CONFLICT-AVERSE UPDATE DECOUPLING FOR FEDERATED MULTI-TASK LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated Multi-Task Learning (FMTL) enables clients with different tasks to collaboratively train models without sharing raw data. However, task and data heterogeneity jointly induce conflicting updates across clients. Existing FMTL methods treat the aggregated update each client receives from the server as a whole, or keep only the dimensions selected by magnitude; such updates may oppose the client's local direction, so the two cancel each other out and cause negative transfer. In this paper, we propose CAUD (**C**onflict-**A**verse **U**pdate **D**ecoupling), a personalized aggregation method that mitigates parameter conflicts in FMTL without introducing any additional learnable parameters or network modules. Specifically, we adopt a vector obtained from a conflict-averse max-min problem as the global guidance, and decouple each client's update into consensus and client-specific parts through element-wise direction consistency with this guidance; the two parts are then modulated with closed-form weights to produce a personalized update for each client, thereby suppressing conflicts while retaining beneficial task-specific knowledge. Extensive experiments on the standard PASCAL-Context and NYUD-v2 benchmarks demonstrate that CAUD consistently improves performance across various heterogeneous client configurations.

1 INTRODUCTION

Federated Learning (FL) (McMahan et al., 2017) has emerged as a prominent distributed training paradigm, attracting increasing attention from both academia and industry (Ji et al., 2024). It enables multiple clients to collaboratively train a global model without sharing raw data (Zhu et al., 2026). Specifically, clients perform local training on their own data and upload model updates to a central server, which aggregates these updates to update the global model. This process facilitates collaborative learning while mitigating privacy risks associated with centralized data collection.

However, FL is often limited by heterogeneity in both data distributions and task objectives (Lu et al., 2024). Although numerous distributed data sources provide abundant data, the data collected by different clients are typically non-independent and identically distributed (non-IID), a property widely regarded as a fundamental challenge in FL. Moreover, clients often need to solve different tasks, and their heterogeneous optimization objectives make it difficult to directly apply existing FL algorithms. These factors pose significant challenges to the performance and practical deployment of FL. Federated Multi-Task Learning (FMTL) (Smith et al., 2017) generalizes FL to multiple correlated tasks, especially dense prediction tasks such as semantic segmentation and depth estimation (Cai et al., 2023), which benefit most from shared parameters or representations. Multiple clients collaboratively train a shared model without uploading raw data, while each client specializes in its own task and leverages task correlations through parameters or representations.

Designing an FMTL method that efficiently coordinates heterogeneous clients remains challenging. Clients differ in data distributions, objective functions, label structures, and even output spaces, which manifest as conflicts in the direction and magnitude of client updates, as well as coupling between shared and client-specific information. Traditional FL such as FedAvg (McMahan et al., 2017) assumes homogeneous client tasks with consistent local objectives, so that local updates approximate the same global objective and weighted parameter averaging is reasonable. In FMTL, directly averaging parameters may cause conflicting dimensions to cancel each other out (Wei et al.,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

2025), leading to negative transfer, whereas independent training cannot exploit shared knowledge across clients (Smith et al., 2017).

Existing FMTL methods such as FedHCA^2 (Lu et al., 2024) propose equipping each client with a learnable hypernetwork that adaptively preserves globally optimal updates beneficial to the local client. Their limitation is that personalization is expressed through learned weights that decide how much of the aggregate each client adopts at the granularity of whole modules, while local direction discrepancies within individual parameter dimensions are not modeled, so conflicting dimensions may still cancel each other out. Wei et al. (2025) applies top-k filtering to the magnitudes of client update dimensions and identifies the largest activations as task-relevant dimensions. Nevertheless, the retained dimensions may still yield update shifts inconsistent with global guidance after aggregation, and the hard mask inevitably incurs information loss.

Our core hypothesis is that recent methods fail to effectively disentangle cross-task shareable knowledge from client-specific knowledge at the parameter level, which leaves update conflicts unresolved during aggregation. To this end, we propose a conflict-averse update decoupling framework for personalized aggregation in FMTL. Specifically, we first construct a global guidance vector by solving a conflict-averse max-min problem (Liu et al., 2021) over the parameter updates uploaded by all clients in the current communication round, looking for the best update within a local ball centered on their average, thereby obtaining an update direction that remains robust to the disagreement induced by heterogeneous tasks. Then, for each client, we perform an element-wise assessment of directional consistency between the client parameter increment and the guidance vector to decouple the increment into a consensus part representing general knowledge and a client-specific part representing task-specific knowledge. Finally, we rescale and combine these two parts using closed-form weights to generate a personalized update for each client. This design aims to mitigate conflicts induced by client task discrepancies while preserving cross-task shared knowledge, thereby enhancing the adaptability and performance of FMTL in heterogeneous environments. Notably, as verified in our ablation study, the conflict-averse guidance alone is not sufficient to resolve the conflicts, whereas the coordinate-level decoupling accounts for the majority of the performance gain, indicating that the bottleneck of existing FMTL methods lies not in constructing a shared direction but in resolving conflicts on individual parameter dimensions.

In summary, our main contribution is a personalized aggregation method for FMTL that is architecture-agnostic and requires no additional learnable parameters or modules. Extensive experiments on the NYUD-v2 and PASCAL-Context benchmarks demonstrate that the method consistently outperforms state-of-the-art FMTL methods.

2 Related Work

Federated Learning. *Federated Learning (FL)* (McMahan et al., 2017) uses private data for local training and shares only aggregated model updates, enabling decentralized collaborative learning while ensuring privacy. To address the common issue of non-independent and identically distributed (non-IID) data in federated learning (FL), traditional FL methods employ a series of strategies to improve the convergence and robustness of the global model, including regularization (Li et al., 2020; Karimireddy et al., 2020; Acar et al., 2021), reparameterization (Hao et al., 2021), adaptive optimization (Reddi et al., 2021), training without aggregation (Wang et al., 2024), and contrastive learning (Li et al., 2021a; Huang et al., 2023). In contrast, *personalized federated learning (pFL)* focuses on preserving substantive differences among client models. By allowing partial aggregation (Li et al., 2021c; Arivazhagan et al., 2019), personalizing classifiers through local fine-tuning (Oh et al., 2022), adopting attention-weighted aggregation (Huang et al., 2021), or optimizing information representation (Huang et al., 2025). Although these methods are effective in mitigating data heterogeneity, they are often difficult to apply when there are significant differences in the task objectives across clients.

Federated Multi-Task Learning. *Multi-Task Learning* (MTL) aims to develop models that can solve multiple tasks jointly while efficiently sharing computation (Caruana, 1997; Vandenhende et al., 2021). Existing popular methods either tackle task interference mainly through multi-task conflict optimization (Kendall et al., 2018; Liu et al., 2019; 2023; Guo et al., 2018; Chen et al., 2018; Navon et al., 2022; Sener & Koltun, 2018; Yu et al., 2020; Liu et al., 2021; Chen et al., 2020; Ban & Ji, 2024) and and extracting shared and task-specific representations via multi-task architectures (Vandenhende

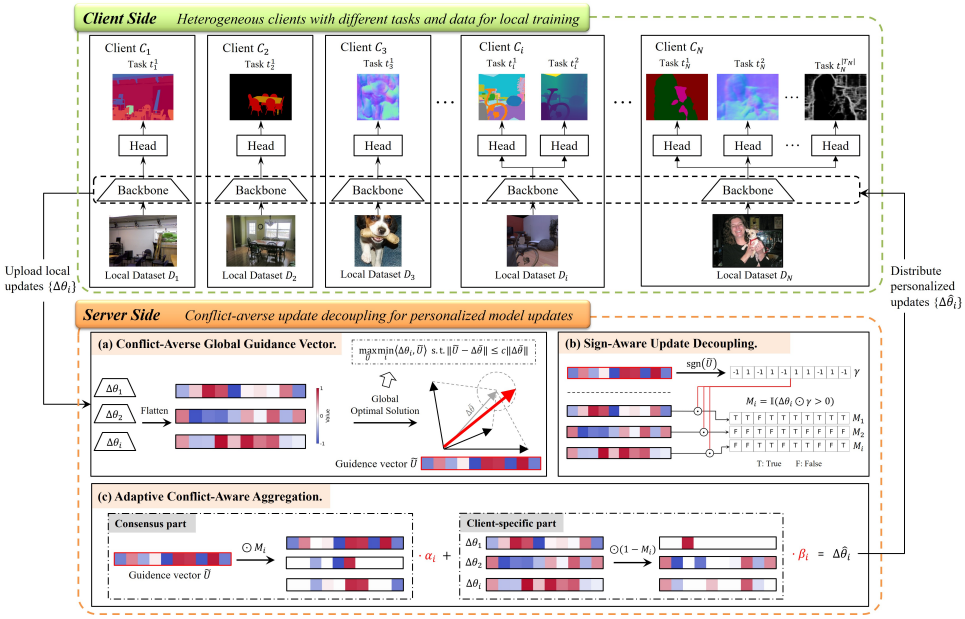


Figure 1: Overview of the proposed CAUD framework. Each client uploads its local update $\Delta\theta_i$ to the server, which (a) constructs a conflict-averse guidance vector $\tilde{U}$ within a ball around the mean update, (b) decouples $\Delta\theta_i$ into consensus and client-specific parts by element-wise direction consistency, and (c) modulates the two parts with closed-form weights to produce the personalized update $\Delta\hat{\theta}_i$ sent back for the next round of local training.

et al., 2019; Sun et al., 2020; Misra et al., 2016; Bruggemann et al., 2020; Guo et al., 2020; liang et al., 2022; Ye & Xu, 2023a; Chen et al., 2023; Yang et al., 2024; Ye & Xu, 2023b; Zhang et al., 2019; Vandenhende et al., 2020; Brüggemann et al., 2021; Ye & Xu, 2022; Lu et al., 2025; Li et al., 2024; Wang et al., 2026). However, different tasks are typically handled by different clients. *Federated Multi-Task Learning (FMTL)* (Smith et al., 2017) was proposed to enable collaborative training among clients with different task objectives. Early FMTL methods focused on achieving personalized adaptation through task-relationship regularization (Smith et al., 2017) and hybrid models (Marfoq et al., 2021). More recent methods further account for both task and data heterogeneity, e.g., through dynamic client clustering (Cai et al., 2023), conflict-avoidance gradient projection (Chen et al., 2024), differential encoder–decoder aggregation with learnable hyper-aggregation weights (Lu et al., 2024), or adversarial task-agnostic representation learning (Zhang et al., 2025).

In summary, existing methods cannot effectively decouple the update directions from different clients, and parameter conflicts between clients will continue to interfere with one another.

3 Methodology

We first introduce the problem setup and define parameter conflicts, then present our method.

3.1 Federated Multi-task Learning

Problem Setup. As shown in Fig. 1 (top), we consider the Heterogeneous-Client Federated Multi-Task Learning (HC-FMTL) scenario (Lu et al., 2024), where each client involve an arbitrary number of tasks. Formally, client C_i ($i \in \{1, \ldots, N\}$) has a local dataset $\mathcal{D}_i = \{(x_n^i, y_n^i)\}_{n=1}^{|\mathcal{D}_i|}$ and a task set $\mathcal{T}_i \subseteq \mathcal{T}$, where $\mathcal{T} = \{1, \ldots, T\}$ is the global task set and T is the total number of tasks. $x_n^i \in \mathcal{X}$ is the input sample and y_n^i is the corresponding label for the tasks in $\mathcal{T}_i$. Each client C_i maintains its own parameters $\theta_i \in \mathbb{R}^d$. For every task $t \in \mathcal{T}_i$, client C_i defines a task loss $\mathcal{L}_{i,t}(\theta_i)$ computed on $\mathcal{D}_i$. The local objective of client C_i is to minimize the aggregated losses over its task set:

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

In the $\kappa \in \{1, \ldots, K\}$th round of communication, let $\Delta\theta_i^{(\kappa)}$ denote the increment produced by client C_i's local training in that round.

Parameter Conflicts. Due to varying client configurations and differences in local training, the uploaded increments $\{\Delta\theta_i^{(\kappa)}\}_{i=1}^N$ may result in *parameter conflicts* among clients. We use the signs of the increment components to characterize conflicts at the element level (Huang et al., 2024). Given two client increments $\Delta\theta_i^{(\kappa)}$ and $\Delta\theta_j^{(\kappa)}$, if the signs of their corresponding components on certain parameter dimensions are opposite, the two increments are considered to be in conflict, that is:

$$\Delta\theta_{i,q}^{(\kappa)} \cdot \Delta\theta_{j,q}^{(\kappa)} < 0 \tag{2}$$

where q denotes the parameter dimension. To mitigate such conflicts, round κ aims to construct a client update $\Delta\hat{\theta}_i^{(\kappa)}$ from $\{\Delta\theta_j^{(\kappa)}\}_{j=1}^N$ that suppresses the conflict while preserving components beneficial to client C_i. Then $\Delta\hat{\theta}_i^{(\kappa)}$ updates the parameters on client C_i, and the updated parameters are used for local training in round $\kappa + 1$.

3.2 Conflict-Averse Update Decoupling

As illustrated in Fig. 1 (bottom), we propose **C**onflict-**A**verse **U**pdate **D**ecoupling (CAUD). It consists of three core components: (i) a Conflict-Averse Global Guidance Vector coordinating updates across clients; (ii) a Sign-Aware Update Decoupling into consensus and client-specific parts; and (iii) an Adaptive Conflict-Aware Aggregation that modulates the two parts with closed-form weights.

Conflict-Averse Global Guidance Vector. At the aggregation step of each communication round, the server collects the increments $\{\Delta\theta_i\}_{i=1}^N$ uploaded by all clients and solves for $\tilde{U}$ once, before any decoupling or subsequent local training, as shown in Fig. 1 (a). The guidance vector is therefore a federation-level quantity determined entirely by the current round's uploaded increments: it is not produced during local training, and it introduces no additional learnable parameters. The simplest way to turn these increments into a shared direction is to assign every client the mean update,

$$\Delta\hat{\theta}_i = \Delta\bar{\theta} = \frac{1}{N}\sum_{j=1}^{N}\Delta\theta_j, \qquad \forall i \in \{1, \ldots, N\}. \tag{3}$$

This implicitly assumes that every client is equally compatible with the shared direction, which rarely holds under strong task and data heterogeneity. Under the sign conflicts of Eq. (2), opposing components cancel along each conflicting coordinate q, so $|\Delta\bar{\theta}_q|$ can fall well below the clients' proposed magnitudes, and C_i's own signal is suppressed exactly where it disagrees with the rest of the federation. Therefore, we construct a global guidance vector $\tilde{U}$ with two requirements. First, it must remain within a bounded ball around the mean update $\Delta\bar{\theta}$, so that global progress stays anchored to the federation's average direction and no single client or coalition can hijack the aggregation. Concretely, this ball is centered on $\Delta\bar{\theta}$ with radius $c\|\Delta\bar{\theta}\|$. Second, it should align as closely as possible with every client's update, so that replacing a client's update by the guidance does not systematically harm that client. Since the inner product $\langle\Delta\theta_i, \tilde{U}\rangle$ measures this alignment, requiring it to be large in the worst case yields

$$\max_{\tilde{U}} \min_i \langle\Delta\theta_i, \tilde{U}\rangle \quad \text{s.t.} \quad \left\|\tilde{U} - \Delta\bar{\theta}\right\| \le c\left\|\Delta\bar{\theta}\right\|, \tag{4}$$

where $c \in [0, 1)$ controls the trust-region radius. Since the uploaded increment $\Delta\theta_i$ points opposite to the gradient accumulated during client C_i's local training in round κ, Eq. (4) equivalently selects the direction that minimizes the maximum first-order loss increase across clients. This is the update-space analogue of the conflict-averse gradient problem (Liu et al., 2021). The same problem was federated in (Lu et al., 2024), where its solution only determines inter-client combination weights and personalization is delegated to a learnable hypernetwork that adapts how much of the aggregate each client adopts. We instead use the solution as a coordinate-level direction reference, and achieve personalization without any learnable parameters through the decoupling below.

Eq. (4) has a closed-form solution via its dual (Liu et al., 2021); the detailed derivation is provided in the Appendix. The guidance vector is recovered as a fixed-radius step from the mean update:

$$\tilde{U} = \Delta\bar{\theta} + \frac{\sqrt{\phi}\, U_{w^\star}}{\|U_{w^\star}\|}. \tag{5}$$

Here $U_{w^\star} = \frac{1}{N}\sum_{i=1}^{N} w_i^\star \, \Delta\theta_i$ is the convex combination of client updates with the dual-optimal coefficients $w^\star$, and $\phi = c^2\left\|\Delta\bar{\theta}\right\|^2$. Geometrically, $\tilde{U}$ shifts the mean update within the trust region toward the direction most compatible with all clients. Two properties of this construction matter below. First, $\tilde{U}$ is *scale-calibrated*: its deviation from the mean is bounded by a fraction of the mean's magnitude, so its magnitude reflects the clients' update scales. Second, it is *worst-case aligned*: it minimizes the largest directional opposition within the ball.

Sign-Aware Update Decoupling. One may let every client simply adopt the guidance vector $\tilde{U}$ as its update. However, $\tilde{U}$ is only *worst-case aligned*: it guarantees no client any individual improvement, and applying it uniformly would overwrite the per-client components carrying each client's own task set. Instead, as in Fig. 1 (b), we decouple $\Delta\theta_i$ into a consensus part and a client-specific part by evaluating element-wise sign consistency with the guidance. We first extract the element-wise direction flag from the guidance:

$$\gamma = \mathrm{sgn}(\tilde{U}), \tag{6}$$

where $\gamma \in \{-1, +1\}^d$ records the conflict-averse direction per dimension. For each client, we build a binary mask marking the coordinates where the client's update direction agrees with the guidance:

$$M_i = \mathbb{I}(\Delta\theta_i \odot \gamma > 0), \tag{7}$$

where $\mathbb{I}(\cdot)$ is the element-wise indicator function. M_i acts as a 1-bit separator: it marks the parameter dimensions where the client's update direction agrees with the guidance direction as *consensus* coordinates, and the remaining dimensions as *client-specific* coordinates. The two parts are then modulated with two closed-form weights derived below.

Adaptive Conflict-Aware Aggregation. With the guidance vector $\tilde{U}$ and the mask M_i above, we express the personalized update of client C_i via adaptive conflict-aware aggregation (Fig. 1 (c)):

$$\Delta\hat{\theta}_i = \alpha_i \left(M_i \odot \tilde{U}\right) + \beta_i \left(\bar{M}_i \odot \Delta\theta_i\right), \tag{8}$$

where $\bar{M}_i = \mathbf{1} - M_i$ complements the mask element-wise. The first term replaces the client's update with the guidance on consensus coordinates, scaled by α_i; the second keeps its update on client-specific coordinates, scaled by β_i. Both weights are determined in closed form below.

Consensus part. On consensus coordinates the client's update already agrees with the guidance in sign, so adopting the guidance preserves its local direction while inheriting a magnitude calibrated against the mean update $\Delta\bar{\theta}$, not any single client. As for scale, the adopted update should stay close to both the guidance and the client's own update; since the two references are not collinear, any $\alpha_i \neq 1$ is a detour whose total distance to both exceeds the direct distance between them, attained exactly at $\alpha_i = 1$. We therefore fix $\alpha_i = 1$: consensus coordinates adopt the guidance directly, with no learnable scale. The proof that $\alpha_i = 1$ is the unique minimizer is in the Appendix.

Client-specific part. Our sign-agreement criterion is inspired by post-hoc model merging, where sign-based masks resolve conflicting coordinates by discarding them outright (Huang et al., 2024):

$$\Delta\hat{\theta}_i = M_i \odot \tilde{U}. \tag{9}$$

However, in an iterative, per-client aggregation pipeline, such a static binary decision is unnecessarily coarse: sign disagreement does not imply that the client's update on these coordinates is harmful, since it may still carry personalized information beneficial to its own tasks, and hard elimination risks discarding much useful update information. We therefore replace the static mask with a continuous, closed-form modulation that shrinks rather than removes conflicting coordinates, determined by the relative strength of the conflict.

Denote by $P_i = \bar{M}_i \odot \Delta\theta_i$ the client's *conflicting component* and by $u_P = \bar{M}_i \odot \tilde{U}$ the guidance's component on the same client-specific coordinates. We set

$$\beta_i = \frac{\|P_i\|}{\|u_P\| + \|P_i\| + \epsilon}, \tag{10}$$

where $\epsilon > 0$ is a small constant for numerical stability and $\|\cdot\|$ is the ℓ_2 norm. Note that β_i is a single scalar per client per round, computed from the overall norm of all client-specific coordinates taken together; the same factor multiplies every client-specific coordinate uniformly, so the direction of the client's conflicting component is preserved and only its overall strength is rescaled.

Discussion. This weight is not heuristic: it uniquely minimizes a per-client trade-off between discarding the client's conflicting component and retaining conflict with the guidance; the derivation is provided in the Appendix. Consequently, $\beta_i \in [0, 1)$: it approaches 1 when the client's conflicting component dominates ($\|u_P\| \to 0$), fully retaining the client's own component, and 0 when the guidance dominates ($\|P_i\| \to 0$), reducing to the hard mask of Eq. (9).

4 EXPERIMENTS

Here we first describe the benchmarks and implementation details, then analyze our method in depth.

4.1 SETUP

Datasets. We conduct the experiments on two widely-used datasets for FMTL dense prediction: NYUD-v2 (Nathan Silberman & Fergus, 2012) and PASCAL-Context (Mottaghi et al., 2014). NYUD-v2 contains 795 training and 654 testing images, semantic segmentation (SemSeg), monocular depth estimation (Depth), surface normal estimation (Normals), and edge detection (Edge). PASCAL-Context contains 4,998 training samples and 5,105 testing samples, annotated for five tasks of SemSeg, human parts segmentation (Parts), Normals, saliency detection (Sal), and Edge. See the supplementary material for details.

Baselines. We compare against traditional FL (FedAvg (McMahan et al., 2017), FedProx (Li et al., 2020)), personalized FL (FedPer (Arivazhagan et al., 2019), FedBABU (Oh et al., 2022), Ditto (Li et al., 2021b), FedAMP (Huang et al., 2021)), MTL (CAGrad (Liu et al., 2021), FairGrad (Ban & Ji, 2024)), model merging (EMR-Merging (Huang et al., 2024)), and FMTL (FedSTA (Park et al., 2021), MaT-FL (Cai et al., 2023), FedHCA^2 (Lu et al., 2024), FedDEA (Wei et al., 2025), FairFHTL (Zhang et al., 2025)) methods. This baseline suite provides a fair and comprehensive evaluation against complementary paradigms.

Implementation Details. All methods are implemented in PyTorch and are based on a pre-trained Swin-T backbone, combined with task-specific FCN decoders and heads. We adopt the same training configuration as described in (Lu et al., 2024): we train for one local epoch per communication round on PASCAL-Context, and four local epochs on NYUD-v2. We set the total number of communication rounds to 100 and the batch size to 8. All models were optimized using AdamW, with both the learning rate and weight decay set to 1×10^{-4}. We set the ball radius $c = 0.4$ in Eq. (4), following the value used for NYUD-v2 in (Liu et al., 2021); this choice is validated in the supplementary material.

Metrics. We follow the prior work and use the same evaluation metrics (Vandenhende et al., 2021; Lu et al., 2024). Specifically, we use mean Intersection over Union (mIoU) for segmentation problems, Maximal F-measure (maxF) for Sal, optimal dataset F-measure (odsF) for Edge, root mean square error (RMSE) for Depth, and mean error (mErr) for Normals. To quantify the average performance gain of the MTL methods, we use the multi-task performance metric Δ_m as in prior work (Vandenhende et al., 2021); its formal definition is provided in the supplementary material.

4.2 MAIN RESULTS

Comparison with SotA methods. Tab. 1 presents the results under the heterogeneous PASCAL-Context (ST)& NYUD-v2 (MT) client configuration, where the composite metric Δ_m measures the average performance change relative to the local baseline. CAUD attains the best Δ_m of 4.18%, outperforming the strongest FMTL baseline FedHCA^2 (2.18%) by 2.00%. CAUD does not lead on every column, and no method does: it leads outright on three (PASCAL-Context SemSeg and Parts, and NYUD-v2 Depth) and ties on a fourth (PASCAL-Context Edge), whereas FedHCA^2 leads on only two. Where CAUD does not lead, the margin is small (no gap exceeds 0.24), and its largest

Table 1: Comparison to representative methods using **PASCAL-Context** for five single-task (**ST**) clients and **NYUD-v2** for one multi-task (**MT**) client. '↑' means higher is better and '↓' means lower is better. '+' indicates a version that has been reproduced based on the original method and adapted to these settings. '$\Delta_m\,\%$' denotes the average performance change w.r.t. local baseline. The **best** and second-best results for each column are highlighted in bold and underlined, respectively.

Method	PASCAL-Context (**ST**)					NYUD-v2 (**MT**)				
	SemSeg mIoU↑	Parts mIoU↑	Sal maxF↑	Normals mErr↓	Edge odsF↑	SemSeg mIoU↑	Depth RMSE↓	Normals mErr↓	Edge odsF↑	$\Delta_m\% \uparrow$
Local	51.69	49.94	**80.91**	15.76	71.95	41.86	0.6487	20.59	76.46	0.00
Traditional Federated Learning										
FedAvg	39.98	37.33	77.56	18.27	69.17	38.94	0.7858	21.62	75.77	-11.76
FedProx	44.42	38.10	77.26	18.03	69.39	39.19	0.8068	21.52	76.03	-10.68
Personalized Federated Learning										
FedPer	54.51	46.56	78.85	16.95	71.00	44.02	0.6467	21.19	76.61	-1.11
Ditto	46.23	39.69	77.99	17.52	69.77	41.49	0.6508	20.60	76.45	-5.57
FedBABU	52.60	46.41	79.47	16.88	71.13	43.98	0.6361	21.14	76.70	-1.19
FedAMP	55.98	52.05	80.79	15.74	72.02	41.67	0.6428	20.54	76.40	1.47
Multi-task Learning										
CAGrad+	57.49	51.30	79.08	16.87	71.39	44.14	0.6175	21.24	76.70	1.26
FairGrad+	58.23	50.52	79.15	16.93	71.48	**44.53**	0.6211	21.68	76.58	1.01
Model Merging										
EMR-Merging+	58.82	51.98	80.56	15.65	**72.11**	38.87	0.6298	20.68	76.39	2.14
Federated Multi-task Learning										
FedSTA	52.24	50.68	77.97	16.90	70.07	35.62	0.7384	22.87	75.35	-5.80
MaT-FL	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
FedDEA	55.03	48.36	79.54	16.35	71.12	41.81	0.6158	20.81	**76.72**	0.03
FairFHTL	57.95	50.20	78.95	17.03	71.39	44.07	0.6328	21.71	76.65	0.69
CAUD (ours)	**61.92**	**53.09**	80.88	15.84	**72.11**	44.30	**0.6066**	20.73	76.58	**4.18**

Table 2: Comparison of multi-task clients on task subsets on NYUD-v2 and PASCAL-Context.

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	0.00
FedHCA^2	64.17	54.25	**14.01**	40.26	**20.55**	0.53
CAUD (ours)	**65.35**	**55.36**	14.10	**41.63**	20.70	**1.72**

regression is 0.14 mErr on surface normal estimation (15.84 vs. 15.76 on PASCAL-Context, 20.73 vs. 20.59 on NYUD-v2). This residual is task-specific rather than client-specific: within the same NYUD-v2 multi-task client, the same aggregation improves SemSeg (+2.44), Depth (-0.042 RMSE), and Edge (+0.12) while normals changes by 0.14. Averaged across tasks and clients, CAUD remains best, improving over the local baseline on six of the nine columns and matching it to within 0.03 on a seventh. This gain requires no additional trainable parameters, indicating that it stems from coordinating conflicting updates during aggregation rather than from increased model capacity. The complementary setting with NYUD-v2 single-task clients and a PASCAL-Context multi-task client is reported in the supplementary material.

Multi-task clients on task subsets. Tab. 2 evaluates a setting where all clients are multi-task models trained on task subsets. CAUD attains a Δ_m of 1.72%, substantially outperforming FedHCA^2 (0.53%), indicating that conflict-averse modulation still extracts beneficial cross-client knowledge without harmful interference even when clients are individually strong multi-task learners. The complementary setting where all clients train on the full task set is reported in the Appendix.

4.3 Ablation Study

Impact of Client Composition. Tab. 3 ablates the effect of client composition on NYUD-v2 single-task clients. With only single-task clients (ST), CAUD raises Δ_m from 0.64% (FedHCA^2) to 4.75%; with multi-task clients joining (ST & MT), it attains 4.99% versus 1.44% for FedHCA^2. In both settings, CAUD substantially improves over the local baseline and outperforms FedHCA^2 by a large margin, regardless of how heterogeneous the federation becomes. These observations indicate that the conflict-averse modulation does not rely on a particular mixture of single-task and multi-task clients, but remains effective as the client composition becomes more heterogeneous.

Table 3: Experimental results under different settings on NYUD-v2 single-task clients.

Setting	SemSeg↑	Depth↓	Normals↓	Edge↑	$\Delta_m\%\uparrow$
ST & Local	33.59	0.7129	23.22	75.02	0.00
ST & FedHCA^2	34.71	0.7170	23.25	74.98	0.64
ST & CAUD (ours)	**37.32**	**0.6668**	**22.93**	**75.14**	**4.75**
ST & MT & FedHCA^2	34.95	0.7018	23.19	75.03	1.44
ST & MT & CAUD (ours)	**37.67**	**0.6702**	**22.90**	**75.36**	**4.99**

Table 4: Ablation study of CAUD under several update variants.

Update rule	PASCAL-Context (**ST**)					NYUD-v2 (**MT**)				$\Delta_m\%\uparrow$
	SemSeg↑	Parts↑	Sal↑	Normals↓	Edge↑	SemSeg↑	Depth↓	Normals↓	Edge↑	
Local : $\Delta\hat{\theta}_i = \Delta\theta_i$	51.69	49.94	80.91	15.76	71.95	41.86	0.6487	**20.59**	76.46	0.00
CAGrad+ : $\Delta\hat{\theta}_i = \tilde{U}$	57.49	51.30	79.08	16.87	71.39	44.14	0.6175	21.24	**76.70**	1.26
① $\Delta\hat{\theta}_i = M_i \odot \tilde{U} + \bar{M}_i \odot \Delta\theta_i$	61.33	52.56	80.74	15.82	72.01	43.71	0.6153	20.73	76.60	3.60
② $\Delta\hat{\theta}_i = M_i \odot \tilde{U}$	61.86	53.05	**81.16**	15.84	71.84	44.03	0.6127	20.94	76.57	3.86
CAUD (ours)	**61.92**	**53.09**	80.88	15.84	**72.11**	**44.30**	**0.6066**	20.73	76.58	**4.18**

Effect of update variants. Tab. 4 ablates each component of CAUD by varying the update rule. Adopting the guidance vector alone as the update (i.e., CAGrad+) improves Δ_m from 0.00% to 1.26%, confirming that the conflict-averse guidance provides meaningful cross-client direction. Replacing the consensus part with the guidance while keeping the client-specific part unmodified (①) yields a large gain ($\Delta_m = 3.60\%$), showing that the sign-aware decoupling into consensus and client-specific parts is the key contributor. Dropping the client-specific part entirely (②, i.e., replacing the closed-form modulation with hard elimination of the client-specific part) slightly outperforms keeping it raw (3.86% vs. 3.60%), indicating that the raw client-specific part is mildly harmful to aggregation. The full CAUD further improves over the hard mask (4.18% vs. 3.86%), as the closed-form weight β_i shrinks the client-specific part instead of zeroing it, thereby retaining the beneficial task-specific information that hard elimination discards. These observations indicate that each component contributes progressively, and that shrinking rather than discarding the conflicting coordinates is essential to fully release the benefit of consensus alignment.

4.4 Further Analysis

Parameter-Level similarity analysis. To understand why CAUD outperforms existing methods from a parameter-update perspective, Fig. 2 analyzes the client-specific updates, since raw parameters are dominated by the shared ImageNet-pretrained initialization. CAUD achieves the highest cross-client cosine similarity (0.284 vs. 0.177 for FedHCA^2 and 0.023 for Local) and the smallest cross-client L2 distance (32.4 vs. 62.5 for FedHCA^2), indicating that its clients converge to the most mutually consistent update directions and thus share knowledge most effectively. In contrast, FedHCA^2 stays closest to independent local training (cos = 0.496), under-sharing knowledge across clients, while FedAvg collapses all clients onto one update that deviates strongly from every client's local direction (cos = 0.108) with the largest magnitude shift (L2 = 52.6), destroying personalization. Notably, CAUD occupies the middle ground: it steers clients toward the shared conflict-averse direction (cos = 0.128 against local training) with a moderate displacement (L2 = 47.9), preserving task specificity while sharing knowledge.

Effect of scaling factors α and β. Fig. 2(c) sweeps CAUD's two scaling factors, where a fixed factor is one scalar shared by all clients. **Effect of α.** Sweeping $\alpha \in [0.4, 1.6]$ with the client-adaptive β_i unchanged, Δ_m rises from 3.48% at $\alpha = 0.4$ to 4.01% at $\alpha = 0.8$, then drops to 3.88% at $\alpha = 1.6$, while the default $\alpha = 1.0$ attains the best Δ_m of 4.18% (dashed reference). Both under-scaling and over-amplification thus degrade performance, corroborating the detour argument in Sec. 3: the guidance vector is calibrated and should be adopted at $\alpha = 1$. **Effect of β.** For a globally fixed β, Δ_m stays within 4.03–4.06% across $\beta \in [0.2, 0.8]$, whereas the extremes drop to 3.86% ($\beta = 0$) and 3.60% ($\beta = 1.0$), showing that no single fixed retention level reconciles all clients' heterogeneous needs. In contrast, the closed-form β_i from the relative conflict strength attains the highest Δ_m of 4.18%, showing that conflicting components are harmful on average and that client-adaptive modulation, not any fixed or hand-tuned scaling, fully unlocks the benefits of consensus alignment.

Training dynamics of sign agreement. Figure 3(a) tracks the sign agreement rate between each client's update and the conflict-averse guidance direction γ, i.e., the fraction of coordinates selected as consensus by the mask in Eq. (7). The single-task clients align progressively with the shared

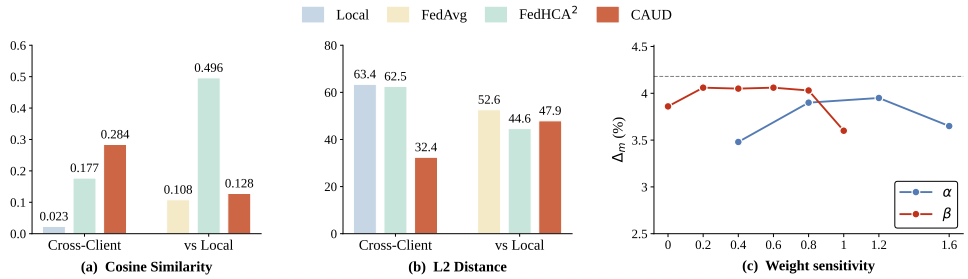


Figure 2: Parameter-level similarity analysis on PASCAL-Context (ST) and NYUD-v2 (MT) setting with the Swin-T backbone. (a) Cosine similarity and (b) L2 distance of client-specific updates ($\Delta = w_{\text{finetuned}} - w_{\text{pretrained}}$); 'Cross-Client' measures the average pairwise similarity among clients, and 'vs Local' the similarity to independent local training. (c) Effect of α and β on Δ_m.

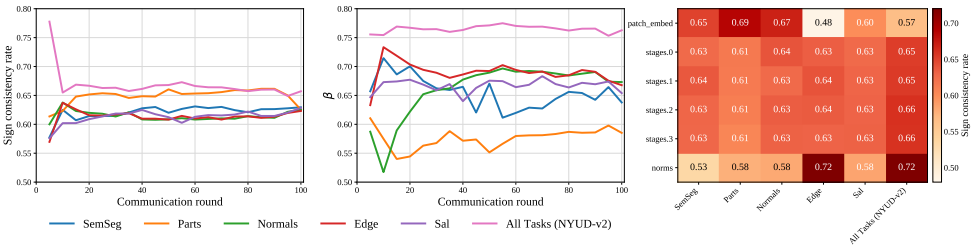


Figure 3: Training dynamics of CAUD on the PASCAL-Context (ST) and NYUD-v2 (MT) setting. (a) Sign agreement rate between each client's update and the conflict-averse guidance vector, and (b) evolution of the closed-form weight β_i, both averaged over blocks of four rounds; (c) sign agreement rate per backbone stage in the final round.

guidance (0.54–0.61 to 0.61–0.63), while the multi-task client relaxes from 0.845 to 0.659 as the guidance adapts to the single-task coalition, so that all clients finally settle into the narrow band of 0.61–0.66. The decoupling is thus stable yet non-trivial: about two thirds of the coordinates act as consensus and one third remain client-specific. Figure 3(b) shows the evolution of the closed-form weight β_i in Eq. (10). Single-task clients settle at 0.59–0.67, while the multi-task client retains the largest share of its conflicting component ($\beta_i = 0.761$), reflecting that its conflicting component dominates the guidance on those coordinates. Since β_i neither collapses to zero nor saturates toward one, the closed-form weight keeps balancing the two parts instead of degenerating into the hard elimination of Eq. (9) or full retention. Notably, the multi-task client consistently maintains both a higher consensus ratio and a larger β_i than the single-task clients throughout training, echoing the intuition that multi-task training provides richer consensus as well as task-specific information, which respectively align it more closely with the shared guidance and justify retaining more of its own component. Figure 3(c) decomposes the final-round agreement rate by backbone stage: the four transformer stages are uniformly shareable (0.61–0.66), whereas divergence concentrates at the two ends of the network (patch embedding 0.48–0.69; output normalization 0.53–0.72). The mask thus localizes personalization to the layers where conflicts actually occur, consistent with the finding that intermediate representations are the most transferable.

5 Conclusion

In this paper, we proposed CAUD to address the parameter conflict problem in Heterogeneous-Client Federated Multi-Task Learning. CAUD constructs a conflict-averse global guidance vector within a ball around the mean update, decouples each client's update into consensus and client-specific parts via element-wise direction consistency, and modulates the two parts with closed-form weights to generate a personalized update for each client. The key advantage of our method is that it introduces no additional learnable parameters or modules, achieving significant performance improvements while maintaining model simplicity. Extensive experiments on two mainstream multi-task dense prediction benchmarks, PASCAL-Context and NYUD-v2, demonstrate that CAUD consistently outperforms existing SotA methods under various heterogeneous client configurations. Future work will explore extending this approach to larger-scale client deployments and more diverse task types.

AI USE STATEMENT

In this work, we used generative AI tools for language translation and grammar polishing of the manuscript, and to find related work; all cited related work was manually verified to ensure its authenticity. Apart from these, we did not use generative AI tools for any other purpose, including research ideation or execution, drafting sections of the paper, generating synthetic datasets, or proving mathematical claims. All AI-assisted content was reviewed and verified by the authors, and we take responsibility for the final content of this work, including all text, claims and artifacts.

REPRODUCIBILITY STATEMENT

We provide the formal proofs of the guidance vector and the closed-form weights in Appendix. Detailed training configurations, hyperparameters, and evaluation protocols are described in Sec. 4 and Appendix. We believe these materials provide sufficient information to reproduce our results.

REFERENCES

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

Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. Federated learning with personalization layers. *arXiv preprint arXiv:1912.00818*, 2019.

Hao Ban and Kaiyi Ji. Fair Resource Allocation in Multi-Task Learning. In *ICML*, volume 235, pp. 2715–2731, 2024.

David Bruggemann, Menelaos Kanakis, Stamatios Georgoulis, and Luc Van Gool. Automated search for resource-efficient branched multi-task networks. In *BMVC*, 2020.

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

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*, pp. 5036–5044, June 2023.

Rich Caruana. Multitask learning. *Machine Learning*, pp. 41–75, Jan 1997.

Yi-Qiang Chen, Teng Zhang, Xin-Long Jiang, Qian Chen, Chen-Long Gao, and Wu-Liang Huang. Fedbone: Towards large-scale federated multi-task learning. *Journal of Computer Science and Technology*, 39(5):1040–1057, 2024.

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In Jennifer G. Dy and Andreas Krause (eds.), *ICML*, volume 80 of *Proceedings of Machine Learning Research*, pp. 793–802. PMLR, 2018.

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 H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *NIPS*, volume 33, pp. 2039–2050. Curran Associates, Inc., 2020.

Zitian Chen, Yikang Shen, Mingyu Ding, Zhenfang Chen, Hengshuang Zhao, Erik G. Learned-Miller, and Chuang Gan. Mod-squad: Designing mixtures of experts as modular multi-task learners. In *CVPR*, pp. 11828–11837, June 2023.

Michelle Guo, Albert Haque, De-An Huang, Serena Yeung, and Li Fei-Fei. Dynamic task prioritization for multitask learning. In *ECCV*, pp. 270–287, 2018.

Pengsheng Guo, Chen-Yu Lee, and Daniel Ulbricht. Learning to branch for multi-task learning. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 3854–3863. PMLR, 13–18 Jul 2020.

Weituo Hao, Mostafa El-Khamy, Jungwon Lee, Jianyi Zhang, Kevin J Liang, Changyou Chen, and Lawrence Carin Duke. Towards fair federated learning with zero-shot data augmentation. In *CVPR*, pp. 3310–3319, June 2021.

Chenyu Huang, Peng Ye, Tao Chen, Tong He, Xiangyu Yue, and Wanli Ouyang. Emr-merging: Tuning-free high-performance model merging. In *NIPS*, volume 37. Curran Associates, Inc., 2024.

Wenke Huang, Mang Ye, Zekun Shi, He Li, and Bo Du. Rethinking federated learning with domain shift: A prototype view. In *CVPR*, pp. 16312–16322, 2023.

Yongqiang Huang, Zerui Shao, Ziyuan Yang, Zexin Lu, and Yi Zhang. Fedrir: Rethinking information representation in federated learning. In *Proceedings of the ACM on Web Conference (WWW)*, pp. 807–816, 2025.

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*, volume 35, pp. 7865–7873, 2021.

Shaoxiong Ji, Yue Tan, Teemu Saravirta, Zhiqin Yang, Yixin Liu, Lauri Vasankari, Shirui Pan, Guodong Long, and Anwar Walid. Emerging trends in federated learning: From model fusion to federated x learning. *International Journal of Machine Learning and Cybernetics*, 15(9): 3769–3790, 2024.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *ICML*, pp. 5132–5143, 2020.

Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *CVPR*, June 2018.

Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *CVPR*, pp. 10713–10722, 2021a.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *Proceedings of Machine Learning and Systems (MLSys)*, volume 2, pp. 429–450, 2020.

Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *ICML*, pp. 6357–6368, 2021b.

Wei-Hong Li, Steven McDonagh, Ales Leonardis, and Hakan Bilen. Multi-task learning with 3d-aware regularization. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *ICLR*, volume 2024, pp. 35258–35273, 2024.

Xiaoxiao Li, Meirui Jiang, Xiaofei Zhang, Michael Kamp, and Qi Dou. Fedbn: Federated learning on non-iid features via local batch normalization. In *ICLR*, 2021c.

hanxue liang, Zhiwen Fan, Rishov Sarkar, Ziyu Jiang, Tianlong Chen, Kai Zou, Yu Cheng, Cong Hao, and Zhangyang Wang. M^3vit: Mixture-of-experts vision transformer for efficient multi-task learning with model-accelerator co-design. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *NIPS*, volume 35, pp. 28441–28457. Curran Associates, Inc., 2022.

Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *NIPS*, volume 34, pp. 18878–18890. Curran Associates, Inc., 2021.

Bo Liu, Yihao Feng, Peter Stone, and Qiang Liu. Famo: Fast adaptive multitask optimization. In A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *NIPS*, volume 36, pp. 57226–57243. Curran Associates, Inc., 2023.

Shikun Liu, Edward Johns, and Andrew J. Davison. End-to-end multi-task learning with attention. In *CVPR*, June 2019.

Yuxiang Lu, Suizhi Huang, Yuwen Yang, Shalayiding Sirejiding, Yue Ding, and Hongtao Lu. Fedhca2: Towards hetero-client federated multi-task learning. In *CVPR*, pp. 5599–5609, 2024.

Yuxiang Lu, Shengcao Cao, and Yu-Xiong Wang. Swiss army knife: Synergizing biases in knowledge from vision foundation models for multi-task learning. In *ICLR*, 2025.

Othmane Marfoq, Giovanni Neglia, Aurélien Bellet, Laetitia Kameni, and Richard Vidal. Federated multi-task learning under a mixture of distributions. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *NIPS*, volume 34, pp. 15434–15447. Curran Associates, Inc., 2021.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 1273–1282, 2017.

Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-stitch networks for multi-task learning. In *CVPR*, June 2016.

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*, 2014.

Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

Aviv Navon, Aviv Shamsian, Idan Achituve, Haggai Maron, Kenji Kawaguchi, Gal Chechik, and Ethan Fetaya. Multi-task learning as a bargaining game. In *ICML*, volume 162, pp. 16428–16446, 2022.

Jaehoon Oh, Sangmook Kim, and Se-Young Yun. Fedbabu: Toward enhanced representation for federated image classification. In *ICLR*, 2022.

Sangjoon Park, Gwanghyun Kim, Jeongsol Kim, Boah Kim, and Jong Chul Ye. Federated split task-agnostic vision transformer for covid-19 cxr diagnosis. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *NIPS*, volume 34, pp. 24617–24630. Curran Associates, Inc., 2021.

Sashank J. Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and Hugh Brendan McMahan. Adaptive federated optimization. In *ICLR*, 2021.

Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *NIPS*, volume 31. Curran Associates, Inc., 2018.

Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet Talwalkar. Federated multi-task learning. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *NIPS*, volume 30, pp. 4424–4434. Curran Associates, Inc., 2017.

Ximeng Sun, Rameswar Panda, Rogerio Feris, and Kate Saenko. Adashare: Learning what to share for efficient deep multi-task learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *NIPS*, volume 33, pp. 8728–8740. Curran Associates, Inc., 2020.

S. Vandenhende, S. Georgoulis, W. Van Gansbeke, M. Proesmans, D. Dai, and L. Van Gool. Multi-task learning for dense prediction tasks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021.

Simon Vandenhende, BertDe Brabandere, and LucVan Gool. Branched multi-task networks: Deciding what layers to share. *British Machine Vision Conference*, 2019.

Simon Vandenhende, Stamatios Georgoulis, and Luc Van Gool. Mti-net: Multi-scale task interaction networks for multi-task learning. *ECCV*, 2020.

Xiaoye Wang, Chen Tang, Xiangyu Yue, and Wei-Hong Li. 3d-aware multi-task learning with cross-view correlations for dense scene understanding. In *CVPR*, 2026.

Yuan Wang, Huazhu Fu, Renuga Kanagavelu, Qingsong Wei, Yong Liu, and Rick Siow Mong Goh. An aggregation-free federated learning for tackling data heterogeneity. In *CVPR*, pp. 26233–26242, 2024.

Yipan Wei, Yuchen Zou, Yapeng Li, and Bo Du. Towards unified modeling in federated multi-task learning via subspace decoupling, 2025.

Yuqi Yang, Peng-Tao Jiang, Qibin Hou, Hao Zhang, Jinwei Chen, and Bo Li. Multi-task dense prediction via mixture of low-rank experts. In *CVPR*, 2024.

Hanrong Ye and Dan Xu. Inverted pyramid multi-task transformer for dense scene understanding. In *ECCV*, 2022.

Hanrong Ye and Dan Xu. Taskexpert: Dynamically assembling multi-task representations with memorial mixture-of-experts. In *ICCV*, pp. 21828–21837, October 2023a.

Hanrong Ye and Dan Xu. Taskprompter: Spatial-channel multi-task prompting for dense scene understanding. In *ICLR*, 2023b.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *NIPS*, volume 33, pp. 5824–5836. Curran Associates, Inc., 2020.

Teng Zhang, Yiqiang Chen, Xinlong Jiang, Wuliang Huang, Qian Chen, Chenlong Gao, Zhirui Wang, and Bingjie Yan. Fairfhtl: Achieving task-agnostic fairness in federated hetero-task learning. In *2025 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6, 2025.

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*, Jun 2019.

Rui Zhu, Liang Bai, Yanming Guo, Yirun Ruan, Tianyuan Yu, and Zhihe Lu. Controllable federated prompt learning at test time. In *CVPR*, pp. 39455–39465, June 2026.

A Appendix

A.1 Derivations in Section 3

In this section, we provide the detailed derivations deferred from Section 3: the closed-form solution of the guidance problem (Eq. (4)), the optimality of the consensus scale $\alpha_i = 1$, and the variational characterization of the conflict-strength weight β_i.

Closed-form solution of the guidance problem. Eq. (4) admits a closed-form solution through its dual (Liu et al., 2021). Write $U_w = \frac{1}{N}\sum_{i=1}^{N} w_i\,\Delta\theta_i$ for the convex combination of the uploaded increments with coefficients w over the simplex $\mathcal{W} = \{w : w_i \geq 0,\ \sum_{i=1}^{N} w_i = 1\}$, and let $\phi = c^2\left\|\Delta\bar{\theta}\right\|^2$. Since a linear function attains its minimum over a simplex at a vertex, the worst-case client can be replaced by the worst convex combination, $\min_i \langle \Delta\theta_i, \tilde{U}\rangle = N \min_{w\in\mathcal{W}} \langle U_w, \tilde{U}\rangle$; the two optimizations then commute because the objective is linear in $\tilde{U}$ and in w and both feasible sets are convex and compact, and for a fixed w Cauchy–Schwarz bounds the inner maximization over the ball, $\langle U_w, \tilde{U}\rangle \leq \langle U_w, \Delta\bar{\theta}\rangle + \sqrt{\phi}\,\|U_w\|$, with equality attained at the boundary point $\tilde{U} = \Delta\bar{\theta} + \sqrt{\phi}\,U_w/\|U_w\|$. The max-min problem therefore reduces to minimizing

$$F(w) = U_w^{\top}\Delta\bar{\theta} + \sqrt{\phi}\,\|U_w\| \tag{11}$$

over the simplex. Let $w^{\star}$ be the minimizer; the guidance vector is then recovered as the fixed-radius step from the mean update in Eq. (5) of the main text.

Optimality of the consensus scale $\alpha_i = 1$. We formalize the argument in the main text that any $\alpha_i \neq 1$ is a detour.

Definition. Let q range over the consensus coordinates of client C_i, with the same meaning as in Eq. (2). Write $\tilde{U}_{i,q} = M_i \odot \tilde{U}$ and $\Delta_{i,q} = M_i \odot \Delta\theta_i$ for the guidance and the client's own update restricted to these coordinates. For a scaled candidate $\alpha\,\tilde{U}_{i,q}$ ($\alpha \geq 0$), define the cost

$$J(\alpha) = \left\|\alpha\tilde{U}_{i,q} - \tilde{U}_{i,q}\right\| + \left\|\alpha\tilde{U}_{i,q} - \Delta_{i,q}\right\|, \tag{12}$$

the sum of the candidate's distances to the guidance and to the client's own update.

Assumption. $\tilde{U}_{i,q} \neq \mathbf{0}$ and $\Delta_{i,q} \neq \mathbf{0}$, and $\Delta_{i,q}$ is not collinear with $\tilde{U}_{i,q}$. Non-collinearity holds because $\tilde{U}$ is a conflict-averse correction of the mean update rather than a copy of any client's update; by the mask construction the two vectors also agree elementwise in sign, so their angle is acute.

Proposition. Under the assumption, $J(\alpha) > J(1)$ for every $\alpha \neq 1$; that is, $\alpha = 1$ is the unique minimizer of J.

Proof. At $\alpha = 1$ the first term vanishes, so $J(1) = \|\tilde{U}_{i,q} - \Delta_{i,q}\|$ is the direct distance between the two references. For any $\alpha \neq 1$, the first term equals $|\alpha - 1|\,\|\tilde{U}_{i,q}\| > 0$, and the triangle inequality gives

$$J(\alpha) \;\geq\; \left\|\tilde{U}_{i,q} - \Delta_{i,q}\right\| \;=\; J(1), \tag{13}$$

i.e., however much the second term shrinks as α slides along the guidance direction, the growth of the first term at least compensates it. Equality in the triangle inequality would require the three points to be collinear, which the assumption rules out; Eq. (13) is therefore strict for every $\alpha \neq 1$: any deviation from $\alpha = 1$ is a detour strictly longer than the straight path. Thus $\alpha = 1$ is the unique minimizer, justifying the fixed scale $\alpha_i = 1$.

Variational characterization of the conflict-strength weight. Let $P_i = \bar{M}_i \odot \Delta\theta_i$ and $u_P = \bar{M}_i \odot \tilde{U}$ be as defined in Section 3. The weight β_i in Eq. (10) is the unique minimizer of a per-client trade-off. Consider the quadratic objective

$$\mathcal{J}_i(w) = (1-w)^2\,\|P_i\| + w^2\,\|u_P\|, \tag{14}$$

whose first term penalizes discarding the client's own conflicting component and whose second term penalizes keeping a conflict against the guidance. The objective is strictly convex on $[0,1]$ whenever $\|P_i\| + \|u_P\| > 0$. Setting its derivative to zero,

$$(1-w)\,\|P_i\| \;=\; w\,\|u_P\|, \tag{15}$$

yields the unique minimizer

$$w^* = \frac{\|P_i\|}{\|P_i\| + \|u_P\|}. \tag{16}$$

In practice, we use Eq. (10). Equation (15) admits a direct interpretation: at the optimum, the marginal penalty for discarding the client's update equals the marginal penalty for retaining the conflict, so moving in either direction only increases the total cost.

A.2 Experimental setups

In this section, we provide more details about our experimental implementation in addition to those discussed in the paper.

Datasets and data augmentation We conduct the experiments on two multi-task datasets: PASCAL-Context (Mottaghi et al., 2014) and NYUD-v2 (Nathan Silberman & Fergus, 2012). We take the same data augmentation as (Lu et al., 2024). We use random scaling with factor between 0.5 and 2.0, random horizontal flipping, and random color jittering. Image normalization is also applied throughout both the training and evaluation phases. For NYUD-v2, all images are resized to 448 × 576 resolution as in (Ye & Xu, 2022). For PASCAL-Context, we resize the images to 512 x 512 pixels

Loss functions and task weight Following previous work (Lu et al., 2024), we employ task-specific loss functions. For semantic segmentation, human parts segmentation, and saliency detection, we use the cross-entropy loss. Surface normal estimation and monocular depth estimation are optimized using the L1 loss. For edge detection, we use the weighted binary cross-entropy loss, with a weight of 0.95 to positive pixels and 0.05 to negative ones. For task-specific losses, we refer to previous studies (Vandenhende et al., 2021; Ye & Xu, 2022; Lu et al., 2024). For NYUD-v2 dataset, the loss weights for SemSeg, Depth, Normals, and Edge tasks are set to 1, 1, 10, and 50, respectively. For PASCAL-Context, Semseg, Parts, Sal, Normals, and Edge tasks are set to 1, 2, 5, 10, and 50, respectively.

Training details Experiments were conducted on two NVIDIA RTX 4090 GPUs. All experiments were repeated three times with different random seeds, and the average results are reported to ensure statistical reliability.

Evaluation metrics. Δ_m is defined as the average performance improvement per-task relative to a local baseline: $\Delta_m = \frac{1}{T}\sum_{t=1}^{T} M_t$, where T is the number of evaluated task–client pairs. $M_t = (-1)^{l_t}(M_{m,t} - M_{b,t})/M_{b,t}$ represents the performance metric for task t, and l_t is a binary indicator, equal to 1 if a lower metric value signifies better performance for task t, and 0 otherwise.

A.3 More results and analysis

In this section, we provide more detailed results and analysis of our method and recent SotA.

Results over multiple runs. Here, we report the results of our method over 3 runs on PASCAL-Context and NYUD-v2 and report the results in Table 5. From the results, we can see that our method is stable (i.e., the standard deviations of the tasks on each client are very small) and improves over the SotA method on seven of the nine metrics.

Table 5: Mean and standard deviation of FedHCA^2 and CAUD over 3 runs on PASCAL-Context (ST) and NYUD-v2 (MT).

Method	PASCAL-Context **(ST)**					NYUD-v2 **(MT)**			
	SemSeg↑	Parts↑	Sal↑	Normals↓	Edge↑	SemSeg↑	Depth↓	Normals↓	Edge↑
FedHCA^2	57.55	52.30	80.71	**15.60**	72.08	41.47	0.6281	**20.53**	76.50
CAUD (ours)	**61.92±0.42**	**53.09±0.26**	**80.88±0.11**	15.84±0.09	**72.11±0.12**	**44.30±0.35**	**0.6066±0.0020**	20.73±0.07	**76.58±0.08**

Multi-task clients on the full task set. When all clients are multi-task learners trained on the full task set, local multi-task training alone is already strong, so the margins narrow; CAUD still attains $\Delta_m = 0.71\%$ versus 0.38% for FedHCA^2, indicating that the conflict-averse modulation extracts beneficial cross-client knowledge without introducing harmful interference in this regime as well.

Table 6: Comparison of multi-task clients trained on the full task set across NYUD-v2 and PASCAL-Context.

Method	PASCAL-Context (**MT**)					NYUD-v2 (**MT**)				$\Delta_m\% \uparrow$
	SemSeg↑	Parts↑	Sal↑	Normals↓	Edge↑	SemSeg↑	Depth↓	Normals↓	Edge↑	
Local	**65.80**	55.01	83.23	14.21	71.89	**41.86**	0.6487	20.59	76.46	0.00
FedHCA^2	65.07	**55.89**	83.23	14.21	72.06	41.56	0.6293	**20.57**	76.52	0.38
CAUD (ours)	65.65	55.71	**83.24**	**14.17**	**72.11**	41.51	**0.6124**	20.63	**76.60**	**0.71**

Results with NYUD-v2 (ST) & PASCAL-Context (MT) setting. Table 7 reports the complementary setting with NYUD-v2 single-task clients and a PASCAL-Context multi-task client. CAUD attains a Δ_m of 2.44%, outperforming all compared methods, indicating that the advantage of conflict-averse decoupling generalizes across the two complementary client arrangements.

Table 7: 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
Traditional Federated Learning										
FedAvg	25.80	0.8295	24.85	75.31	64.63	52.88	81.08	15.56	68.95	-7.56
FedProx	25.96	0.8316	25.20	75.34	64.97	50.78	81.29	15.83	69.81	-8.12
Personalized Federated Learning										
FedPer	35.93	0.7460	23.75	75.53	67.78	54.75	82.50	14.75	71.90	-0.16
Ditto	28.15	0.7482	23.96	75.42	65.99	51.45	81.74	15.29	69.96	-4.67
FedAMP	34.75	0.7103	23.31	75.03	66.08	54.10	**83.35**	14.20	71.88	0.27
Multi-task Learning										
CAGrad+	35.84	0.6980	23.76	75.56	67.32	55.10	82.70	14.65	71.88	0.66
FairGrad+	36.54	0.6990	23.91	**75.71**	67.50	54.78	82.68	14.89	72.00	0.62
Model Merging										
EMR-Merging+	36.71	0.6846	22.93	75.14	65.61	**55.83**	82.64	14.23	72.03	1.76
Federated Multi-task Learning										
MaT-FL	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
FedDEA	32.99	0.7014	23.56	75.33	65.21	54.77	83.10	14.49	71.41	-0.59
FairFHTL	37.26	0.6898	24.20	75.69	66.51	53.81	82.28	15.51	71.68	-0.25
CAUD (ours)	**37.67**	**0.6702**	**22.90**	75.36	67.45	55.56	83.13	14.47	**72.05**	**2.44**

Results during training. Fig. 4 compares the per-round evaluation curves of FedHCA^2 and CAUD on the PASCAL-Context (ST) and NYUD-v2 (MT) setting, where the models are evaluated every four communication rounds. CAUD consistently outperforms FedHCA^2 on all three segmentation metrics throughout training, and the margin widens as communication proceeds. Notably, the advantage already appears at the very first evaluation on NYUD-v2 SemSeg, suggesting that the conflict-averse guidance accelerates the early-stage convergence of the multi-task client. Overall, these curves show that the advantage of CAUD accumulates progressively over communication rounds rather than arising from a late-stage artifact, and that the gains are consistent across both the single-task and the multi-task clients.

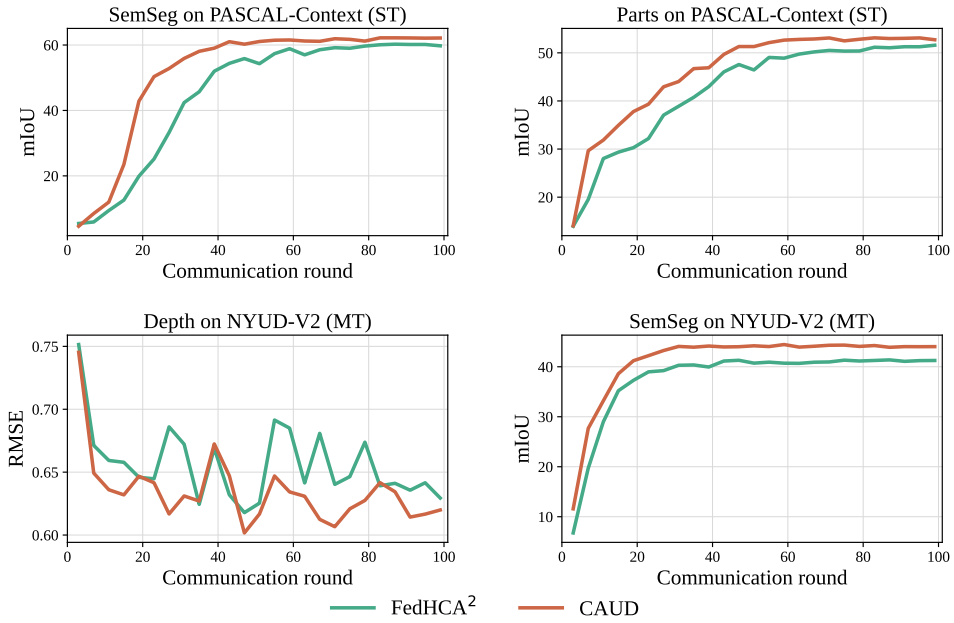


Figure 4: Evaluation curves of FedHCA^2 and CAUD on the PASCAL-Context (ST) and NYUD-v2 (MT) setting. Models are evaluated every four communication rounds. mIoU is reported for SemSeg and Parts, and RMSE for Depth.

Sensitivity to the ball radius. Fig. 5 sweeps the ball radius c in Eq. (4), which bounds how far the guidance vector can deviate from the mean update. Sweeping $c \in [0.2, 0.6]$, Δ_m rises from 3.87% at $c = 0.2$ to a peak of 4.18% at $c = 0.4$, then drops to 4.06% at $c = 0.5$ and 4% at $c = 0.6$. Both an overly small and an overly large radius thus degrade performance: the former keeps the guidance too close to the mean update and limits its ability to correct conflicting directions, while the latter lets the guidance deviate too far and reintroduces interference. We therefore adopt $c = 0.4$ throughout, following the value used for NYUD-v2 in (Liu et al., 2021). Notably, Δ_m stays within a narrow band of 3.87%–4.18% across the entire swept range and always remains well above the strongest baseline, indicating that CAUD is not highly sensitive to the choice of c.

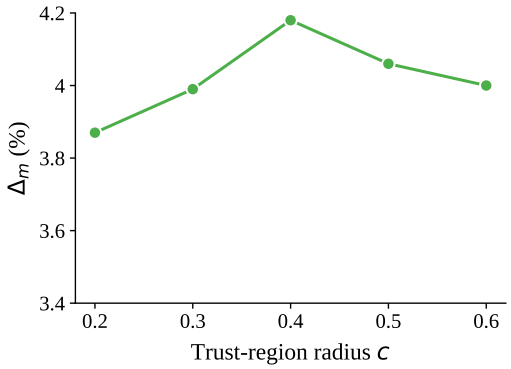


Figure 5: Sensitivity of Δ_m to the ball radius c on the PASCAL-Context (ST) and NYUD-v2 (MT) setting.