# Towards Hetero-Client Federated Multi-Task Learning

Yuxiang Lu*, Suizhi Huang*, Yuwen Yang, Shalayiding Sirejiding, Yue Ding, Hongtao Lu
Department of Computer Science and Engineering, Shanghai Jiao Tong University

## Abstract

*Federated Learning (FL) enables joint training across distributed clients using their local data privately. Federated Multi-Task Learning (FMTL) builds on FL to handle multiple tasks, assuming model congruity that identical model architecture is deployed in each client. To relax this assumption and thus extend real-world applicability, we introduce a novel problem setting, Hetero-Client Federated Multi-Task Learning (HC-FMTL), to accommodate diverse task setups. The main challenge of HC-FMTL is the model incongruity issue that invalidates conventional aggregation methods. It also escalates the difficulties in accurate model aggregation to deal with data and task heterogeneity inherent in FMTL. To address these challenges, we propose the* FedHCA$^2$ *framework, which allows for federated training of personalized models by modeling relationships among heterogeneous clients. Drawing on our theoretical insights into the difference between multi-task and federated optimization, we propose the Hyper Conflict-Averse Aggregation scheme to mitigate conflicts during encoder updates. Additionally, inspired by task interaction in MTL, the Hyper Cross Attention Aggregation scheme uses layer-wise cross attention to enhance decoder interactions while alleviating model incongruity. Moreover, we employ learnable Hyper Aggregation Weights for each client to customize personalized parameter updates. Extensive experiments demonstrate the superior performance of* FedHCA$^2$ *in various HC-FMTL scenarios compared to representative methods. Our code will be made publicly available.*

## 1. Introduction

Federated Learning (FL) [31] has emerged as a prominent paradigm in distributed training, gaining attention in both academic and industrial fields [4, 5, 17, 27, 79]. The FL framework empowers to collaboratively train models across multiple clients, like mobile devices or distributed data centers, while preserving data privacy and reducing communication costs. The impetus behind FL lies in the recognition that harnessing a broader dataset can improve model performance, but it also introduces the *data heterogeneity* issue, as clients often collect samples from non-i.i.d. data distributions. Nevertheless, most FL research is centered on single-task scenarios, overlooking applications that demand simultaneous multi-task processing, *e.g.*, autonomous driving [23]. This gap has led to the integration of Multi-Task Learning (MTL) with FL, giving rise to Federated Multi-Task Learning (FMTL) [22, 54]. While existing FMTL approaches primarily address statistical challenges [51, 65], recent studies [8, 12, 92] have highlighted the importance of *task heterogeneity*, particularly for dense predictions such as semantic segmentation and depth estimation [11, 58, 74].

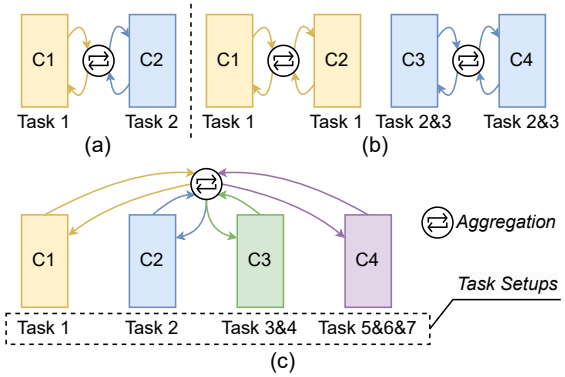


Figure 1. Comparison of different settings in FMTL. (a) Each client is dedicated to a single task. (b) Clients are grouped with peers, and peers in the same group share identical task setting. (c) Our proposed HC-FMTL setting that enables flexible collaboration among clients with different task setups.

However, these FMTL methods often assume model congruity among clients, *i.e.*, all participants either engage in a single task or aggregate with peers handling identical task sets, as shown in Fig. 1a, 1b. Considering the discrepancy of heterogeneous tasks in practical applications as well as the expensive labor of annotating task-specific labels, clients often have different task setups in different environments. Here *task setup* describes a set of tasks that can vary in both number and type. We define this as a new problem setting: ***Hetero-Client Federated Multi-Task Learning*** (HC-FMTL), as depicted in Fig. 1c. HC-FMTL relaxes the constraints on model congruity, facilitating more flexible collaborative learning of various tasks across diverse private

*Equal contribution.

data domains and making FMTL scenarios more universally applicable.

As a more pervasive setting relaxed from FMTL, HC-FMTL introduces an additional challenge of ***model incongruity***, which exacerbates client heterogeneity. This issue arises from clients having different task setups, coupled with the prevalent use of encoder-decoder architectures in vision tasks, leading to a disparity in multi-task model structures. Model incongruity not only increases the complexity of model aggregation but also coexists with the data and task heterogeneity inherent in FMTL. Data heterogeneity is a consequence of clients encountering distinct data domains, as clients tend to use data from different domains to handle different target tasks without any overlap, which can result in performance degradation of collective learning. Meanwhile, task heterogeneity, which assigns different objectives for each task, could impede joint optimization and magnify the influence of data heterogeneity.

In this paper, we propose a novel framework named **FEDHCA$^2$**, designed for HC-FMTL. Our goal is to adaptively discern the relationships among heterogeneous clients and learn personalized yet globally collaborative models that benefit from both synergies and distinctions among clients and tasks. Since model incongruity precludes the straightforward application of conventional aggregation methods in FL, our approach involves the server disassembling client models into encoders and decoders for independent aggregation. For the encoders, we design the Hyper Conflict-Averse Aggregation scheme to alleviate update conflicts among clients. The motivation behind this is grounded in our theoretical analysis (see Theorem 1) that the optimization processes of MTL and FL are closely connected and share similarities. By incorporating an approximated gradient smoothing technique, we can find an appropriate update direction for all clients that mitigates the negative effects of conflicting parameter updates caused by data and task heterogeneity. When aggregating the decoders, we devise the Hyper Cross Attention Aggregation scheme to accommodate client heterogeneity. We draw inspiration from the modeling of task interaction in MTL [55, 77] and apply it to FL. Specifically, we implement a layer-wise cross attention mechanism to model the interplay between client decoders, enabling the capture of both the commonalities and discrepancies among different tasks in a fine-grained manner and thereby alleviating the incongruity at the model level. In addition, the personalized parameter updates for each client are tailored by learnable Hyper Aggregation Weights, which encourage encoders and decoders to adaptively assimilate knowledge from peers that offer helpful complementary information.

Our contributions are summarized as follows:

- We introduce a novel setting of Hetero-Client Federated Multi-Task Learning (HC-FMTL) alongside the FEDHCA$^2$ framework. It supports collaborative training across clients, each with its unique task setups, addressing the complexities of data and task heterogeneity, and the newly identified challenge of model incongruity. The relaxed setting broadens the FMTL's applicability to include a wider variety of clients, tasks, and data situations.
- We reveal the connection between the optimization of MTL and FL in Theorem 1 and underscore the importance of circumventing update conflicts among clients, which are exacerbated by data and task heterogeneity in HC-FMTL. We propose a Hyper Conflict-Averse Aggregation scheme, designed to alleviate the adverse effects on encoders when absorbing shared knowledge.
- We develop a Hyper Cross Attention Aggregation scheme to facilitate task interaction in decoders by modeling the fine-grained cross-task relationships among each decoder layer, tackling both intra- and inter-client heterogeneity.
- We evaluate FEDHCA$^2$ using a composite of two benchmark datasets, PASCAL-Context and NYUD-v2, for various HC-FMTL scenarios. Extensive experiments demonstrate that our approach outperforms existing methods.

## 2. Related Work

### 2.1. Personalized Federated Learning

Federated Learning (FL) can be broadly classified into traditional and personalized types, depending on the characteristics of data distribution [42, 68]. Traditional Federated Learning, exemplified by the widely used FedAvg [53], has undergone refinements to tackle challenges such as data heterogeneity [1, 33, 34, 72, 73, 83, 85, 91], communication efficiency [18, 29, 32, 52, 90], and privacy concerns [3, 25]. In contrast, personalized Federated Learning (pFL) emerges as a specialized variant designed to cater to individual client needs and address data heterogeneity more effectively [65, 68]. Techniques like meta-learning [19], regularization [24, 35, 67], personalized-head methods [2, 10, 15, 59], and other innovative approaches [36, 37, 76] are widely employed in pFL. In essence, both traditional FL and pFL aim to grapple with the inherent challenge of data heterogeneity.

### 2.2. Multi-Task Learning

Multi-Task Learning (MTL) aims to improve overall performance while reducing parameters and speeding up training or inference compared to training individual models for each task in isolation [9, 16, 60, 70]. The main directions of MTL research can be roughly categorized into network architecture design and multi-task optimization strategy [16]. Network structure design employs methods such as parameter sharing [28, 47, 50, 61, 66], task interaction [44, 64, 78, 86, 87, 89], and prediction distillation [69, 81, 82, 88]. Regarding multi-task optimization,

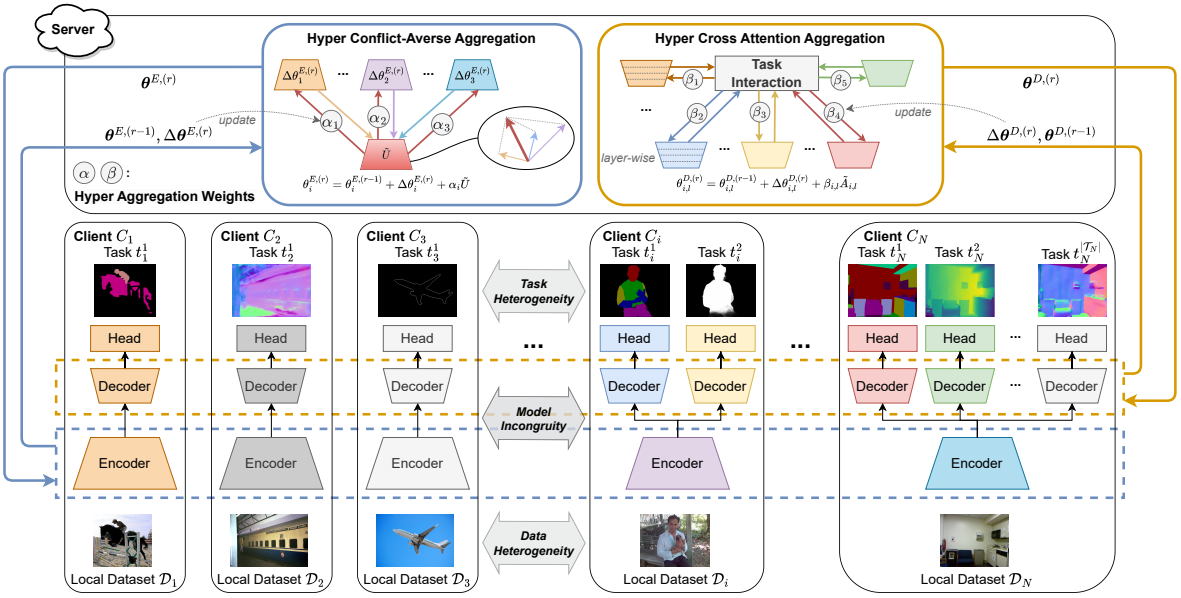


Figure 2. Illustration of the HC-FMTL setting and our proposed FEDHCA$^2$ framework. HC-FMTL enables clients to have different task setups, from single-task (*e.g.* client $C_1, C_2, C_3$) to multi-task (*e.g.* client $C_i, C_N$). HC-FMTL faces three main challenges: model incongruity due to different client model structures, data heterogeneity from different local data domains, and task heterogeneity from varied target tasks. The FL system includes a server and several clients. Our framework decomposes model aggregation into two parts: Hyper Conflict-Averse Aggregation for encoders and Hyper Cross Attention Aggregation for decoders. Learnable Hyper Aggregation Weights are employed to customize personalized parameter updates and are iteratively updated by local model updates from clients.

strategies are differentiated into loss balancing and gradient balancing. Loss balancing techniques are designed to produce suitable loss weights to reduce conflicts among multiple tasks [30, 40, 41, 80]. Gradient balancing, on the other hand, addresses task interference by directly adjusting gradients, with recent methods concentrating on the formulation of a unified gradient vector subject to diverse constraints [13, 14, 26, 38, 62, 75, 84]. In essence, MTL is dedicated to addressing the intrinsic challenges associated with the heterogeneity of tasks.

### 2.3. Federated Multi-Task Learning

It is essential to note that conventional Federated Multi-Task Learning (FMTL) is a branch of personalized Federated Learning that primarily deals with data heterogeneity across clients [22, 39, 54]. Representative works like MOCHA [65] and FedEM [51] attempt to train models across clients with diverse data distributions within an MTL setting. Recent advancements, including FedBone [12], MAS [92], and MaT-FL [8], have aimed to address both task and data heterogeneity in FMTL. FedBone aggregates the encoders by gradients uploaded from each client, enhancing feature extraction capability. MaT-FL uses dynamic grouping to combine different client models. MAS distributes varied multi-task models to clients and aggregates models among those with the same task sets. Nevertheless, FedBone and MaT-FL are limited to each client managing a single task (Fig. 1a). MAS supports multi-task clients but is still limited to identical task sets for aggregation (Fig. 1b). In contrast, our proposed framework enables aggregation across clients with varying numbers and types of tasks, offering a more flexible collaboration.

## 3. Methodology

### 3.1. Preliminary

Within Hetero-Client Federated Multi-Task Learning (HC-FMTL), clients are assigned flexible task setups, spanning from single-task to multi-task configurations, with an arbitrary number of tasks per client. Formally, given a pool of $N$ clients, with client $C_i$ addressing task sets $\mathcal{T}_i$ on a corresponding local dataset $\mathcal{D}_i = \{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^{|\mathcal{D}_i|}$, where $\mathbf{x}_n$ is the input sample and $\mathbf{y}_n = \bigcup_{t\in\mathcal{T}_i} \mathbf{y}_{n,t}$ contains the ground-truth labels for all tasks in $\mathcal{T}_i$.

In line with FMTL, the objective of HC-FMTL is to train client-specific models $\boldsymbol{\theta} = \{\theta_1, \dots, \theta_N\}$ that benefit from collaborative optimization with other clients, thus improving performance on their local tasks. The learning objective is to optimize personalized client models with Multi-Task Learning, formulated as follows:

$$\min_{\theta_i} \sum_{t\in\mathcal{T}_i} \mathcal{L}_{i,t}(\theta_i), \quad \forall i \in \{1, \dots, N\}, \tag{1}$$

where $\mathcal{L}_{i,t}$ is the loss function computed over client $C_i$'s local dataset $\mathcal{D}_i$ for task $t$.

### 3.2. Architecture Overview

The overall architecture of our proposed FEDHCA$^2$ is depicted in Fig. 2. It contains a pool of clients that perform local training on their private datasets and a server that coordinates the aggregation of models from these clients. Concerning dense prediction tasks, each client $C_i$ utilizes an encoder-decoder structure consisting of a shared encoder $\theta_i^E$, task-specific decoders $\{\theta_i^{D,1}, \ldots, \theta_i^{D,|\mathcal{T}_i|}\}$ and prediction heads for each task type they handle. In each communication round $r$, after all clients finish their local training, they send the model parameters of previous round $\boldsymbol{\theta}^{(r-1)}$ and the updates in current round $\Delta\boldsymbol{\theta}^{(r)}$ to the server. The server first disassembles these models into encoders and decoders and then performs independent aggregation processes. The prediction heads, due to their varying parameter dimensions tailored to specific task outputs, are excluded from the aggregation process and remain localized to individual clients. The encoder parameters from all $N$ clients undergo Hyper Conflict-Averse Aggregation. Meanwhile, the server aggregates the parameters of all $K = \sum_{i=1}^{N} \mathcal{T}_i$ decoders through Hyper Cross Attention Aggregation. The entire pipeline of our framework is outlined in Algorithm 1.

**Algorithm 1** Pseudo-codes for FEDHCA$^2$

**Input:** $N$ clients $\{C_1, \ldots, C_N\}$ with private local datasets $\{\mathcal{D}_1, \ldots, \mathcal{D}_N\}$, client $C_i$ addresses tasks $\mathcal{T}_i$, total communication rounds $R$, local epoch $E$, learning rate $\eta$
**Output:** Trained models $\boldsymbol{\theta}^{(R)} = \{\theta_1^{(R)}, \ldots, \theta_N^{(R)}\}$

Clients initialize models $\boldsymbol{\theta}^{(0)} = \{\theta_1^{(0)}, \ldots, \theta_N^{(0)}\}$, each model $\theta_i$ consists of a shared encoder $\theta_i^E$ and $|\mathcal{T}_i|$ task-specific decoders $\bigcup_{j=1}^{|\mathcal{T}_i|} \theta_i^{D,j}$ and heads
Server initializes Hyper Aggregation Weights $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$
**procedure** SERVER UPDATE
  **for** each communication round $r \in \{1, \ldots, R\}$ **do**
    **for** each client $C_i$ in parallel **do**
      $\Delta\theta_i^{(r)} \leftarrow$ CLIENT UPDATE$(\theta_i^{(r-1)})$
    **end for**
    Server gathers updates of client models $\Delta\boldsymbol{\theta}^{(r)}$
    Update $\boldsymbol{\alpha}, \boldsymbol{\beta}$ using $\Delta\boldsymbol{\theta}^{(r)}$ with Eq. (17)
    $\boldsymbol{\theta}^{(r)} \leftarrow$ AGGREGATION$(\boldsymbol{\theta}^{(r-1)}, \Delta\boldsymbol{\theta}^{(r)})$
  **end for**
**end procedure**
**procedure** CLIENT UPDATE$(\theta_i^{(r-1)})$
  $\theta_i \leftarrow \theta_i^{(r-1)}$
  **for** each local epoch $e \in \{1, \ldots, E\}$ **do**
    **for** mini-batch $\mathcal{B}_i \subset \mathcal{D}_i$ **do**
      Compute losses $\mathcal{L}_i = \sum_{j=1}^{|\mathcal{T}_i|} \mathcal{L}_i^j(\theta_i; \mathcal{B}_i)$
      Update model $\theta_i \leftarrow \theta_i - \eta \nabla_{\theta_i} \mathcal{L}_i$
    **end for**
  **end for**
  **return** $\Delta\theta_i^{(r)} = \theta_i - \theta_i^{(r-1)}$
**end procedure**

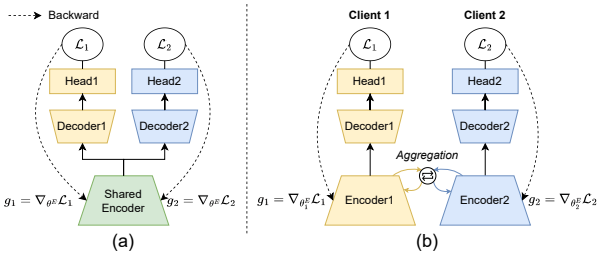


Figure 3. Comparison of optimization in MTL and FL. (a) The shared encoder in MTL is updated by gradient accumulation from all tasks. (b) The clients' encoders are updated independently and then aggregated in FL.

### 3.3. Hyper Conflict-Averse Aggregation

In MTL, the encoder typically employs a parameter-sharing mechanism to capture common task-agnostic information, thereby serving as a general feature extractor for all tasks and enhancing their generalization capabilities. Within our encoder aggregation process, we anticipate that encoders from various clients, each addressing distinct tasks on different data domains, are able to acquire general knowledge from other client encoders akin to MTL. To elucidate this, we begin with a theoretical analysis of the correlation between the optimization process of MTL and FL.

As depicted in Fig. 3a, consider an MTL scenario where $N$ tasks are learned simultaneously using a standard multi-decoder architecture. In each mini-batch, the network back-propagates the loss functions $\mathcal{L}_1, \ldots, \mathcal{L}_N$ onto the shared encoder $\theta^E$ to calculate its gradient and update:

$$g = \sum_{i=1}^{N} g_i = \sum_{i=1}^{N} \nabla_{\theta^E} \mathcal{L}_i, \tag{2}$$

$$\Delta\theta^E = -\eta g = -\eta \sum_{i=1}^{N} g_i, \tag{3}$$

where $g$ represents the cumulative gradient on the encoder and $\eta$ signifies the learning rate. By updating through the summation of gradients from diverse tasks, the encoder assimilates knowledge from various task domains, aligning with the objective of MTL. Meanwhile, in an FL setting shown in Fig. 3b, suppose there are $N$ clients, each using separate networks with the same architectures as the multi-task model but with independent encoders $\theta_1^E, \ldots, \theta_N^E$ to learn the same $N$ tasks. Assuming identical initial weights $\theta^E$ with MTL, and they are trained for only one mini-batch to obtain gradients $g_i = \nabla_{\theta^E} \mathcal{L}_i$. FL typically aggregates these encoders by averaging the parameters of all clients:

$$\tilde{\theta}_i^E = \frac{1}{N} \sum_{i=1}^{N} \theta_i^E, \tag{4}$$

where $\tilde{\theta}_i^E$ is the aggregated encoder parameters. Consider-

ing its change from the initial weight:

$$\Delta\tilde{\theta}_i^E = \frac{1}{N}\sum_{i=1}^{N}\Delta\theta_i^E = \frac{1}{N}(-\eta)\sum_{i=1}^{N} g_i, \tag{5}$$

it means the update for the aggregated encoder mirrors the update of the shared encoder in MTL, if we regard the optimizer as capable of automatically scaling the learning rate $\eta$ in Eq. (3). While FL typically aggregates client models after several local training epochs in a communication round, this implies that there are differences between the learning processes of MTL and FL:

**Theorem 1 (Difference in optimizing MTL and FL)** *Given clients with a shared encoder and task-specific decoder structure, the gradient descent in the shared encoder of MTL is equivalent to averaging parameter aggregation in FL, adding an extra term that maximizes the inner product of gradients between all pairs of tasks in each iteration.*

We provide proofs and in-depth analysis in Appendix A. As the inner product of gradients is a measurement of accordance, maximizing the inner product is equal to reducing the conflict of gradients [48]. Hence, Theorem 1 states the necessity of integrating optimization techniques to mitigate gradient conflicts during encoder aggregation in HC-FMTL. Inspired by CAGrad [38], for each communication round, we aim to find an optimal aggregated update $\tilde{U}$ for the encoder that minimizes conflicts while optimizing the main objective with optimization problem:

$$\max_{\tilde{U}} \min_i \langle \Delta\theta_i^E, \tilde{U}\rangle \quad \text{s.t.} \|\tilde{U} - \Delta\bar{\theta}^E\| \leq c\|\Delta\bar{\theta}^E\|, \tag{6}$$

where $\Delta\bar{\theta}^E = \frac{1}{N}\sum_{i=1}^{N}\Delta\theta_i^E$ is the average parameter update and $c \in [0, 1)$ is a hyper-parameter controlling the convergence rate. Here $\min_i \langle \Delta\theta_i^E, \tilde{U}\rangle$ measures the maximum conflict between client updates and the target update, which is an approximation to the conflict between gradients, as the server only receives parameter updates after several local training epochs rather than the gradients in each iteration. Therefore, maximizing this term can minimize the conflict in parameter optimization, which is consistent with our findings in Theorem 1. With constraint $\sum_{i=1}^{N} w_i = 1, w_i \geq 0$, solving this problem using Lagrangian simplifies to:

$$\min_w F(w) = U_w^\top \Delta\bar{\theta}^E + \sqrt{\phi}\|U_w\|, \tag{7}$$

$$\text{where } U_w = \frac{1}{N}\sum_{i=1}^{N} w_i \Delta\theta_i^E, \phi = c^2\|\Delta\bar{\theta}^E\|^2. \tag{8}$$

Upon finding the optimum $w^*$ and the optimal $\lambda^* = ||U_{w^*}|| / \phi^{1/2}$, we have the unified aggregated update:

$$\tilde{U} = \Delta\bar{\theta}^E + U_{w^*}/\lambda^* = \Delta\bar{\theta}^E + \frac{\sqrt{\phi}}{\|U_w\|}U_w. \tag{9}$$

### 3.4. Hyper Cross Attention Aggregation

The significance of task interaction in MTL is well-established [7, 20, 55, 69, 77], as it allows for exchanging knowledge among tasks and benefiting from complementary information. In representative methods [55, 77], task interaction is facilitated by adding the target task's feature with those from source tasks in decoders, formulated as:

$$\mathbf{z}_i^D = \sum_{j=1}^{N} \gamma_{i,j}(\theta_j^D)^\top \mathbf{z}^E, \quad \forall i \in \{1, \ldots, N\}, \tag{10}$$

where $\mathbf{z}^E$ denotes the output feature of the shared encoder, $\theta_j^D$ represents the decoder of task $j$, and $(\theta_j^D)^\top \mathbf{z}^E$ yields task-specific feature from the decoder. The coefficient $\gamma_{i,j}$ manages the flow of features from source to target tasks within the interaction and is usually a learnable parameter. To emulate this task interaction within the FL context, we intuitively aggregate the decoder parameters as follows:

$$\tilde{\theta}_i^D = \sum_{j=1}^{N} \gamma_{i,j}\, \theta_j^D. \tag{11}$$

Due to model incongruity in the HC-FMTL environment, the decoder parameters sent to the server originate from diverse clients with heterogeneous tasks. This intricacy leads to a complex landscape where decoders may align or diverge in both data domain and task type, requiring the aggregation process to discern the nuanced relationships among them. Our approach improves the decoder aggregation by adopting a cross attention mechanism to further promote the exchange of inter-task knowledge among clients with model incongruity. It calculates dependencies among the local updates of $K$ decoders, thereby modeling the interplay among tasks. Recognizing that decoders often exhibit varied utilities across different network layers [6, 21, 46, 71], we apply a layer-wise strategy [49] to precisely capture the cross-task attention at each decoder layer, allows for a more fine-grained personalized aggregation that can benefit the transfer of task-specific knowledge. The computation of cross attention is defined as:

$$V_l = [\Delta\theta_{1,l}^D, \ldots, \Delta\theta_{K,l}^D]^\top, \tag{12}$$

$$\tilde{A}_{i,l} = \text{Softmax}(\Delta\theta_{i,l}^D\, V_l^\top / \sqrt{d})V_l, \tag{13}$$

where $[\cdot,\cdot]$ indicates concatenation, $\Delta\theta_{i,l}$ and $\tilde{A}_{i,l}$ are the original update and aggregated update for the $l$-th layer of the $i$-th decoder, with a dimension of $d$.

### 3.5. Hyper Aggregation Weights

As pointed out by pFL, a unified update for all clients is restricted in addressing client heterogeneity. Hence, we propose Hyper Aggregation Weights, which adaptively assess the importance of the aggregated parameters from peers and empower clients with analogous data domains and task ob-

Table 1. Comparison to representative methods using PASCAL-Context for five **S**ingle-**T**ask clients and NYUD-v2 for one **M**ulti-**T**ask client. '↑' means higher is better and '↓' means lower is better. '$\Delta_m\%$' denotes the average performance drop w.r.t. local baseline.

| Method | PASCAL-Context (**ST**) | | | | | NYUD-v2 (**MT**) | | | | $\Delta_m\% \uparrow$ |
|---|---|---|---|---|---|---|---|---|---|---|
| | SemSeg mIoU↑ | Parts mIoU↑ | Sal maxF↑ | Normals mErr↓ | Edge odsF↑ | SemSeg mIoU↑ | Depth RMSE↓ | Normals mErr↓ | Edge odsF↑ | |
| Local | 51.69 | 49.94 | **80.91** | 15.76 | 71.95 | 41.86 | 0.6487 | 20.59 | 76.46 | 0.00 |
| FedAvg [53] | 39.98 | 37.33 | 77.56 | 18.27 | 69.17 | 38.94 | 0.7858 | 21.62 | 75.77 | -11.76 |
| FedProx [34] | 44.42 | 38.10 | 77.26 | 18.03 | 69.39 | 39.19 | 0.8068 | 21.52 | 76.03 | -10.68 |
| FedPer [2] | 54.51 | 46.56 | 78.85 | 16.95 | 71.00 | **44.02** | 0.6467 | 21.19 | **76.61** | -1.11 |
| Ditto [35] | 46.23 | 39.69 | 77.99 | 17.52 | 69.77 | 41.49 | 0.6508 | 20.60 | 76.45 | -5.57 |
| FedAMP [24] | 55.98 | 52.05 | 80.79 | 15.74 | 72.02 | 41.67 | 0.6428 | 20.54 | 76.40 | 1.47 |
| MaT-FL [8] | 57.45 | 48.63 | 79.26 | 17.26 | 71.23 | 40.99 | 0.6352 | 20.65 | 76.59 | -0.46 |
| FedHCA$^2$ | **57.55** | **52.30** | 80.71 | **15.60** | **72.08** | 41.47 | **0.6281** | **20.53** | 76.50 | **2.18** |

jectives to have higher aggregation weights. This enhancement reinforces the mutual contribution from complementary information, thus serving as high-level guidance in harmonizing the local updates with the collaborative updates. Specifically, the server maintains a dedicated set of weights for each client, which are applied as follows in the personalized aggregation:

$$\theta_i^{(r)} = \theta_i^{(r-1)} + \Delta\theta_i^{(r)} + \psi_i\tilde{\theta}_i, \tag{14}$$

where $\psi_i$ denotes the hyper weights for client $C_i$, *i.e.* $\alpha_i$ for encoder or $\beta_i$ for decoder, and $\tilde{\theta}_i$ is the aggregated update $\tilde{U}$ from Eq. (9) or $\tilde{A}_{i,l}$ from Eq. (13). It is worth noting that we implement distinct weights for each decoder layer rather than a single weight value to be consistent with the layer-wise computation of cross attention.

Furthermore, we design Hyper Aggregation Weights to be learnable parameters that are dynamically updated throughout the training phase. This adaptability ensures that the weights are optimized in conjunction with the system's overall objective. By employing the chain rule, we can derive the gradient of $\psi_i$ as follows:

$$\nabla_{\psi_i}\mathcal{L}_i = (\nabla_{\psi_i}\theta_i^{(r)})^\top \nabla_{\theta_i^{(r)}}\mathcal{L}_i = (\tilde{\theta}_i)^\top \nabla_{\theta_i^{(r)}}\mathcal{L}_i. \tag{15}$$

To better align this update rule with the FL paradigm, we can reformulate Eq. (15) by substituting gradients with model updates, which is the negative accumulation of gradients over batches:

$$\Delta\alpha_i = (\tilde{U}^{(r)})^\top \Delta\theta_i^{E,(r)}, \tag{16}$$

$$\Delta\beta_{i,l} = (\tilde{A}_{i,l}^{(r)})^\top \Delta\theta_{i,l}^{D,(r)}. \tag{17}$$

It indicates that the update of Hyper Aggregation Weights can be attained by the alteration in model parameters following local training in subsequent communication rounds.

# 4. Experiments

## 4.1. Experimental Setup

**Datasets.** We conduct experiments with two established benchmark datasets for multi-task dense prediction: PASCAL-Context [56] and NYUD-v2 [63]. The PASCAL-Context dataset contains 4,998 images for training and 5,105 for testing, annotated for five tasks: edge detection ('Edge'), semantic segmentation ('SemSeg'), human parts segmentation ('Parts'), surface normal estimation ('Normals'), and saliency detection ('Sal'). The NYUD-v2 dataset consists of 795 training images and 654 testing images, all depicting indoor scenes, and provides annotations for four tasks: edge detection, semantic segmentation, surface normal estimation, and depth estimation ('Depth').

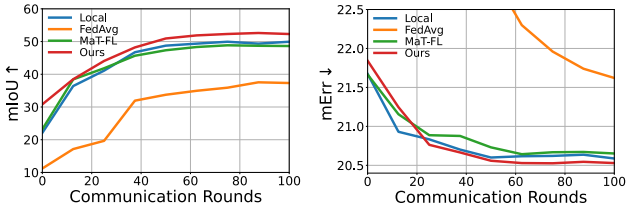


Figure 4. Evaluation results during training. (a) Parts from PASCAL-Context on single-task client. (b) Normals from NYUD-v2 on multi-task client.

To evaluate our algorithm, we configure two HC-FMTL benchmark scenarios: 1) Five single-task clients address five tasks in PASCAL-Context, and one multi-task client addresses four tasks in NYUD-v2; 2) Conversely, four single-task clients address four tasks in NYUD-v2, and one multi-task client addresses five tasks in PASCAL-Context. Following MaT-FL [8], we set an equal number of data samples among the respective clients through random partitioning.

**Implementation.** Our client architecture employs a pre-trained Swin-T [43] backbone coupled with simple FCN decoders and heads. Considering the varying capacities of datasets, we use one local epoch for PASCAL-Context and four for NYUD-v2, setting the total number of communication rounds to 100 and the batch size to 8. We train all models using AdamW optimizer [45] with an initial learning rate and weight decay rate set at 1e-4. We implement all methods with PyTorch [57] and run experiments on two NVIDIA RTX4090 GPUs. To adapt existing methods to the HC-FMTL setting, we decouple the models into encoders and decoders for separate aggregation across all methods.

**Metrics.** We adhere to established evaluation metrics. Specifically, we measure semantic segmentation and human parts segmentation using the mean Intersection over

Table 2. Comparison to representative methods using NYUD-v2 for four single-task clients and PASCAL-Context for one multi-task client.

| Method | NYUD-v2 (**ST**) | | | | PASCAL-Context (**MT**) | | | | | $\Delta_m\% \uparrow$ |
|---|---|---|---|---|---|---|---|---|---|---|
| | SemSeg mIoU↑ | Depth RMSE↓ | Normals mErr↓ | Edge odsF↑ | SemSeg mIoU↑ | Parts mIoU↑ | Sal maxF↑ | Normals mErr↓ | Edge odsF↑ | |
| Local | 33.59 | 0.7129 | 23.22 | 75.02 | 65.80 | 55.01 | 83.23 | 14.21 | 71.89 | 0.00 |
| FedAvg [53] | 25.80 | 0.8295 | 24.85 | 75.31 | 64.63 | 52.88 | 81.08 | 15.56 | 68.95 | -7.56 |
| FedProx [34] | 25.96 | 0.8316 | 25.20 | 75.34 | 64.97 | 50.78 | 81.29 | 15.83 | 69.81 | -8.12 |
| FedPer [2] | **35.93** | 0.7460 | 23.75 | **75.53** | 67.78 | 54.75 | 82.50 | 14.75 | 71.90 | -0.16 |
| Ditto [35] | 28.15 | 0.7482 | 23.96 | 75.42 | 65.99 | 51.45 | 81.74 | 15.29 | 69.96 | -4.67 |
| FedAMP [24] | 34.75 | 0.7103 | 23.31 | 75.03 | 66.08 | 54.10 | **83.35** | 14.20 | 71.88 | 0.27 |
| MaT-FL [8] | 35.05 | 0.7504 | 23.39 | 75.33 | **67.90** | 54.78 | 82.84 | 14.58 | 71.94 | -0.16 |
| FEDHCA$^2$ | 34.95 | **0.7018** | **23.19** | 75.03 | 65.81 | **55.01** | 83.18 | **14.08** | **71.97** | **0.75** |

Table 3. Ablation study on our proposed aggregation schemes. '+Enc' and '+Dec' denote the integration of Hyper Conflict-Averse Aggregation for the encoders and Hyper Cross Attention Aggregation for the decoders, respectively.

| Method | PASCAL-Context (**ST**) | | | | | NYUD-v2 (**MT**) | | | | $\Delta_m\% \uparrow$ |
|---|---|---|---|---|---|---|---|---|---|---|
| | SemSeg↑ | Parts↑ | Sal↑ | Normals↓ | Edge↑ | SemSeg↑ | Depth↓ | Normals↓ | Edge↑ | |
| Local | 51.69 | 49.94 | **80.91** | 15.76 | 71.95 | **41.86** | 0.6487 | 20.59 | 76.46 | 0.00 |
| +Enc | **58.38** | 51.64 | 80.44 | 15.65 | **72.09** | 41.21 | 0.6377 | 20.55 | 76.50 | 1.89 |
| +Dec | 57.39 | 51.65 | 80.75 | 15.69 | 72.06 | 41.48 | 0.6344 | 20.56 | 76.41 | 1.80 |
| +Enc+Dec | 57.55 | **52.30** | 80.71 | **15.60** | 72.08 | 41.47 | **0.6281** | **20.53** | **76.50** | **2.18** |

Union (mIoU). Saliency detection is evaluated with the maximum F-measure (maxF), while surface normal estimation is assessed by the mean error (mErr). Edge detection utilizes the optimal-dataset-scale F-measure (odsF), and depth estimation uses the Root Mean Square Error (RMSE). To provide an overall evaluation of different algorithms, we calculate the average per-task performance drop [50] relative to the local training baseline, which is trained without aggregation. The formula is as follows: $\Delta_m = \frac{1}{N}\sum_{i=1}^{N}(-1)^{l_i}\frac{M_{\texttt{Fed},i}-M_{\texttt{Local},i}}{M_{\texttt{Local},i}}$, where $N$ is the count of tasks, $M_{\texttt{Fed},i}$ and $M_{\texttt{Local},i}$ correspond to the performance of task $i$ for federated methods and the local baseline, respectively. $l_i = 1$ if a lower metric value is better for task $i$, and $l_i = 0$ otherwise.

## 4.2. Main Results

To evaluate the performance of our method, we compare with representative works including two traditional FL approaches FedAvg [53] and FedProx [34], three pFL methods FedPer [2], Ditto [35], FedAMP [24], and one FMTL method MaT-FL [8]. The results presented in Tab. 1 and Tab. 2 demonstrate that FEDHCA$^2$ consistently delivers the best performance across most metrics. More importantly, it outperforms all representative methods when considering the average per-task performance drop, which is a widely acknowledged indicator for assessing the overall performance of MTL. In addition, Fig. 4 shows that FEDHCA$^2$ converges faster to a better result on different tasks.

## 4.3. Indepth Analysis

**Ablation Study.** An ablation study is conducted to discern the individual contributions of each component within FEDHCA$^2$, as shown in Tab. 3. The results indicate that incorporating either encoder or decoder aggregation enhances performance relative to the baseline. The simultaneous employment of both Hyper Conflict-Averse and Hyper Cross Attention Aggregations enables FEDHCA$^2$ to achieve optimal performance across the evaluated configurations. This result supports the idea that using these two aggregation schemes together enhances cooperation among different clients while simultaneously reducing negative conflicts between various tasks.

Table 4. Comparison between different settings. 'ST+Local' and 'ST+Ours' denote the setting with four single-task clients on NYUD-v2, trained with local baseline and FEDHCA$^2$, respectively. 'ST+MT+Ours' denotes the setting in Tab. 2 trained with our framework. '$\Delta_m$' is calculated w.r.t. 'ST+Local' baseline.

| Setting | SemSeg ↑ | Depth ↓ | Normals ↓ | Edge ↑ | $\Delta_m\% \uparrow$ |
|---|---|---|---|---|---|
| ST+Local | 33.59 | 0.7129 | 23.22 | 75.02 | 0.00 |
| ST+Ours | 34.71 | 0.7170 | 23.25 | 74.98 | 0.64 |
| ST+MT+Ours | **34.95** | **0.7018** | **23.19** | **75.03** | **1.44** |

Table 5. Comparison to local baseline on the setting with only multi-task clients on two datasets.

| Method | PASCAL-Context (**MT**) | | | NYUD-v2 (**MT**) | | $\Delta_m\% \uparrow$ |
|---|---|---|---|---|---|---|
| | SemSeg↑ | Parts↑ | Normals↓ | SemSeg↑ | Normals↓ | |
| Local | **64.87** | 53.34 | 14.07 | 39.81 | 20.65 | 0 |
| Ours | 64.17 | **54.25** | **14.01** | **40.26** | **20.55** | **0.53** |

**Impact of different FMTL scenarios.** To further verify the necessity of introducing our new setting, we conduct experiments comparing two scenarios: 1) each client handles a single task, and 2) HC-FMTL encompasses both single-task and multi-task clients. As Tab. 4 illustrates, while FEDHCA$^2$ improves upon the local baseline in the single-task client scenario, integrating the multi-task client results in a greater enhancement. This improvement is attributed to the expanded pool of data and the knowledge jointly learned from additional tasks. Further experiments are carried out on another scenario of HC-FMTL setting which exclusively involves multi-task clients. Specifically, we select three tasks from PASCAL-Context and two tasks from NYUD-v2 to create two multi-task client setups. The out-

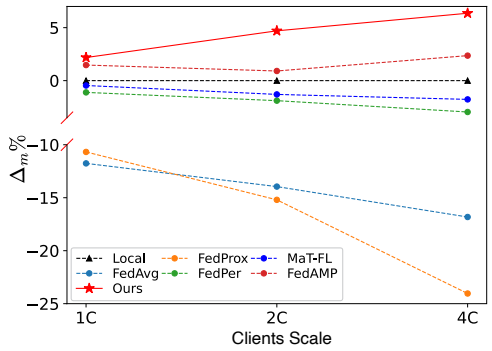


Figure 5. The performance changes of different methods with the number of clients scaling to 2 and 4 times. '$\Delta_m$' is calculated w.r.t. corresponding local baseline of 1C, 2C, or 4C. When the number of clients increases, our method can consistently provide superior performance, and an overall growth trend could be observed.

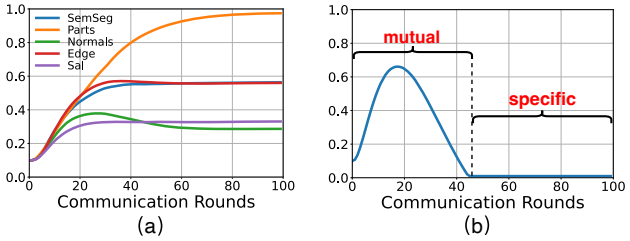


Figure 6. Hyper Aggregation Weights $\alpha$ for encoders of the client models. (a) Weights of five single-task clients. (b) Weights of the multi-task client which differs in two stages.

comes presented in Tab. 5 align with our primary findings in Sec. 4.2 that nearly all metrics surpass the local baseline, further confirming the efficacy of our approach in this specialized setting.

**Impact of the number of clients.** To assess the effectiveness of FEDHCA$^2$ across varying client counts, we conduct tests by scaling the number of clients per task by factors of 2 and 4, with the datasets evenly split. As depicted in Fig. 5, FEDHCA$^2$ consistently outperforms all comparative methods, exhibiting a positive correlation between the number of clients and performance improvement. This trend contrasts with the performance decline seen with other methods as the client count increases—a result typically attributed to the diminished dataset available to each client and the increased decentralization within the federated learning system. The success of FEDHCA$^2$ substantiates the efficacy of the Hyper Conflict-Averse Aggregation and Hyper Cross Attention Aggregation schemes, especially in scenarios characterized by pronounced data and task heterogeneity.

**Interaction between tasks.** We investigate the dynamic learning process of Hyper Aggregation Weights for both encoders and decoders, aiming to understand their role in facilitating personalized aggregation for different clients. Fig. 6a reveals that the evolution of weights for encoders in single-task clients shows a rising trend, suggesting a consistent uptake of knowledge from peers throughout the training period. In contrast, the encoder weight of the multi-task client, as depicted in Fig. 6b, exhibits two stages. Initially, the multi-task client mutually assimilates knowledge from single-task clients, a process that is crucial for rapid model convergence. The mutual learning for the multi-task client reaches its peak at about 20 rounds when the encoder weights are comparable. Subsequently, in the second phase, due to the heterogeneity in data and tasks, the multi-task client tends to enhance its feature extraction capabilities specific to its own data domain.

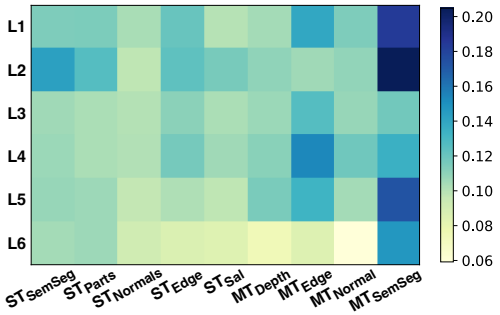


Figure 7. Learned Hyper Aggregation Weights $\beta$ across decoders for different tasks, spanning layers from L1 to L6.

Weights for decoders, as shown in Fig. 7, vary significantly across different tasks and decoder layers. From a layer-oriented perspective, the layer closest to the output head, *i.e.*, L6, depends least on cross-task information, which ensures that the final output is finely tuned to the specific task. In terms of task-related differences, a phenomenon markedly distinct from encoders is observed. For decoders of multi-task client, there is a persistent information integration from other tasks until the end of training. This empirical evidence substantiates the significance of employing task interaction in decoder aggregation.

## 5. Conclusion

In conclusion, this paper addresses the challenges of heterogeneity in the novel Hetero-Client Federated Multi-Task Learning (HC-FMTL) setting through the novel FEDHCA$^2$ framework. By recognizing and tackling the issues of model incongruity, data heterogeneity, and task heterogeneity, FEDHCA$^2$ learns personalized models with synergies of the proposed Hyper Conflict-Averse Aggregation, Hyper Cross Attention Aggregation, and Hyper Aggregation Weights. Theoretical insights and extensive experiments confirm the effectiveness of our methodology. Our work opens possibilities for more flexible FL systems in diverse and realistic settings. For future work, we aim to delve into greater model heterogeneity that accommodates varied network structures across clients, and to integrate specific modules into multi-task clients to further enhance task interaction, drawing on advancements in MTL.

## References


[1] Durmus Alp Emre Acar, Yue Zhao, Ramon Matas, Matthew Mattina, Paul Whatmough, and Venkatesh Saligrama. Federated learning based on dynamic regularization. In *ICLR*, 2021. 2

[2] Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. Federated learning with personalization layers. *CoRR*, abs/1912.00818, 2019. 2, 6, 7

[3] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In *AISTATS*, pages 2938–2948, 2020. 2

[4] Xiang Bai, Hanchen Wang, Liya Ma, Yongchao Xu, et al. Advancing COVID-19 diagnosis with privacy-preserving collaboration in artificial intelligence. *Nature Machine Intelligence*, 3(12):1081–1089, 2021. 1

[5] Cosmin I. Bercea, Benedikt Wiestler, Daniel Rueckert, and Shadi Albarqouni. Federated disentangled representation learning for unsupervised brain anomaly detection. *Nature Machine Intelligence*, 4(8):685–695, 2022. 1

[6] David Brüggemann, Menelaos Kanakis, Stamatios Georgoulis, and Luc Van Gool. Automated search for resource-efficient branched multi-task networks. In *BMVC*, page 359, 2020. 5

[7] David Brüggemann, Menelaos Kanakis, Anton Obukhov, Stamatios Georgoulis, and Luc Van Gool. Exploring relational context for multi-task dense prediction. In *ICCV*, pages 15869–15878, 2021. 5

[8] Ruisi Cai, Xiaohan Chen, Shiwei Liu, Jayanth Srinivasa, Myungjin Lee, Ramana Kompella, and Zhangyang Wang. Many-task federated learning: A new problem setting and a simple baseline. In *CVPR*, pages 5037–5045, 2023. 1, 3, 6, 7

[9] Rich Caruana. Multitask learning. *Machine learning*, 28(1): 41–75, 1997. 2

[10] Hong-You Chen and Wei-Lun Chao. On bridging generic and personalized federated learning for image classification. In *ICLR*, 2022. 2

[11] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, pages 801–818, 2018. 1

[12] Yiqiang Chen, Teng Zhang, Xinlong Jiang, Qian Chen, Chenlong Gao, and Wuliang Huang. Fedbone: Towards large-scale federated multi-task learning. *arXiv preprint arXiv:2306.17465*, 2023. 1, 3

[13] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *ICML*, pages 794–803, 2018. 3

[14] Zhao Chen, Jiquan Ngiam, Yanping Huang, Thang Luong, Henrik Kretzschmar, Yuning Chai, and Dragomir Anguelov. Just pick a sign: Optimizing deep multitask models with gradient sign dropout. In *NeurIPS*, 2020. 3

[15] Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Exploiting shared representations for personalized federated learning. In *ICML*, pages 2089–2099, 2021. 2

[16] Michael Crawshaw. Multi-task learning with deep neural networks: A survey. *arXiv preprint arXiv:2009.09796*, 2020. 2

[17] Ittai Dayan, Holger R. Roth, Aoxiao Zhong, Ahmed Harouni, Amilcare Gentili, et al. Federated learning for predicting clinical outcomes in patients with COVID-19. *Nature Medicine*, 27(10):1735–1743, 2021. 1

[18] Enmao Diao, Jie Ding, and Vahid Tarokh. Heterofl: Computation and communication efficient federated learning for heterogeneous clients. In *ECCV*, 2021. 2

[19] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. *NeurIPS*, 33:3557–3568, 2020. 2

[20] Yuan Gao, Jiayi Ma, Mingbo Zhao, Wei Liu, and Alan L Yuille. Nddr-cnn: Layerwise feature fusing in multi-task cnns by neural discriminative dimensionality reduction. In *CVPR*, pages 3205–3214, 2019. 5

[21] Pengsheng Guo, Chen-Yu Lee, and Daniel Ulbricht. Learning to branch for multi-task learning. In *ICML*, pages 3854–3863, 2020. 5

[22] Chaoyang He, Emir Ceyani, Keshav Balasubramanian, Murali Annavaram, and Salman Avestimehr. Spreadgnn: Decentralized multi-task federated learning for graph neural networks on molecular data. In *AAAI*, pages 6865–6873, 2022. 1, 3

[23] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *CVPR*, pages 17853–17862, 2023. 1

[24] Yutao Huang, Lingyang Chu, Zirui Zhou, Lanjun Wang, Jiangchuan Liu, Jian Pei, and Yong Zhang. Personalized cross-silo federated learning on non-iid data. In *AAAI*, pages 7865–7873, 2021. 2, 6, 7

[25] Yangsibo Huang, Samyak Gupta, Zhao Song, Kai Li, and Sanjeev Arora. Evaluating gradient inversion attacks and defenses in federated learning. *NeurIPS*, 34:7232–7241, 2021. 2

[26] Adrián Javaloy and Isabel Valera. Rotograd: Gradient homogenization in multitask learning. In *ICLR*, 2022. 3

[27] Peter Kairouz, H. Brendan McMahan, Brendan Avent, et al. Advances and open problems in federated learning. *Found. Trends Mach. Learn.*, 14(1-2):1–210, 2021. 1

[28] Menelaos Kanakis, David Bruggemann, Suman Saha, Stamatios Georgoulis, Anton Obukhov, and Luc Van Gool. Reparameterizing convolutions for incremental multi-task learning without task interference. In *ECCV*, pages 689–707, 2020. 2

[29] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J. Reddi, Sebastian U. Stich, and Ananda Theertha Suresh. SCAFFOLD: stochastic controlled averaging for federated learning. In *ICML*, pages 5132–5143, 2020. 2

[30] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *CVPR*, pages 7482–7491, 2018. 3

[31] Jakub Konečný, Brendan McMahan, and Daniel Ramage. Federated optimization: Distributed optimization beyond the datacenter. *CoRR*, abs/1511.03575, 2015. 1

[32] Jakub Konečný, H. Brendan McMahan, Felix X. Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *CoRR*, abs/1610.05492, 2016. 2

[33] Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *CVPR*, pages 10713–10722, 2021. 2

[34] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *MLSys*, 2020. 2, 6, 7

[35] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *ICML*, pages 6357–6368, 2021. 2, 6, 7

[36] Xiaoxiao Li, Meirui JIANG, Xiaofei Zhang, Michael Kamp, and Qi Dou. FedBN: Federated learning on non-IID features via local batch normalization. In *ICLR*, 2021. 2

[37] Xin-Chun Li, De-Chuan Zhan, Yunfeng Shao, Bingshuai Li, and Shaoming Song. Fedphp: Federated personalization with inherited private models. In *ECML PKDD*, pages 587–602, 2021. 2

[38] Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. In *NeurIPS*, pages 18878–18890, 2021. 3, 5

[39] Ken Liu, Shengyuan Hu, Steven Z Wu, and Virginia Smith. On privacy and personalization in cross-silo federated learning. *NeurIPS*, 35:5925–5940, 2022. 3

[40] Liyang Liu, Yi Li, Zhanghui Kuang, Jing-Hao Xue, Yimin Chen, Wenming Yang, Qingmin Liao, and Wayne Zhang. Towards impartial multi-task learning. In *ICLR*, 2021. 3

[41] Shikun Liu, Edward Johns, and Andrew J. Davison. End-to-end multi-task learning with attention. In *CVPR*, pages 1871–1880, 2019. 3

[42] Yang Liu, Yan Kang, Tianyuan Zou, Yanhong Pu, Yuanqin He, Xiaozhou Ye, Ye Ouyang, Ya-Qin Zhang, and Qiang Yang. Vertical federated learning. *CoRR*, abs/2211.12814, 2022. 2

[43] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pages 10012–10022, 2021. 6

[44] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Philip S Yu. Learning multiple tasks with multilinear relationship networks. *NeurIPS*, 30, 2017. 2

[45] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 6

[46] Yongxi Lu, Abhishek Kumar, Shuangfei Zhai, Yu Cheng, Tara Javidi, and Rogerio Feris. Fully-adaptive feature sharing in multi-task networks with applications in person attribute classification. In *CVPR*, pages 5334–5343, 2017. 5

[47] Yuxiang Lu, Shalayiding Sirejiding, Yue Ding, Chunlin Wang, and Hongtao Lu. Prompt guided transformer for multi-task dense prediction. *arXiv preprint arXiv:2307.15362*, 2023. 2

[48] Linhao Luo, Yumeng Li, Buyu Gao, Shuai Tang, Sinan Wang, Jiancheng Li, Tanchao Zhu, Jiancai Liu, Zhao Li, and Shirui Pan. MAMDR: A model agnostic learning framework for multi-domain recommendation. In *ICDE*, pages 3079–3092, 2023. 5

[49] Xiaosong Ma, Jie Zhang, Song Guo, and Wenchao Xu. Layer-wised model aggregation for personalized federated learning. In *CVPR*, pages 10092–10101, 2022. 5

[50] Kevis-Kokitsi Maninis, Ilija Radosavovic, and Iasonas Kokkinos. Attentive single-tasking of multiple tasks. In *CVPR*, pages 1851–1860, 2019. 2, 7

[51] Othmane Marfoq, Giovanni Neglia, Aurélien Bellet, Laetitia Kameni, and Richard Vidal. Federated multi-task learning under a mixture of distributions. In *NeurIPS*, pages 15434–15447, 2021. 1, 3

[52] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *AISTATS*, pages 1273–1282, 2017. 2

[53] Brendan McMahan, Eider Moore, Daniel Ramage, et al. Communication-efficient learning of deep networks from decentralized data. In *AISTATS*, pages 1273–1282, 2017. 2, 6, 7

[54] Jed Mills, Jia Hu, and Geyong Min. Multi-task federated learning for personalised deep neural networks in edge computing. *TPDS*, 33(3):630–641, 2021. 1, 3

[55] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-stitch networks for multi-task learning. In *CVPR*, pages 3994–4003, 2016. 2, 5

[56] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *CVPR*, pages 891–898, 2014. 6

[57] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, and Luca Antiga. Pytorch: An imperative style, high-performance deep learning library. *NeurIPS*, 32, 2019. 6

[58] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *ICCV*, pages 12179–12188, 2021. 1

[59] Jiawei Ren, Cunjun Yu, Xiao Ma, Haiyu Zhao, Shuai Yi, et al. Balanced meta-softmax for long-tailed visual recognition. *NeurIPS*, 33:4175–4186, 2020. 2

[60] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017. 2

[61] Sebastian Ruder, Joachim Bingel, Isabelle Augenstein, and Anders Søgaard. Latent multi-task architecture learning. In *AAAI*, pages 4822–4829, 2019. 2

[62] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. In *NeurIPS*, pages 525–536, 2018. 3

[63] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, pages 746–760, 2012. 6

[64] Shalayiding Sirejiding, Yuxiang Lu, Hongtao Lu, and Yue Ding. Scale-aware task message transferring for multi-task learning. In *ICME*, pages 1859–1864, 2023. 2

[65] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet Talwalkar. Federated multi-task learning. In *NeurIPS*, pages 4424–4434, 2017. 1, 2, 3

[66] Guolei Sun, Thomas Probst, Danda Pani Paudel, Nikola Popović, Menelaos Kanakis, Jagruti Patel, Dengxin Dai, and Luc Van Gool. Task switching network for multi-task learning. In *ICCV*, pages 8291–8300, 2021. 2

[67] Canh T Dinh, Nguyen Tran, and Josh Nguyen. Personalized federated learning with moreau envelopes. *NeurIPS*, 33:21394–21405, 2020. 2

[68] Alysa Ziying Tan, Han Yu, Lizhen Cui, and Qiang Yang. Towards Personalized Federated Learning. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–17, 2022. 2

[69] Simon Vandenhende, Stamatios Georgoulis, and Luc Van Gool. Mti-net: Multi-scale task interaction networks for multi-task learning. In *ECCV*, pages 527–543, 2020. 2, 5

[70] Simon Vandenhende, Stamatios Georgoulis, Wouter Van Gansbeke, Marc Proesmans, Dengxin Dai, and Luc Van Gool. Multi-task learning for dense prediction tasks: A survey. *IEEE TPAMI*, 44(7):3614–3633, 2021. 2

[71] Matthew Wallingford, Hao Li, Alessandro Achille, Avinash Ravichandran, Charless Fowlkes, Rahul Bhotika, and Stefano Soatto. Task adaptive parameter sharing for multi-task learning. In *CVPR*, pages 7561–7570, 2022. 5

[72] Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris S. Papailiopoulos, and Yasaman Khazaeni. Federated learning with matched averaging. In *ECCV*, 2020. 2

[73] Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H. Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. In *NeurIPS*, 2020. 2

[74] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *ICCV*, pages 568–578, 2021. 1

[75] Zirui Wang, Yulia Tsvetkov, Orhan Firat, and Yuan Cao. Gradient vaccine: Investigating and improving multi-task optimization in massively multilingual models. In *ICLR*, 2021. 3

[76] Chulin Xie, De-An Huang, Wenda Chu, Daguang Xu, Chaowei Xiao, Bo Li, and Anima Anandkumar. Perada: Parameter-efficient and generalizable federated learning personalization with guarantees. *CoRR*, abs/2302.06637, 2023. 2

[77] Dan Xu, Wanli Ouyang, Xiaogang Wang, and Nicu Sebe. Pad-net: Multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing. In *CVPR*, pages 675–684, 2018. 2, 5

[78] Yangyang Xu, Xiangtai Li, Haobo Yuan, Yibo Yang, and Lefei Zhang. Multi-task learning with multi-query transformer for dense prediction. *IEEE TCSVT*, 2023. 2

[79] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Trans. Intell. Syst. Technol.*, 10(2):12:1–12:19, 2019. 1

[80] Feiyang Ye, Baijiong Lin, Zhixiong Yue, Pengxin Guo, Qiao Xiao, and Yu Zhang. Multi-objective meta learning. In *NeurIPS*, pages 21338–21351, 2021. 3

[81] Hanrong Ye and Dan Xu. Inverted pyramid multi-task transformer for dense scene understanding. In *ECCV*, pages 514–530, 2022. 2

[82] Hanrong Ye and Dan Xu. Invpt++: Inverted pyramid multi-task transformer for visual scene understanding. *arXiv preprint arXiv:2306.04842*, 2023. 2

[83] Fuxun Yu, Weishan Zhang, Zhuwei Qin, Zirui Xu, Di Wang, Chenchen Liu, Zhi Tian, and Xiang Chen. Fed2: Feature-aligned federated learning. In *ACM SIGKDD*, pages 2066–2074, 2021. 2

[84] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *NeurIPS*, 2020. 3

[85] Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan H. Greenewald, Trong Nghia Hoang, and Yasaman Khazaeni. Bayesian nonparametric federated learning of neural networks. In *ICML*, pages 7252–7261, 2019. 2

[86] Xiaoya Zhang, Ling Zhou, Yong Li, Zhen Cui, Jin Xie, and Jian Yang. Transfer vision patterns for multi-task pixel learning. In *ACM MM*, pages 97–106, 2021. 2

[87] Zhenyu Zhang, Zhen Cui, Chunyan Xu, Zequn Jie, Xiang Li, and Jian Yang. Joint task-recursive learning for semantic segmentation and depth estimation. In *ECCV*, pages 235–251, 2018. 2

[88] Zhenyu Zhang, Zhen Cui, Chunyan Xu, Yan Yan, Nicu Sebe, and Jian Yang. Pattern-affinitive propagation across depth, surface normal and semantic segmentation. In *CVPR*, pages 4106–4115, 2019. 2

[89] Ling Zhou, Zhen Cui, Chunyan Xu, Zhenyu Zhang, Chaoqun Wang, Tong Zhang, and Jian Yang. Pattern-structure diffusion for multi-task learning. In *CVPR*, pages 4514–4523, 2020. 2

[90] Ligeng Zhu, Hongzhou Lin, Yao Lu, Yujun Lin, and Song Han. Delayed gradient averaging: Tolerate the communication latency for federated learning. *NeurIPS*, 34:29995–30007, 2021. 2

[91] Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. Data-free knowledge distillation for heterogeneous federated learning. In *ICML*, pages 12878–12889, 2021. 2

[92] Weiming Zhuang, Yonggang Wen, Lingjuan Lyu, and Shuai Zhang. Mas: Towards resource-efficient federated multiple-task learning. In *ICCV*, pages 23414–23424, 2023. 1, 3