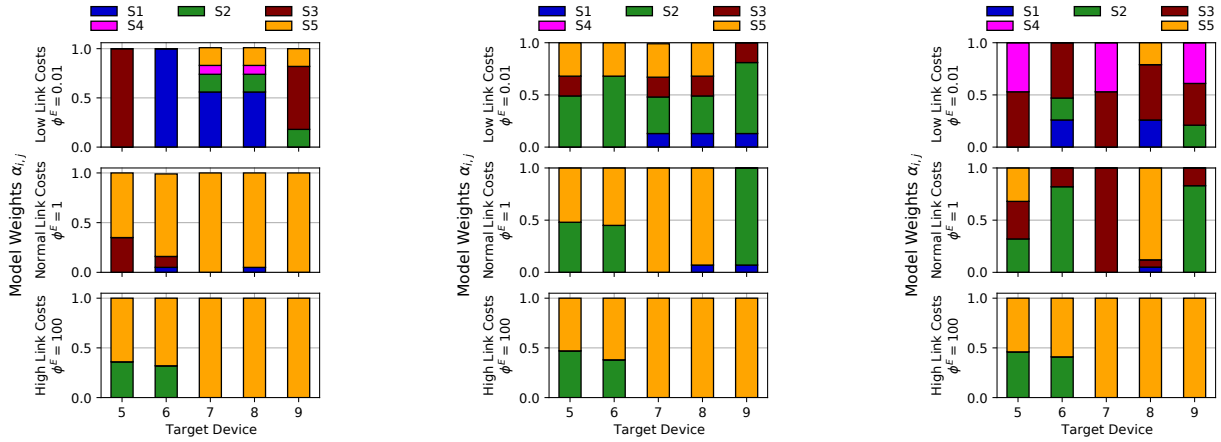
We consider a network of 10 devices, training two-layer CNNs (with 10 and 20 maps respectively) followed by two fully connected layers for all cases. For our federated divergence estimation (Algorithm 1), we use a CNN with the same architecture as above, except that the dimension at the output of the final connected layer is 2 rather than 10. Next, we distribute data across the devices in a non-i.i.d. manner, where each device has a unique Dirichlet distribution of all labels or subset of labels from the full training dataset [49]. Half of the network will have partially labeled datasets with randomly determined labeled-to-unlabeled data ratios, while the rest of the network will have completely unlabeled datasets. For the singular dataset tests involving MNIST and USPS, devices draw Dirichlet distributions from a subset of 4 labels, while for simulations involving MNIST-M and/or multiple simultaneous datasets, devices draw from the full training dataset. We locally train ML models at source devices using stochastic gradient descent (SGD) as in conventional FL, with 100 iterations, a mini-batch size of 10, and a learning rate of 0.01. Training is conducted with Pytorch [50] on a 6GB GTX 1660 SUPER with 16 GB RAM. Additionally, for ST-LF, we use $\phi^{\mathsf{S}} = 1$, $\phi^{\mathsf{T}} = 5$, and $\phi^{\mathsf{E}} = 1$ in $(\boldsymbol{\mathcal{P}})$, and all figures are the averaged results over 5 independent runs, unless otherwise stated.

Communication Energy Determination. The communication energy $E_{i,j}$ defined in (14) relies on $K_{i,j}$, a strictly communication-dependent term. For any given device pair (i,j), we define $K_{i,j} = \frac{M}{R_{i,j}} P_i$, where P_i is transmission power at device i (watts), $R_{i,j}$ is transmission rate from device i to device j (bits/sec), and M is the size of the hypothesis (i.e., ML model) transmitted in bits. For all devices $i \in \mathcal{N}$, we determine the transmission power P_i randomly between $P_{min} = 23$ dBm and $P_{max} = 25$ dBm. Similarly, for all device pairs (i,j), we select the transmission rate $R_{i,j}$ randomly between $R_{min} = 63$ Mbps and $R_{max} = 85$ Mbps. The size



A) MNIST B) USPS C) MNIST-M

FIGURE 6: Total device communication resource consumption and reduction in transmissions as a function of the energy cost of ML model transmissions (ϕ^{E}) in ($\boldsymbol{\mathcal{P}}$). As ϕ^{E} increases, ST-LF adjusts the combination weights $\boldsymbol{\alpha}$ so that fewer sources transfer their models to targets, with the energy consumption decreasing accordingly. Each dataset has different sensitivity to ϕ^{E}.



A) MNIST B) USPS C) MNIST-M

FIGURE 7: Effect of communication energy scaling, (i) low cost - $\phi^{\mathsf{E}} = 0.01$, (ii) medium cost - $\phi^{\mathsf{E}} = 1$, and (iii) high cost - $\phi^{\mathsf{E}} = 100$, on the link formation weights $\boldsymbol{\alpha}$ for each of the datasets. The network and experiment setup are identical to those of Fig. 6. As ϕ^{E} increases, we see new combination weights involving fewer unique source-to-target transmissions in order to reduce communication costs.

of the hypothesis in bits is assumed to be fixed at 1 Gbit. In particular, these measurements are similar to commonly used measurements in autonomous systems research [51], and their selection highlights the versatility of our formulation and optimization methodologies.

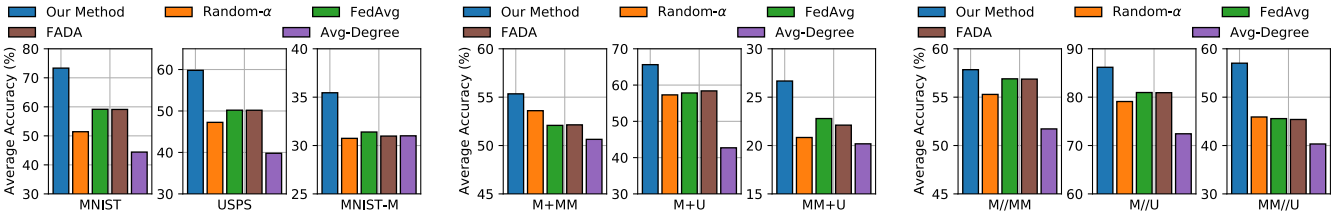
A. ST-LF Optimization Solution

We study multiple aspects of ($\boldsymbol{\mathcal{P}}$) in-depth. We first demonstrate the convergence of our iterative methodology for ($\boldsymbol{\mathcal{P}}$) (Algorithm 2), and thereafter focus on our proposed federated divergence estimation technique (Algorithm 1), which influences source-target classification and link formation weights. Finally, we examine the effect of ϕ^{E} on the network communication resource consumption.

Optimization Convergence and Source Error Sensitivity. In Fig. 4A) and Fig. 4B), we investigate (i) the convergence of our iterative methodology for ($\boldsymbol{\mathcal{P}}$) and (ii) the effect of having a large empirical error when a device has labelled data. For both figures, we assume that five devices have labeled or partially labeled datasets and the other five devices only have unlabeled data. We first show that, regardless of empirical error on a labeled dataset, our solution for ($\boldsymbol{\mathcal{P}}$) is able to reach convergence in a monotonically decreasing fashion. Then, in Fig. 4B), we show that a large empirical error at device 3 results in ($\boldsymbol{\mathcal{P}}$) classifying that device as a target even though it may have labeled data. Essentially, our optimization is able to balance the trade-offs between having more sources (i.e., more training data) versus the quality of those sources. In scenarios where a device has a small quantity and/or poor quality of locally labeled data, it may be more effective from a network resource perspective to classify that device as a target.

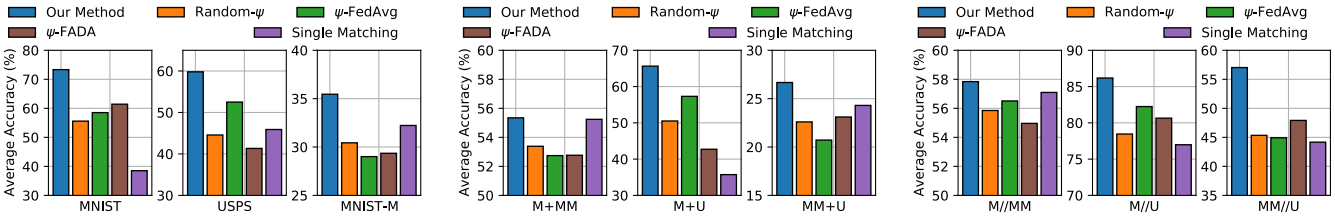
Adapting to Distribution Divergence. Fig. 5 demonstrates the effect of distribution divergence $\widehat{d}_{\mathcal{H}\Delta\mathcal{H}}$ on ST-LF's optimization solution. Recall from Sec. IV-B and Algorithm 1 that a large divergence means that two devices are statistically dissimilar while a small divergence implies that they have highly similar local dataset characteristics. We depict three separate regimes in the columns of Fig. 5: (A) *uniform:* devices have identical pairwise divergences of 1; (B) *extreme:* one device has the minimum pairwise divergence, 0, to all other devices while other pairs have the maximum, 1; and (C) *random:* device pairs have randomly assigned divergences. The first row of Fig. 5 are 2D colormaps of the divergence values, where element (i, j) indicates $\widehat{d}_{\mathcal{H}\Delta\mathcal{H}}(i, j)$, while the second and third row depict the resulting source/target classifications $\boldsymbol{\psi}$ and combination weights $\boldsymbol{\alpha}$ from solving ($\boldsymbol{\mathcal{P}}$).

In the uniform regime (A1-A3), devices are essentially statistically identical to each other, so targets should receive uniformly weighted model parameters from the sources. ST-LF adapts to this by classifying all devices with labeled data (i.e., devices 1 to 5) as sources and unlabeled devices as targets in (A2), and forming uniformly weighted links between all



A) Single dataset B) Mixed dataset C) Split dataset

FIGURE 8: Comparing the model weight determination, $\boldsymbol{\alpha}$, of our method (ST-LF) versus several baselines (e.g., FedAvg [3], FADA [8]) on single, mixed, and split dataset settings. All methods rely on ST-LF's source/target classification output ($\boldsymbol{\psi}$), with each bar representing the average classification accuracy across all target devices obtained over five independent runs of each algorithm. Mixed datasets are denoted by "+", e.g., M+U means devices have a mixture of data from the MNIST and USPS datasets, while split datasets are described by "//", e.g., M//U means that some devices only have data from the MNIST dataset while others will exclusively contain data from the USPS dataset.



A) Single dataset B) Mixed dataset C) Split dataset

FIGURE 9: The impact of source/target classification, $\boldsymbol{\psi}$, on average target classification accuracy. When baseline methods (e.g., ψ-FL [3], ψ-FADA [8]) do not have access to optimal source/target determination from our method (ST-LF), they generally perform worse than in Fig. 8 relative to our method, demonstrating the importance of joint source/target classification, $\boldsymbol{\psi}$, and model/link weight determination, $\boldsymbol{\alpha}$.

source-target pairs in (A3). In the extreme divergence regime (B1-B3), there is one device that is statistically perfect for every other device in the network. ST-LF detects this and correspondingly assigns device 1 as the only source, with all source-target combination weights set to 1. With the random divergence regime (C1-C3), our methodology needs to be able to pick-out trade-offs among the devices. For example, if a device has labeled data but is very similar to the data at another device with labeled data, then it may be better to only keep one of these two devices as a source. ST-LF adapts well in this more challenging setting, determining that device 3 has high statistical similarity with device 4, and thus reclassifies device 3 as a target in (C2). Finally, the link weights $\boldsymbol{\alpha}$, which also reflect model combination weights, vary based on the $d_{\mathcal{H}\Delta\mathcal{H}}$ values, which makes sense as target devices should favor the trained models at statistically similar sources devices.

Impact of Communication Energy Importance. Communication energy expenditures in ST-LF follow a discrete pattern, where the network only sees a change in both total transmission and energy consumption at distinct thresholds. For example, when $\alpha_{i,j}$ changes numerically from 0.2 to 0.6, the link between device i and device j incurs the same energy cost, though the weight of those ML model parameters changes. To effect a change, $\alpha_{i,j}$ must become deactivated, i.e., goes to zero, or become activated, i.e., increases from zero to a non-zero value. We investigate the ability of our methodology to control these thresholds via varying the communication energy scaling ϕ^{E} from $(\boldsymbol{\mathcal{P}})$ in Fig. 6.

Specifically, Fig. 6 depicts the ability of ϕ^{E} to influence total network transmissions and energy consumption. On the left y-axis, Fig. 6 measures the normalized energy consumption, a metric computed from term (e) in (11), while the right y-axis measures saved transmissions. Both normalized energy consumption and saved transmissions are measured relative to the case of $\phi^{\mathsf{E}} = 0$. For the lowest values of ϕ^{E} in Fig. 6, the energy consumption is saturated, effectively equivalent to solving $(\boldsymbol{\mathcal{P}})$ without the energy term (e). As ϕ^{E} increases, energy consumption decreases at discrete intervals, corresponding to exact points of ϕ^{E} when links deactivate. Additionally, different datasets have inherently unique response patterns to ϕ^{E}: ST-LF saves up to 13, 16, and 20 transmissions for MNIST, USPS, and MNIST-M when $\phi^{\mathsf{E}} \geq 1e3$. Finally, we notice a saturation effect as ϕ^{E} reaches $1e3$, as further increases in ϕ^{E} do not reduce energy consumption or save any additional transmissions. This is because, when ϕ^{E} exceeds $1e3$, links become prohibitively expensive to activate, so ST-LF designates all the devices as targets instead.

We visualize the change in weights $\alpha_{i,j}$ as a result of different ϕ^{E} values in Fig. 7. For example, for the MNIST experiment in Fig. 7A), increasing ϕ^{E} from 0.01 to 1 eliminates transmissions from source device 4 (i.e., $S4$), and this leads to some energy savings and saved transmissions in Fig. 6A). Fig. 7 also shows that for scenarios with large ϕ^{E}, the cost of model transmissions becomes the optimization bottleneck, resulting in only two sources that transmit to targets. Conversely, for scenarios with low ϕ^{E}, the optimization is able to focus more on the expected ML performance, resulting in more sources that transmit their ML models to targets. The lowest value $\phi^{\mathsf{E}} = 0.01$ corresponds to the case of full energy consumption from Fig. 6.

B. ST-LF Accuracy and Efficiency Enhancements

To evaluate the full ST-LF methodology, we now compare its effectiveness, as measured by average classification accura-

cies at target devices and total network energy consumption, relative to eight baselines on three commonly used domain adaptation datasets (MNIST [46], USPS [52], MNIST-M [53], [32]) depicted in Fig. 3. We will use the abbreviations M for MNIST, U for USPS, and MM for MNIST-M in Fig. 8, Fig. 9, and Table I. We also provide an empirical investigation on the tightness/looseness of the theoretical results in Appendix G

Current literature in the field commonly assumes that source/target devices are known apriori, and, to the best of our knowledge, works have yet to investigate source/target determination. Instead, current literature such as FADA [8] has focused on determining model transmission or link weights, $\boldsymbol{\alpha}$, from source devices to a single target device. As a result, two of the baselines that we consider, (i) FedAvg [3] and (ii) FADA [8], require source/target device classification prior to operation. In this case, we feed these two algorithms the source/target classification output from ST-LF, i.e., the $\boldsymbol{\psi}$ from our method. Two other baselines, (iii) ψ-FedAvg and (iv) ψ-FADA, rely on heuristic determinations of source/target classification ψ combined with the model weight determination, α, from FedAvg [3] and FADA [8]. In order to ensure a fair comparison between FADA [8] and ST-LF, FADA's generators have the same architecture as ST-LF's binary hypothesis classifiers, except that FADA's generators have a larger output layer dimension to accommodate its subsequent operations. In this manner, we are able to separately measure the benefits of joint source/target classification and link weight determination from ST-LF. We also provide four additional heuristic baselines, (v) Random-α, (vi) Avg Degree, (vii) Random-ψ, and (viii) Single Matching; we will detail each of these baselines in their relevant sections.

Effect of Link Formation $\boldsymbol{\alpha}$. In Fig. 8 and Table I, we compare ST-LF against the four baselines that determine $\boldsymbol{\alpha}$:

(i) *Random-α (Rnd-α)*, which allocates link formation $\boldsymbol{\alpha}$ according to a Dirichlet distribution so that $\sum_{s\in\mathcal{S}} \alpha_{s,t} = 1$,

(ii) *FedAvg* [3], which scales link weights proportionally to sources' local dataset size,

(iii) *FADA* [8], which relies on unsupervised data clustering, distributed adversarial alignment (similar to GAN techniques), and feature disentangling networks to determine the link weights in federated domain adaptation,

(iv) *Avg Degree (AvgD)*, where each source is allocated the average number of links/source from ST-LF, but the specific links and weights are randomly determined.

All baselines rely on ST-LF's $\boldsymbol{\psi}$ determination.

The experiments shown in Fig. 8 consist of three kinds of ML dataset manipulation: (i) single ML dataset (e.g., USPS) - all devices draw data from the same underlying dataset, (ii) mixed ML dataset (e.g. M+MM) - all devices draw data from the same underlying dataset composed of multiple datasets from Fig. 3, (iii) split ML dataset (e.g., U//MM) - devices will draw from different ML datasets (e.g., for U//MM, a source device may draw data from USPS while a target device may contain data from MNIST-M). The single dataset experiment is a more classical federated scenario, where devices have non-i.i.d. data distributions drawn from the same ML dataset, but augmented with unlabeled data. Meanwhile, the mixed and split dataset experiments are bigger challenges, jointly involving statistical heterogeneity (i.e., non-i.i.d. data distributions) and dataset heterogeneity.

TABLE I: Average target accuracy vs. communication energy (Nrg) consumption obtained by algorithms depicted in Fig. 8 and 9. Communication energy is normalized (norm) relative to the maximum energy use in each category, and we exclude energy consumption measurements of server-dependent algorithms. The improvements that our methodology (ST-LF) obtains on both metrics emphasizes the importance of joint optimization in decentralized federated settings.

	Comparing ψ			**Comparing α**		
	Method	**Avg Acc (%)**	**Norm Nrg (%)**	**Method**	**Avg Acc (%)**	**Norm Nrg (%)**
MNIST	Ours	**71.32**	21.10	Ours	**71.32**	**21.10**
	Rnd-ψ	55.82	94.87	Rnd-α	54.54	99.15
	ψ-FedAvg	59.64	—	FedAvg	60.95	—
	ψ-FADA	63.18	93.79	FADA	61.40	100
	SM	51.07	**19.38**	AvgD	58.63	55.15
USPS	Ours	**60.23**	22.20	Ours	**60.23**	**22.20**
	Rnd-ψ	47.97	94.87	Rnd-α	52.40	99.18
	ψ-FedAvg	55.55	—	FedAvg	54.00	—
	ψ-FADA	46.60	87.21	FADA	54.46	100
	SM	44.72	**19.38**	AvgD	53.35	57.83
MNIST-M	Ours	**36.44**	27.70	Ours	**36.44**	**27.70**
	Rnd-ψ	30.11	94.87	Rnd-α	29.69	99.38
	ψ-FedAvg	29.64	—	FedAvg	30.74	—
	ψ-FADA	28.79	93.67	FADA	30.20	100
	SM	32.05	**19.38**	AvgD	30.60	69.60
M+MM	Ours	**56.43**	24.11	Ours	**56.43**	**24.11**
	Rnd-ψ	54.02	94.87	Rnd-α	53.28	98.41
	ψ-FedAvg	53.15	—	FedAvg	52.00	—
	ψ-FADA	54.23	87.78	FADA	52.00	100
	SM	54.93	**19.38**	AvgD	52.28	70.69
M+U	Ours	**68.01**	20.56	Ours	**68.01**	**20.56**
	Rnd-ψ	49.93	94.87	Rnd-α	58.60	100
	ψ-FedAvg	55.85	—	FedAvg	59.61	—
	ψ-FADA	40.79	82.44	FADA	59.90	100
	SM	51.73	**19.38**	AvgD	56.49	55.15
MM+U	Ours	**26.64**	22.44	Ours	**26.64**	**22.44**
	Rnd-ψ	22.04	94.42	Rnd-α	20.89	98.60
	ψ-FedAvg	19.60	—	FedAvg	23.39	—
	ψ-FADA	21.62	100	FADA	22.66	99.53
	SM	23.33	**19.29**	AvgD	22.08	67.69
M//MM	Ours	**56.95**	23.28	Ours	**56.95**	**23.28**
	Rnd-ψ	56.20	94.87	Rnd-α	54.90	100
	ψ-FedAvg	56.12	—	FedAvg	56.36	—
	ψ-FADA	55.30	93.79	FADA	56.23	100
	SM	56.86	**19.37**	AvgD	56.11	63.28
M/U	Ours	**86.26**	25.28	Ours	**86.26**	**25.28**
	Rnd-ψ	78.46	94.87	Rnd-α	78.46	99.35
	ψ-FedAvg	82.25	—	FedAvg	80.39	—
	ψ-FADA	80.65	97.68	FADA	80.47	100
	SM	79.11	**19.38**	AvgD	79.79	65.72
MM/U	Ours	**54.70**	23.49	Ours	**54.70**	**23.49**
	Rnd-ψ	45.34	94.87	Rnd-α	46.46	99.35
	ψ-FedAvg	44.94	—	FedAvg	45.24	—
	ψ-FADA	47.92	94.23	FADA	45.39	100
	SM	47.29	**19.38**	AvgD	45.90	63.43

ST-LF outperforms the α-baselines substantially in all single dataset cases: by $>9\%$ for MNIST, and $>5\%$ for both USPS and MNIST-M in terms of average accuracy. The mixed and split dataset evaluations also show consistent and significant improvements, though more marginal for cases involving combinations of MNIST-M with other datasets, which is

expected given the more challenging nature of mixed and split dataset scenarios. The importance is that ST-LF maintains its significant performance benefits over all baselines in all cases, highlighting its effectiveness at determining model/link weights $\boldsymbol{\alpha}$.

In Table I, we also see substantial improvements in network-wide communication energy consumption compared to the multi-source baselines, due to the fact that ST-LF forms fewer source-target links. Despite forming fewer source-to-target links, our method achieves the best accuracies and the largest energy consumption savings relative to all other baselines for model/link weights $\boldsymbol{\alpha}$. Mixed and split dataset energy consumption measurements in Table I preserve the same pattern as those for single dataset experiments. Energy measurements for server dependent baselines (i.e., FedAvg for $\boldsymbol{\alpha}$ baselines and ψ-FedAvg for $\boldsymbol{\psi}$ baselines) are omitted as they depend on the distance between server and devices. Overall, we see significant improvements for model/link weights $\boldsymbol{\alpha}$ even when the baselines employ ST-LF's solution for source/-target classification $\boldsymbol{\psi}$, which emphasizes the importance of our joint optimization methodology.

Effect of Source/Target Determination $\boldsymbol{\psi}$. Next, we investigate the impact of source/target determination $\boldsymbol{\psi}$ by comparing ST-LF against the four baselines that influence ψ:

(v) *Random-ψ (Rnd-ψ)*, which randomly classifies each device as a source or target and then uses Rnd-α for the model/link weights,

(vi) *ψ-FedAvg*, which uses the same heuristic method to determine $\boldsymbol{\psi}$ as (i) combined with FedAvg's method for model/link weights $\boldsymbol{\alpha}$,

(vii) *ψ-FADA*, which uses the same method as (ii) but with FADA [8] replacing FedAvg [3],

(viii) *Single Matching (SM)*, which is single source to single target matching inspired by [34].

Fig. 9 compares the resulting target accuracies across all methods while parts of Table I show the resulting energy consumption for all methods. Analogous to Fig. 8, Fig. 9 compares these methods on three kinds of ML dataset manipulations: (1) single ML dataset in Fig. 9A), (2) mixed ML datasets in Fig. 9B), and (3) split ML datasets in Fig. 9C).

The single dataset experiment, Fig. 9A), effectively compares ST-LF against the baselines in the classical federated scenario, where devices have non-i.i.d. data distributions albeit combined with the presence of unlabeled data. In this setting, ST-LF obtains significant performance improvements over the $\boldsymbol{\psi}$-baselines, specifically by $> 8\%$ on MNIST, and $> 4\%$ on both USPS and MNIST-M, demonstrating its ability to select the most optimal sources in the classical federated case.

On the other hand, the mixed dataset case in Fig. 9B) and the split dataset case in Fig. 9C) feature multiple ML datasets in the same network environment, thus augmenting the classical federated challenge of statistical heterogeneity (i.e., non-i.i.d. data distributions) with dataset heterogeneity. In this more challenging problem, ST-LF maintains a clear and consistent performance advantage relative to all baselines, demonstrating its ability to jointly consider statistical similarity as well as dataset similarity when determining the optimal set of source devices. In Table I, ST-LF continues to demonstrate the best accuracies relative to other baselines. One heuristic baseline, single matching (SM), does provide incremental energy savings over ST-LF, but leads to substantially worse ML performance across all test cases. Furthermore, because it is a one-to-one matching, the performance of SM is highly variable across different settings (e.g., the 20% drop in performance relative to ST-LF on MNIST). Overall, when considering joint design of source/target classification, source-to-target model/link weights, and energy resource efficiency, ST-LF continues to show a commanding performance.

VI. Conclusion

We investigated decentralized FL in settings where devices have partially labeled datasets of varying quality. This challenging problem augments standard federated settings, which are known for communication heterogeneity and statistical heterogeneity (i.e., non-i.i.d. data distributions) across network devices, by introducing heterogeneous quantities and distributions of unlabeled data across network devices and integrating combinations of unique/independent ML datasets into the standard FL problem. We addressed this problem through a novel methodology, ST-LF, for multi-source to multi-target federated domain adaptation. In developing ST-LF, we obtained theoretical results for domain generalization errors which are measurable in real-world systems. Based on these results, we formulated a concrete optimization problem that jointly (i) determines the optimal source/target classification of devices, (ii) obtains link weights (model combination weights) to match sources to targets, and (iii) considers communication efficiency of model transmissions. We showed that ST-LF belongs to a class of mixed-integer signomial programs, which are NP-hard and non-convex, and developed an iterative method to solve it based on refining convex inner approximations. Finally, we demonstrated both performance and energy efficiency improvements obtained by ST-LF relative to baselines on common domain adaptation datasets.

References

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

[2] 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 Commun. Surv. Tut.*, vol. 22, no. 3, pp. 2031–2063, 2020.

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

[4] B. Luo, X. Li, S. Wang, J. Huang, and L. Tassiulas, "Cost-effective federated learning design," in *INFOCOM*, 2021, pp. 1–10.

[5] H. Ye, L. Liang, and G. Y. Li, "Decentralized federated learning with unreliable communications," *IEEE J. Sel. Topics Signal Process.*, vol. 16, no. 3, pp. 487–500, 2022.

[6] A. G. Roy, S. Siddiqui, S. Pölsterl, N. Navab, and C. Wachinger, "Braintorrent: A peer-to-peer environment for decentralized federated learning," *arXiv:1905.06731*, 2019.

[7] S. Wang, M. Lee, S. Hosseinalipour, R. Morabito, M. Chiang, and C. G. Brinton, "Device sampling for heterogeneous federated learning: Theory, algorithms, and implementation," in *INFOCOM*, 2021, pp. 1–10.

[8] X. Peng, Z. Huang, Y. Zhu, and K. Saenko, "Federated adversarial domain adaptation," in *Int. Conf. Learn. Representations*, 2019.

[9] L. Schmarje, M. Santarossa, S.-M. Schröder, and R. Koch, "A survey on semi-, self-and unsupervised learning for image classification," *IEEE Access*, vol. 9, pp. 82 146–82 168, 2021.

[10] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," *Mach. Learn.*, vol. 79, no. 1, pp. 151–175, 2010.

[11] Y. Mansour, M. Mohri, and A. Rostamizadeh, "Domain adaptation with multiple sources," in *Advances Neural Inf. Process. Syst.*, vol. 21, 2008.

[12] H. Zhao, S. Zhang, G. Wu, J. M. Moura, J. P. Costeira, and G. J. Gordon, "Adversarial multiple source domain adaptation," in *Advances Neural Inf. Process. Syst.*, vol. 31, 2018.

[13] N. Zhao, Z. Liu, and Y. Cheng, "Multi-agent deep reinforcement learning for trajectory design and power allocation in multi-UAV networks," *IEEE Access*, vol. 8, pp. 139 670–139 679, 2020.

[14] H. Wang, Z. Kaplan, D. Niu, and B. Li, "Optimizing federated learning on non-iid data with reinforcement learning," in *INFOCOM*, 2020, pp. 1698–1707.

[15] M. Tang and V. W. Wong, "An incentive mechanism for cross-silo federated learning: A public goods perspective," in *INFOCOM*, 2021, pp. 1–10.

[16] V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar, "Federated multi-task learning," in *Advances Neural Inf. Process. Syst.*, vol. 30, 2017.

[17] A. Fallah, A. Mokhtari, and A. Ozdaglar, "Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach," in *Advances Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 3557–3568.

[18] C. T Dinh, N. Tran, and J. Nguyen, "Personalized federated learning with moreau envelopes," in *Advances Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 21 394–21 405.

[19] S. Wang, S. Hosseinalipour, M. Gorlatova, C. G. Brinton, and M. Chiang, "Uav-assisted online machine learning over multi-tiered networks: A hierarchical nested personalized federated learning approach," *IEEE Trans. Netw. Service Manage.*, 2022.

[20] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-iid data," *arXiv:1806.00582*, 2018.

[21] N. Lu, Z. Wang, X. Li, G. Niu, Q. Dou, and M. Sugiyama, "Federated learning from only unlabeled data with class-conditional-sharing clients," in *Int. Conf. Learn. Representations*, 2021.

[22] Q. Li, B. He, and D. Song, "Model-contrastive federated learning," in *IEEE/CVF Conf. Comput. Vision Pattern Recognit.*, 2021, pp. 10 713–10 722.

[23] S. Wang, S. Hosseinalipour, V. Aggarwal, C. G. Brinton, D. J. Love, W. Su, and M. Chiang, "Towards cooperative federated learning over heterogeneous edge/fog networks," *arXiv:2303.08361*, 2023.

[24] S. Wang, R. Sahay, and C. G. Brinton, "How potent are evasion attacks for poisoning federated learning-based signal classifiers?" *arXiv:2301.08866*, 2023.

[25] J. Zhang, N. Li, and M. Dedeoglu, "Federated learning over wireless networks: A band-limited coordinated descent approach," in *INFOCOM*. IEEE, 2021, pp. 1–10.

[26] S. Wang, Y. Ruan, Y. Tu, S. Wagle, C. G. Brinton, and C. Joe-Wong, "Network-aware optimization of distributed learning for fog computing," *IEEE/ACM Trans. Netw.*, vol. 29, no. 5, pp. 2019–2032, 2021.

[27] D. Avdiukhin and S. Kasiviswanathan, "Federated learning under arbitrary communication patterns," in *Int. Conf. Mach. Learn.* PMLR, 2021, pp. 425–435.

[28] S. Hosseinalipour, S. Wang, N. Michelusi, V. Aggarwal, C. G. Brinton, D. J. Love, and M. Chiang, "Parallel successive learning for dynamic distributed model training over heterogeneous wireless networks," *arXiv:2202.02947*, 2022.

[29] S. Motiian, M. Piccirilli, D. A. Adjeroh, and G. Doretto, "Unified deep supervised domain adaptation and generalization," in *Int. Conf. Comput. Vis.*, 2017, pp. 5715–5725.

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

[31] K. Saito, D. Kim, S. Sclaroff, T. Darrell, and K. Saenko, "Semi-supervised domain adaptation via minimax entropy," in *Int. Conf. Comput. Vis.*, 2019, pp. 8050–8058.

[32] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2096–2030, 2016.

[33] B. Sun and K. Saenko, "Deep coral: Correlation alignment for deep domain adaptation," *arXiv:1607.01719*, 2016.

[34] H. Zhao, R. T. Des Combes, K. Zhang, and G. Gordon, "On learning invariant representations for domain adaptation," in *Int. Conf. Mach. Learn*. PMLR, 2019, pp. 7523–7532.

[35] L. Zhang, X. Lei, Y. Shi, H. Huang, and C. Chen, "Federated learning for iot devices with domain generalization," *IEEE Internet Things J.*, 2023.

[36] V.-T. Tran, H.-H. Pham, and K.-S. Wong, "Personalized privacy-preserving framework for cross-silo federated learning," *arXiv:2302.12020*, 2023.

[37] C.-H. Yao, B. Gong, H. Qi, Y. Cui, Y. Zhu, and M.-H. Yang, "Federated multi-target domain adaptation," in *IEEE/CVF Winter Conf. Appl. Comput. Vision*, 2022, pp. 1424–1433.

[38] P. L. Bartlett and S. Mendelson, "Rademacher and gaussian complexities: Risk bounds and structural results," *J. Mach. Learn. Res.*, vol. 3, no. Nov, pp. 463–482, 2002.

[39] N. H. Tran, W. Bao, A. Zomaya, N. M. NH, and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *INFOCOM*.

[40] Z. Ning, X. Wang, X. Kong, and W. Hou, "A social-aware group formation framework for information diffusion in narrowband internet of things," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1527–1538, 2017.

[41] J. Zhang and S. Y. Philip, "Integrated anchor and social link predictions across social networks," in *Int. Joint Conf. Artif. Intell.*, 2015.

[42] S. Diamond and S. Boyd, "CVXPY: A Python-embedded modeling language for convex optimization," *J. Machine Learn. Res.*, vol. 17, no. 83, pp. 1–5, 2016.

[43] M. Chiang, *Geometric programming for communication systems*. Now Publishers Inc, 2005.

[44] G. Xu, "Global optimization of signomial geometric programming problems," *Eur. J. Oper. Res.*, vol. 233, no. 3, pp. 500–510, 2014.

[45] R. J. Duffin and E. L. Peterson, "Reversed geometric programs treated by harmonic means," *Indiana Univ. Math. J.*, vol. 22, no. 6, pp. 531–550, 1972.

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

[47] J. J. Hull, "A database for handwritten text recognition research," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 5, pp. 550–554, 1994.

[48] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, 2010.

[49] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, "A novel framework for the analysis and design of heterogeneous federated learning," *IEEE Trans. Signal Process.*, vol. 69, pp. 5234–5249, 2021.

[50] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances Neural Inf. Process. Syst.*, vol. 32, 2019.

[51] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Mobile unmanned aerial vehicles (uavs) for energy-efficient internet of things communications," *IEEE Trans. Wireless Commun.*, vol. 16, no. 11, pp. 7574–7589, 2017.

[52] J. J. Hull, "A database for handwritten text recognition research," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 5, pp. 550–554, 1994.

[53] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Int. Conf. Mach. Learn*.

[54] J. Abernethy, Y.-J. Chang, Y. Sun, and D. Hong. Lecture 13: Rademacher complexity and massart's lemma. [Online]. Available: https://web.eecs.umich.edu/~jabernet/eecs598course/fall2015/web/notes/lec13_102715.pdf

[55] S. Shalev-Shwartz and S. Ben-David, *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[56] S. Boyd, S.-J. Kim, L. Vandenberghe, and A. Hassibi, "A tutorial on geometric programming," *Opt. Eng.*, vol. 8, no. 1, p. 67, 2007.

[57] M. Chiang, C. W. Tan, D. P. Palomar, D. O'neill, and D. Julian, "Power control by geometric programming," *IEEE Trans. Wireless Commun.*, vol. 6, no. 7, pp. 2640–2651, 2007.

(Henry) Su Wang received the Ph.D. in ECE from Purdue University.

Seyyedali Hosseinalipour is an assistant professor of EE at the University at Buffalo (SUNY).

Christopher G. Brinton (S'08, M'16, SM'20) is the Elmore Rising Star Assistant Professor of ECE at Purdue University.