SUM: Unified Geometric Surgery on Spatio-Temporal Adaptation Vectors for Federated Class Incremental Learning

Jaeik Kim[1] and Jaeyoung Do[1,2⋆]

AIDAS Lab, [1]IPAI & [2]ECE, Seoul National University, Seoul, Republic of Korea
{jake630,jaeyoung.do}@snu.ac.kr

Abstract. Real-world intelligent systems often require both distributed collaboration across data-isolated clients and continual adaptation to evolving tasks. This setting naturally gives rise to Federated Class Incremental Learning (FCIL), which combines Federated Learning (FL) and Continual Learning (CL). However, their combination introduces two coupled sources of interference: spatial interference from heterogeneous clients and temporal interference from sequential tasks, jointly leading to Spatial–Temporal Catastrophic Forgetting (ST-CF). Existing approaches typically address spatial and temporal interference with separate mechanisms, often incurring additional client-side computation or communication, while leaving directional interactions among updates during aggregation unregulated. In this paper, we reinterpret FCIL as a unified multi-task learning problem, where both client and task updates are represented as adaptation vectors in a shared parameter space. Based on this view, we propose SURGERY & MERGE (SUM), a purely server-side framework that performs geometric surgery on adaptation vectors during aggregation. Spatial SUM mitigates client-level interference within each round, while causal online temporal SUM removes cross-task interference over time without additional client-side computation, communication, or memory beyond standard federated training. Empirically, SUM achieves up to 22% improvement over prior FCIL methods across diverse vision and language benchmarks while remaining robust to unreliable clients and maintaining computational efficiency.

1 Introduction

Federated Learning (FL) enables collaboration across distributed clients without sharing private data [35,56], while Continual Learning (CL) allows models to adapt to evolving tasks over time [18,46]. In many real-world settings, these capabilities are required simultaneously: clients must learn sequential tasks from heterogeneous, non-independent and identically distributed (non-IID) data streams while keeping data local. Federated Class-Incremental Learning (FCIL) therefore emerges as a practical paradigm for lifelong federated intelligence [6, 61].

⋆ Corresponding author.

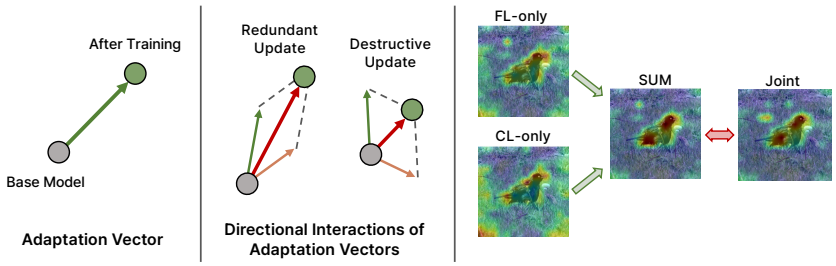


Fig. 1: Left: Client or task updates during FCIL are represented as adaptation vectors in parameter space from the base model. **Middle:** Directional interactions among these vectors may lead to redundant or destructive updates. **Right:** Rather than resolving these interactions along a single axis (FL-only or CL-only), SUM unifies both axes and regulates them, producing representations comparable to joint training.

However, combining FL and CL introduces tightly coupled sources of interference: *spatial interference* caused by client drift under non-IID data [17], and *temporal interference* caused by catastrophic forgetting across tasks [34]. These effects reinforce each other, forming spatial–temporal catastrophic forgetting (ST-CF) [55,59]. Consequently, mitigating only one axis often leaves residual interference from the other.

During FCIL, updates from heterogeneous clients and sequential tasks are repeatedly combined in the shared parameter space. Their directional interactions may produce *redundant* or *destructive* updates (Fig. 1), leading to degraded optimization. Similar directional effects have been extensively studied in multi-task learning (MTL), where gradients from different tasks may point in incompatible directions, resulting in degraded generalization [27,44,52,60]. However, existing FCIL methods mitigate ST-CF primarily through FL-oriented alignment [6], CL-style regularization [18], replay [1,31,61], or prototype anchoring [8,59]. Although some works address spatio-temporal interference jointly [3,37,58,59], most methods regulate update generation during local optimization through constraints, auxiliary objectives, or data, while directional interactions among updates during aggregation remain largely unregulated.

In this paper, we revisit FCIL from a unified MTL perspective, where both client and task updates are represented as adaptation vectors—interpretable as gradient-like directions—in the shared parameter space. This view exposes their directional interactions explicitly and enables analysis of these interactions at aggregation time while preserving standard parameter communication [35]. From this perspective, ST-CF can be understood as the cumulative effect of unresolved directional interactions: spatial drift introduces heterogeneous update directions across clients, while sequential tasks introduce interfering directions over time.

Building on this insight, we propose SURGERY & MERGE (SUM), a purely server-side aggregation-time refinement framework that performs geometric surgery on adaptation vectors before merging. A parallel spatial SUM resolves client-level interactions within each round, while a causal online temporal SUM regu-

Table 1: Comparison with representative FCIL methods.

Method	Client-side overhead				Property	
	Extra objective	Replay	Extra memory	Training modification	Server-only	Interaction modeling
GLFC [6]	✓	✓	✓	✓	×	×
TARGET [61]	✓	✓	✓	✓	×	×
MFCL [1]	✓	✓	×	✓	×	×
PILoRA [8]	×	×	✓	✓	×	×
FOT [3]	×	×	✓	✓	×	×
STAMP [37]	✓	✓	✓	✓	×	×
LoRM [41]	×	×	✓	×	×	×
SUM (ours)	×	×	×	×	✓	✓

lates cross-task interactions over time. Importantly, SUM operates entirely at the server during aggregation, introducing no additional client computation, communication, or memory beyond standard federated training. Extensive experiments on diverse vision and language benchmarks show that SUM improves final averaged accuracy by up to 22% over prior FCIL methods, remains robust under malicious-client settings, and maintains competitive computational overhead compared to existing FCIL approaches.

2 Related Works

Federated Class Incremental Learning. Federated Class Incremental Learning (FCIL) combines Federated Learning (FL) and Continual Learning (CL), yet most existing methods address spatial drift and temporal forgetting separately. Methods such as GLFC [6], CFeD [31], and TARGET [61] extend FedAvg [35] with rehearsal or distillation, while parameter-efficient variants (*e.g.* FedWeIT [57], PILoRA [8], LoRM [41], Fed-CPrompt [2]) introduce task- or client-specific modules. Recent FCIL methods also use regularization, replay, stabilization, or distillation to preserve previous-task knowledge [1,23–25,59]. Other works, including FOT [3] and STAMP [37], reduce interference through projection or gradient-alignment mechanisms during local optimization: FOT constrains updates using activation-derived subspaces at task boundaries, while STAMP enforces gradient alignment during client training. As a result, these methods regulate update generation largely during local training or through auxiliary knowledge-preservation mechanisms, but do not explicitly model how client and task updates interact when accumulated on the server. In contrast, SUM operates purely at aggregation time, treating updates as adaptation vectors and explicitly decomposing their directional interactions before merging. This design requires no modification to client-side training, since refinement is applied only to vectors produced by standard local training and communication. Tab. 1 summarizes the key differences between SUM and representative FCIL approaches.

Directional Gradient Interactions. In Multi-Task Learning (MTL), directional interactions among task gradients are a well-known source of redundant or destructive interference that can destabilize optimization and degrade generaliza-

tion. Many works therefore mitigate such interference by explicitly manipulating gradients during optimization, including PCGrad [60], MGDA [43], CAGrad [27], GEM [29], Aligned-MTL [44], and GPM [40]. These methods stabilize training by projecting gradients or enforcing compatible descent directions across tasks. In FCIL, STAMP [37] follows a similar philosophy by performing spatio-temporal gradient matching during client optimization. In contrast, SUM resolves interference after local training by refining the geometry of communicated adaptation vectors at server aggregation time, rather than relying on centralized optimization-time gradients or modifying client-side training. This preserves the standard federated communication and training protocol.

Adaptation Vectors and Model Merging. Model merging studies how independently trained task updates can be represented and composed in a shared parameter space. In this context, task-specific updates are often represented as *adaptation vectors*, defined as parameter differences between a base model and a task-adapted model. Task Arithmetic [15] introduced this formulation, later extended by TIES-Merging [53] and EMR-Merging [13], and applied to continual (*e.g.* MagMax [32]) and federated settings (*e.g.* FedAWA [45], FedMerge [5]). These approaches focus on parameter selection or weighted aggregation, but do not explicitly decompose directional interactions among coexisting adaptation vectors. In FCIL, LoRM [41] computes closed-form spatio-temporal merging weights, yet still treats task vectors primarily as additive components. In contrast, SUM models both client- and task-induced updates as interacting adaptation vectors, explicitly handling within-round client coupling and cross-task accumulation through a unified server-side refinement protocol.

3 Foundations

3.1 FCIL as a Unified Multi-Task Learning Problem

We view Federated Class Incremental Learning (FCIL) through a multi-task learning (MTL) perspective. Federated Learning corresponds to *spatial MTL*, where heterogeneous clients optimize distinct objectives under shared parameters, while Continual Learning corresponds to *temporal MTL*, where tasks arrive sequentially under the same model [32, 40]. FCIL naturally combines both structures: objectives vary across clients and evolve across tasks within a shared parameter space. A formal joint objective and its equivalence to this unified MTL view are provided in Appendix B.1. Under this view, interference arises when updates from different objectives interact through aggregation, which prior MTL studies attribute largely to directional relationships among gradients [27, 60].

3.2 Adaptation Vectors in a Shared Parameter Space

To formalize the directional perspective inspired by MTL, we analyze update interactions directly in the parameter space. In FCIL, clients exchange model

parameters rather than gradients, making the parameter space the natural domain for analyzing update interactions. We therefore represent both client- and task-induced updates as *adaptation vectors* in the parameter space of the shared model. Intuitively, an adaptation vector corresponds to the parameter change between two model states (*e.g.* before and after training) [15, 53]. Since these updates are produced by gradient-based optimization, they capture the accumulated effect of gradient directions over local training steps or sequential tasks (Appendix B.2). This allows directional interactions to be analyzed through the resulting adaptation vectors, which are actually merged during federated and continual aggregation. We first define the client and task adaptation vectors.

Client Adaptation (Spatial Axis). At communication round t, the server broadcasts θ_G^t. Each client i performs K_i steps of local training and returns the updated parameters θ_i^{t,K_i}. We define the client adaptation vector τ_i^t as:

$$\tau_i^t := \theta_i^{t,K_i} - \theta_G^t. \tag{1}$$

Using this adaptation vector, standard FedAvg [35] can be written as

$$\theta_G^{t+1} = \theta_G^t + \sum_{i \in \mathcal{S}_t} \lambda_i \tau_i^t. \tag{2}$$

Task Adaptation (Temporal Axis). Let $\theta_G^{(k)}$ denote the global model obtained after completing training rounds on task k. Across tasks, we define an *accumulated* task adaptation vector τ_k, relative to a common pretrained base θ_{pre}:

$$\tau_k := \theta_G^{(k)} - \theta_{\text{pre}}. \tag{3}$$

Rather than measuring updates relative to the immediately preceding model, we anchor all tasks to the same pretrained base. This shared reference places adaptation vectors in a common parameter frame, which both satisfies the conditions of our theoretical analysis (Thm. 1) and enables explicit regulation of their directional interactions. As we show in Sec. 4.3, operating on these accumulated vectors is equivalent to resolving the interactions among the underlying true sequential task updates.

Unifying Perspective. Both client updates τ_i^t and task updates τ_k reside in the same shared parameter space and accumulate through repeated aggregation. Their directional interactions therefore shape the evolution of the global model. Across communication rounds, heterogeneous clients introduce diverse update directions, while sequential tasks introduce additional directions over time. From MTL perspective, when such directional interactions are not explicitly regulated, their accumulated effects can distort shared representations [27, 60], providing a geometric view of spatial–temporal catastrophic forgetting (ST-CF). This perspective motivates our proposed SUM introduced in Sec. 4.

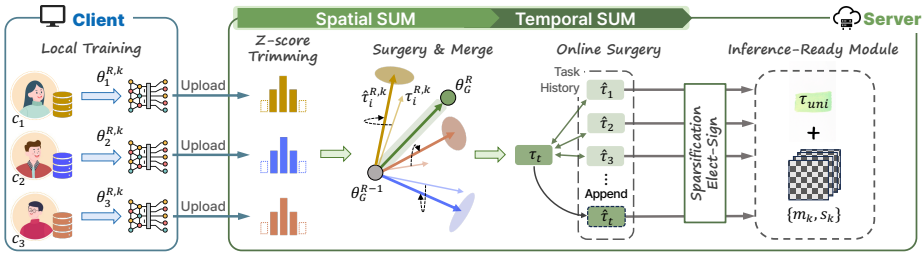


Fig. 2: Overview of Surgery & Merge (Sum). After local training for task k over R communication rounds, clients upload their updated weights to the server. The server then applies spatial and temporal Sum to refine the adaptation vectors and construct an inference-ready module for task k. Notably, Sum introduces no additional client-side overhead beyond standard federated model training and communication.

4 Method

Building on Sec. 3, we present a unified framework for resolving spatio-temporal interference in Federated Class Incremental Learning (FCIL). We first introduce our unified projection principle.

4.1 Surgery & Merge

Overall Pipeline. As shown in Fig. 2, Sum operates in three stages across tasks. The server applies Spatial SUM to remove directional interference among client adaptation vectors, and Temporal SUM to regulate interactions with previously learned tasks. The resulting task vectors are stored as a temporal basis and converted into a compact inference module after each task. The resulting module is used for inference, while subsequent local training proceeds from the spatially aggregated model.

Core Idea. Before introducing our refinement, we denote the standard aggregation result as

$$\theta_{\text{standard}} = \theta_{\text{base}} + \sum_i \lambda_i \mathbf{v}_i, \tag{4}$$

where θ_{base} denotes the anchor model (*e.g.* θ_G^t or θ_{pre}) and $\mathbf{v}_i$ denotes an adaptation vector produced during training (*e.g.* a client update τ_i^t or a task update τ_k). In FCIL, the server aggregates multiple adaptation vectors produced by heterogeneous clients or sequential tasks using Eq. 4. When these vectors are combined through naive summation, their directional interactions may produce *redundant* updates or *destructive* updates, which distort the resulting representation.

Our goal is therefore to regulate these directional interactions during aggregation. Given adaptation vectors $\{\mathbf{v}_i\}$, Sum refines each vector by removing components aligned with others, retaining only its independent contribution in

the shared parameter space. Formally, we compute a refined vector

$$\hat{\mathbf{v}}_i = \mathbf{v}_i - \sum_{\substack{j=1 \\ j \neq i}}^{n} \frac{\mathbf{v}_i \cdot \mathbf{v}_j}{\|\mathbf{v}_j\|_2^2} \mathbf{v}_j. \tag{5}$$

The resulting model is then obtained as

$$\theta_{\text{SUM}} = \theta_{\text{base}} + \sum_{i=1}^{n} \lambda_i \hat{\mathbf{v}}_i. \tag{6}$$

Intuitively, this refinement reduces directional interference among adaptation vectors, preventing overlapping or conflicting updates from dominating the aggregated model. Importantly, SUM does not discard entire client or task updates; it removes only the components that are directionally coupled with other vectors, thereby preserving residual client- or task-specific directions. This interpretation is empirically supported by our surgery-target ablation in Fig. 3, where one-sided suppression of selected interactions is insufficient, whereas SUM preserves useful residual directions and achieves the best performance. Because the refinement operates directly in the parameter space, SUM can be applied once at aggregation on the server side without modifying client optimization. Unlike optimization-time methods [3,6,61], it requires no gradient storage or additional forward/backward passes.

We further provide a local first-order analysis showing when SUM can improve the aggregation-level training descent bound under standard smoothness assumptions [27, 60].

Theorem 1. *Let θ_{SUM} (Eq. 6) and $\theta_{standard}$ (Eq. 4) be models obtained from refined and standard merging, respectively. If all $\mathbf{v}_i$ share the common base θ_{base} and task losses are convex with L-Lipschitz gradients at θ_{base}, then:*

$$\mathcal{L}(\theta_{\text{SUM}}) \leq \mathcal{L}(\theta_{standard}).$$

See Appendix B.4 for the proof. This result should be interpreted as an aggregation-level local training-objective statement under the stated common-base and smoothness assumptions, rather than as a universal test-loss or global convergence guarantee. It formalizes a sufficient condition under which reducing directional interference yields a tighter local descent bound than naive aggregation. Generalization behavior is therefore evaluated empirically through held-out Final Averaged Accuracy (FAA), forgetting, sharpness, and class-margin analyses (Sec. 5, Appendix D).

4.2 Spatial SUM

In FL, client adaptation vectors τ_i^t coexist within each communication round and often diverge due to non-IID data. Spatial SUM instantiates Eq. 6 on the

set of client adaptation vectors at round t:

$$\theta_G^{t+1} = \theta_G^t + \lambda_S \sum_{i \in \mathcal{S}} \hat{\tau}_i^t, \tag{7}$$

where λ_S is a scaling factor for spatial aggregation. Before applying Eq. 7 to the client adaptation vectors, we suppress extreme coordinates via Z-score trimming. For each coordinate index p, we zero entries with $|z[p]| > z_{\text{thr}}$, where $z[p] = (\tau[p] - \mu)/\sigma$ denotes the Z-score of the p-th parameter coordinate. Without this step, outlier coordinates may disproportionately dominate the inner products used in Eq. 5. This can make the projection excessively large or even flip its direction, resulting in unstable refinement updates (Tab. 5, Appendix D.3). Classifier heads are merged using RegMean [16].

System and Overhead. Clients perform standard local training and upload model updates exactly as in conventional FL; all SUM operations are executed on the server, introducing no additional client-side computation or communication.

4.3 Temporal SUM

After the final FL communication round R for task k, the server obtains $\theta_G^R = \theta_G^{(k)}$ via the Spatial SUM and proceeds to Temporal SUM. Let the true sequential updates be $\Delta_k = \theta_G^{(k)} - \theta_G^{(k-1)}$. Although $\{\Delta_k\}$ capture the true learning dynamics of task k, each update is defined relative to a different base model $\theta_G^{(k-1)}$. Applying Eq. 5 directly to these updates therefore violates the common-base assumption required by Thm. 1, which assumes that all vectors are defined relative to a shared base model θ_{base}. To satisfy this condition, we re-anchor task adaptations to a common base model:

$$\tau_k = \theta_G^{(k)} - \theta_{\text{pre}}. \tag{8}$$

Online Surgery. As tasks arrive sequentially in FCIL, we perform an online projection that incrementally handles directional interactions along previously stored task adaptation vectors.

$$\hat{\tau}_k = \tau_k - \sum_{j=1}^{k-1} \frac{\tau_k \cdot \hat{\tau}_j}{\|\hat{\tau}_j\|_2^2} \hat{\tau}_j, \quad (\hat{\tau}_1 = \tau_1). \tag{9}$$

This online form is a sequential implementation of the same projection principle used in Eq. 5. As discussed in Appendix B.5, it admits the same local descent-bound interpretation under the stated assumptions. The following proposition shows that performing projection on the accumulated vectors $\{\tau_k\}$ is equivalent to resolving directional interactions among the true sequential task updates $\{\Delta_k\}$ (see Appendix B.6 for the proof).

Proposition 1. *Applying online surgery to $\{\tau_k\}$ produces the identical orthogonal vectors as applying the same procedure directly to $\{\Delta_k\}$.*

System and Overhead. Temporal SUM is executed entirely on the server using the uploaded model weights after each task. The server computes τ_k, performs the projection, and stores the vectors $\{\hat{\tau}_k\}$.

4.4 Inference-ready Module Construction

After temporal SUM, each $\hat{\tau}_k$ captures the interference-free adaptation of task k. While client training continues from the global model $\theta_G^{(k)}$ for the next task, we construct compact inference modules for deployment. Specifically, we convert $\{\hat{\tau}_k\}$ into lightweight task-specific modules using a modified EMR-Merging [13].

Step 1: Sparsification. Each $\hat{\tau}_k$ is sparsified via magnitude-based top-k selection, retaining the largest k_{pct} fraction of coordinates. Let Ω_k denote the set of indices corresponding to the largest k_{pct} fraction of $|\hat{\tau}_k|$:

$$\bar{\tau}_k[p] = \begin{cases} \hat{\tau}_k[p], & p \in \Omega_k, \\ 0, & \text{otherwise.} \end{cases}$$

In practice, we find that retaining only these high-magnitude coordinates is sufficient to preserve the dominant task-specific adaptation patterns (Fig. 6).

Step 2: Elect-Sign. From $\bar{\mathcal{B}} = \{\bar{\tau}_1, \ldots, \bar{\tau}_k\}$, we derive a unified direction τ_{uni} using a coordinate-wise sign-consensus rule. For each parameter coordinate, we determine the majority-supported sign across tasks and assign its magnitude using the largest aligned value. Coordinates whose signs disagree across tasks are set to zero. This retains directions that are consistently aligned across tasks while removing conflicting updates.

Step 3: Task-Specific Activation. To preserve task-specific behavior, each task activates a subset of the unified direction. We construct a binary mask m_k where $m_k[p] = 1$ if $\tau_k[p]\tau_{\text{uni}}[p] > 0$ and 0 otherwise. A scalar factor $s_k = \frac{\|\tau_k\|_1}{\|m_k \odot \tau_{\text{uni}}\|_1 + \varepsilon}$ rescales the activation to match the original task magnitude, where ε is a small constant for numerical stability. The inference model is reconstructed as: $\theta_{\text{infer}}^{(k)} = \theta_{\text{pre}} + m_k \odot (s_k \cdot \tau_{\text{uni}})$. The server distributes the compact module $(\tau_{\text{uni}}, m_k, s_k)$, while the shared pretrained backbone θ_{pre} is already available on clients.

5 Experiments

5.1 Experimental Settings

We follow a standard FCIL setup [41,42], using the same pure ImageNet-pretrained ViT-B/16 [7] backbone for vision tasks (Sec. 5.2) and T5-Small [39] for language tasks (Sec. 5.3). Unless stated otherwise, experiments use 10 clients, 10 sequential class-split tasks, and 5 communication rounds per task. Client data follow a Dirichlet label-imbalance partition with $\beta \in \{0.5, 0.1, 0.05\}$ or $\{1.0, 0.5, 0.2\}$

Table 2: Evaluation on various vision-domain datasets in FAA (↑). Bold indicates the best and underlined the second-best results.

	CIFAR-100			ImageNet-R			ImageNet-A			EuroSAT			CARS-196			CUB-200		
Joint	**92.75**			**84.02**			**54.64**			**98.42**			**85.62**			**86.04**		
Distrib. β	0.5	0.1	0.05	0.5	0.1	0.05	1.0	0.5	0.2	1.0	0.5	0.2	1.0	0.5	0.2	1.0	0.5	0.2
EWC	78.46	72.42	64.51	58.93	48.15	43.68	10.86	10.07	8.89	64.12	59.30	56.52	19.55	18.02	18.29	31.46	29.60	27.89
LwF	62.87	55.56	47.09	54.03	41.02	46.07	8.89	8.89	7.90	31.91	21.26	31.42	20.84	22.72	31.76	25.25	21.11	18.54
FisherAvg	76.10	74.43	65.31	58.68	50.82	47.33	11.59	11.06	10.14	58.84	59.94	55.86	26.03	24.60	21.58	30.45	28.39	25.06
RegMean	59.80	45.88	39.08	61.18	57.00	55.80	8.56	6.22	4.34	48.74	51.73	45.27	21.83	20.36	15.92	35.57	32.84	32.83
CCVR	79.95	75.14	65.30	70.00	62.60	60.38	39.50	36.27	35.94	64.44	57.93	62.69	38.99	37.81	35.31	62.67	59.48	56.33
L2P	83.88	61.54	55.00	42.08	23.85	16.98	20.14	17.31	16.85	40.63	51.78	45.46	35.49	31.00	20.01	56.23	47.31	38.16
CODA-P	82.25	61.82	46.74	61.18	36.73	25.82	18.30	14.48	7.31	73.38	69.42	66.69	28.04	20.83	14.53	42.53	37.71	29.19
FedProto	75.79	70.02	60.55	58.52	47.30	52.93	9.87	9.22	10.01	58.79	62.85	64.17	26.08	24.55	22.75	30.22	28.27	26.01
TARGET	74.72	72.32	62.60	54.65	45.83	41.32	10.27	11.39	10.73	52.74	52.74	45.11	28.65	27.20	26.13	39.30	38.40	34.79
PILoRA	76.48	75.81	74.80	53.67	51.62	49.37	19.62	18.70	20.01	48.35	32.89	31.22	37.57	37.92	36.95	61.11	60.68	60.39
FOT	79.26	76.45	69.89	68.65	59.92	55.12	18.37	17.25	15.01	59.58	57.07	48.07	33.53	33.49	30.57	45.88	41.78	41.28
LoRM	86.95	81.75	82.76	72.48	63.83	66.45	37.26	36.34	33.11	84.23	77.26	81.36	54.41	51.87	48.81	64.60	63.67	60.06
Sum	**97.03**	**96.93**	**96.32**	**83.80**	**85.58**	**86.81**	**49.90**	**50.82**	**47.86**	**96.51**	**93.28**	**98.57**	**68.78**	**68.83**	**67.57**	**81.69**	**79.79**	**79.43**

depending on dataset scale. We report Final Averaged Accuracy (FAA) after the last task. For Sum, we set $z_{\text{thr}} = 4.5$, $\lambda_S = 0.4$, and $k_{pct} = 5\%$. A centralized *Joint* baseline is included. All baselines are evaluated under the same FCIL protocol [41], using the same backbone, task splits, client partitions, communication rounds, and data-access constraints. Method-specific hyperparameters are selected following the original recommendations and reported in Appendix C.1.

5.2 Vision Domain

Datasets. We evaluate Sum on six datasets spanning varying proximity to ImageNet. Following prior work [41], we group them into three categories: in-domain (CIFAR-100 [20], ImageNet-R [10], ImageNet-A [11]), fine-grained (Cars-196 [19], CUB-200 [50]), and out-of-domain (EuroSAT [9]). All datasets are split into class-incremental tasks (10 for most, 5 for EuroSAT), and ImageNet-A and Cars-196 use 10 FL rounds for stable convergence [41, 42].

Baselines. We compare Sum with representative methods from FL, CL, model merging, and FCIL. From FL, we include CCVR [30] and FedProto [48], extended to continual learning using ACE [4]. From CL, we evaluate EWC [18], LwF [26], L2P [51], and CODA-Prompt [47], adapted to the federated setting via FedAvg [35]. From model merging, we include FisherAvg [33] and RegMean [16]. Finally, we compare with FCIL-specific methods TARGET [61], PILoRA [8], FOT [3], and LoRM [41]. For fairness, we exclude approaches requiring surrogate or centralized server data [14, 31, 59].

Results. As shown in Tab. 2, Sum achieves the highest FAA across all task settings, outperforming all baselines by a large margin. The improvement is consistent across all three dataset categories (*i.e.* in-domain, fine-grained, and out-of-domain), demonstrating the robustness of spatio-temporal refinement of Sum.

Table 3: Evaluation on language-domain datasets in FAA (↑).

	20-NewsGroups			CLINC-150		
Joint	**93.37**			**95.53**		
Distrib. β	1.0	0.5	0.2	1.0	0.5	0.2
RegMean	52.38	50.96	58.31	74.16	74.80	70.78
CCVR	72.84	77.48	69.72	39.18	38.13	30.36
CODA-P	68.71	57.66	56.35	69.02	59.40	38.67
LoRM	88.58	84.73	88.10	84.78	82.58	77.67
SUM	**89.42**	**85.53**	**89.52**	**96.24**	**95.93**	**94.11**

Table 4: Backbone-wise robustness evaluation of SUM.

Backbone	Method	Distrib. β		
		0.5	0.1	0.05
LoRA	Joint		92.93	
	SUM	**96.40**	**95.50**	**96.35**
ConvNeXt	Joint		91.46	
	SUM	**96.19**	**93.75**	**93.93**

Table 5: Ablation study on the components of SUM on CIFAR-100 ($\beta = 0.05$)

	Ablation Setting	FAA (↑)	Wall Time
Spatial SUM	w/o Z-score Trimming	85.87 (↓10.45)	1.00×
	w/o Spatial Surgery	78.93 (↓17.39)	0.77×
Temporal SUM	w/o Temporal Surgery	87.67 (↓8.65)	0.90×
Inference-ready Module	w/o Sparsification	90.23 (↓6.09)	0.92×
	w/o Elect-Sign	94.58 (↓1.74)	0.93×
	w/o Mask Modular	91.24 (↓5.08)	1.00×
SUM	–	96.32	1.00×

Notably, on several datasets such as CIFAR-100, ImageNet-R, and EuroSAT, SUM even surpasses *Joint*. Sec. 5.6 analyzes this behavior through optimization sharpness and representation separation.

5.3 Language Domain

Datasets. To verify that SUM generalizes beyond vision, we further evaluate it on two text classification datasets: 20-Newsgroups [21] for in-domain performance and CLINC-150 (Out-of-Scope) [22] for robustness under domain shift.

Baselines. We adopt four representative baselines covering the paradigms of FL, CL, model merging, and FCIL: CCVR, CODA-Prompt, RegMean, and LoRM.

Results. Tab. 3 presents the results of SUM on the language domain. SUM retains its advantage beyond vision, achieving the highest FAA across all distribution levels. It consistently outperforms all baselines and, in many cases, even surpasses *Joint* under severe distribution shifts. These results show that SUM generalizes seamlessly from vision to language, maintaining stable optimization and consistent representations across heterogeneous domains.

5.4 Architectural Robustness

We assess the architectural robustness of SUM on CIFAR-100 under two representative settings: (1) a parameter-efficient backbone (LoRA [12]) commonly used in FCIL, and (2) a CNN backbone (ConvNeXt [28]) to test generality beyond ViTs. As shown in Tab. 4, SUM maintains strong performance across both

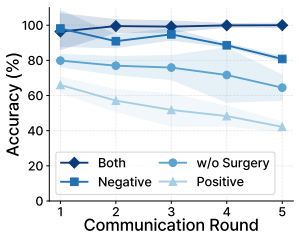


Fig. 3: Ablation study on the target interaction for SUM.

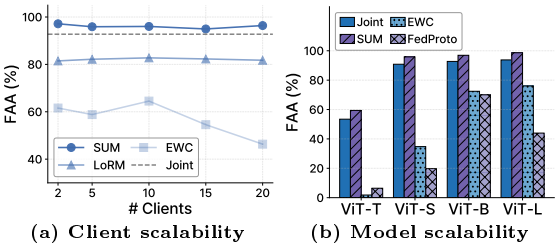


Fig. 4: Scaling behavior of SUM. (a) Number of clients. (b) ViT backbone sizes.

architectures and varying levels of client heterogeneity. Notably, the LoRA-based SUM even surpasses the full-finetuning (Tab. 2), consistent with findings that parameter-efficient modules preserve pretrained representations better than full fine-tuning while reducing communication cost [8, 41]. Overall, these results indicate that SUM is architecture-agnostic and applicable across diverse models.

5.5 Ablation Study

SUM Components. Tab. 5 evaluates the contribution of each component of SUM by removing them from the spatial and temporal axes as well as the inference-ready module. The results show that spatial and temporal SUM form the core of the framework, as removing either leads to a large performance drop. The inference module components are also non-negligible: removing sparsification, Elect-Sign, or task-specific activation consistently degrades FAA, indicating that compact residual selection and task-wise modulation are required to preserve task-specific behavior after temporal surgery. Together, they enable the consistent gains achieved by SUM.

Surgery Target. SUM handles both positive and negative cosine similarities of adaptation vectors to mitigate redundant or destructive updates. A potential concern is that modifying positively aligned vectors may remove beneficial shared information. To assess this, we compare three projection targets: positive-only, negative-only, and bidirectional (both) cosine interactions. As shown in Fig. 3 (CIFAR-100, $\beta = 0.05$), the positive-only variant even underperforms the w/o-SUM baseline, suggesting that naively altering positive directions disrupts the consistent shared representation. In contrast, SUM operates on both positive and negative components while improving performance, preserving essential semantics and resolving both over-alignment and opposition.

5.6 In-depth Analysis

Scaling. We evaluate the client and model scalability of SUM on CIFAR-100 ($\beta = 0.05$) (Fig. 4a and Fig. 4b). For client scaling, we compare against EWC [18],

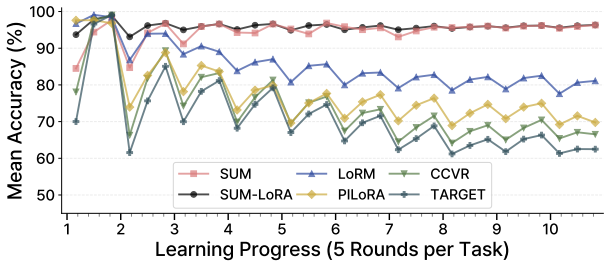
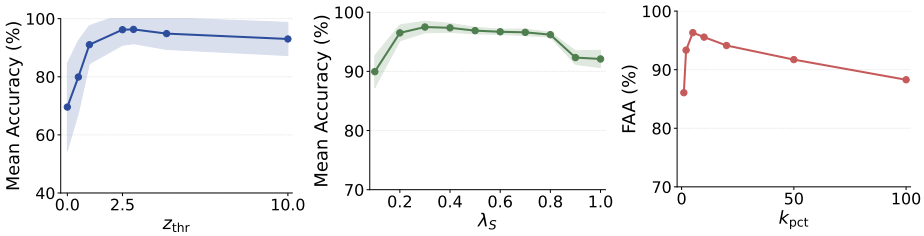


Fig. 5: Learning curves of SUM. A shallower V-shape and stable performance indicate fast adaptation and reduced forgetting.



(a) Z-score threshold **(b)** Scaling coefficient **(c)** Top-k sparsification

Fig. 6: Hyperparameter sensitivity analysis of SUM.

a representative baseline sensitive to heterogeneity, and LoRM [41], a SOTA method. As the number of clients increases, EWC quickly degrades and LoRM plateaus below *Joint*, while SUM maintains and even surpasses *Joint* performance. For model scaling, we include FedProto [48], the strongest full-finetuning approach (Tab. 2). As ViT size grows from Tiny to Large, FedProto and EWC show limited or negative gains, whereas SUM scales smoothly and consistently outperforms *Joint* baseline across all model sizes.

Learning Dynamics To analyze spatio-temporal learning dynamics, we compare baselines with SUM in Fig. 5 by sampling mean accuracies at the beginning, middle, and end of each task. Across baselines, two consistent patterns emerge: (1) a *V-shaped learning curve* during federated rounds and (2) a gradual performance decline as tasks progress. The *V-shape* reflects slower convergence in early rounds, while the decline indicates catastrophic forgetting. In contrast, SUM shows a much shallower V-shape and no noticeable drop across tasks, indicating faster adaptation and better retention.

Hyperparameter Sensitivity We evaluate the robustness of SUM to three hyperparameters: the Z-score threshold z_{thr}, the scaling coefficient λ_S, and the sparsification ratio k_{pct}. As shown in Fig. 6, SUM remains stable across broad ranges, confirming its insensitivity to hyperparameter settings. Accuracy peaks near $z_{\text{thr}} = 3.0$, where large outliers are trimmed without removing informative parameters. λ_S performs consistently well between 0.2 and 0.8, with 0.4 as

Table 6: Computational overhead comparison across FCIL methods. D denotes the model parameter dimension, T the number of tasks, r the LoRA rank, D_{attn} the attention parameter dimension, P the prototype size, G the Gram statistics size, B the orthogonal basis size, and S the synthetic data size.

Method	Upload	Download	Server Memory	Round Time (s)	Task Final. Time (s)
FedAvg	$\mathcal{O}(D)$	$\mathcal{O}(D)$	$\mathcal{O}(D)$	239.4	9.1
PiLoRA	$\mathcal{O}(TrD_{\text{attn}}+P)$	$\mathcal{O}(TrD_{\text{attn}}+TP)$	$\mathcal{O}(D+TrD_{\text{attn}}+TP)$	269.3	0.6
TARGET	$\mathcal{O}(D)$	$\mathcal{O}(D)$	$\mathcal{O}(D+TS)$	272.6	771.3
FOT	$\mathcal{O}(D)$	$\mathcal{O}(D)$	$\mathcal{O}(D+B)$	392.8	21.1
LoRM	$\mathcal{O}(D+G)$	$\mathcal{O}(D+G)$	$\mathcal{O}(D+G)$	689.7	24.5
SUM	$\mathcal{O}(D)$	$\mathcal{O}(D)$	$\mathcal{O}(TD)$	324.2	37.7

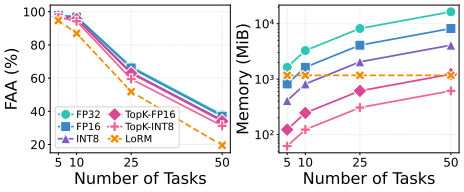


Fig. 7: Memory footprint and mitigation for server-side task scalability of SUM.

Table 7: Robustness to unreliable clients on CIFAR-100 ($\beta = 0.05$).

Method	Clean	Avg. FAA ↑
LoRM	82.8	67.7 ± 8.7 (↓18.2%)
SUM	**96.3**	**80.5 ± 9.6** (↓16.4%)

a balanced default. Finally, moderate sparsification (up to 5%) improves accuracy by suppressing residual interference and highlighting essential parameters, emphasizing the value of compact, interference-free representations in FCIL.

vs. Joint Training *Joint* does not explicitly regulate directional interactions among gradients during centralized optimization. Prior work in MTL and CL shows that such interactions can lead to sharper minima and less separated representations [27,38,52], implying that *Joint* is not necessarily an oracle. In contrast, SUM explicitly removes redundant and destructive components from adaptation vectors before aggregation. To examine this behavior, we analyze the local loss landscape and representation geometry on CIFAR-100 ($\beta = 0.05$) test set, measuring sharpness as the loss increase under ℓ_2 weight perturbations ($\rho = 0.2$) and class separation via inter-class cosine distances between penultimate-layer representations. As shown in Appendix D.7, SUM converges to flatter minima (sharpness $7.7\times10^{-4}\pm1.2\times10^{-4}$ vs. $3.18\times10^{-3}\pm4.5\times10^{-4}$) and larger margins ($0.482\pm0.021$ vs. 0.362 ± 0.018) than *Joint*. These results suggest that resolving directional conflicts before aggregation leads to more stable optimization and improved class separation, explaining why SUM can outperform *Joint*.

5.7 Efficiency and Practical Considerations

We analyze the computational overhead of SUM compared to representative FCIL baselines. Tab. 6 summarizes communication complexity, server memory requirements, and measured runtime on a single H200 GPU, including round time and task finalization time. Although SUM operates on adaptation vectors,

it introduces only lightweight server-side computation while keeping communication costs identical to standard FL updates.

Memory Footprint and Mitigation. In the worst case, maintaining past refined task vectors on the server scales as $O(TD)$, where T is the number of tasks and D is the model parameter dimension. Since this cost is entirely server-side, it does not introduce additional client memory, computation, or communication, but it can become non-negligible as the number of tasks grows. We therefore evaluate simple task-vector compression strategies: low-precision storage using FP16 or INT8, and top-k sparsification that keeps only the largest coordinates of each task vector. As shown in Fig. 7, these compression schemes substantially reduce temporal storage up to 50 tasks while preserving the qualitative FAA trend. This suggests that the $O(TD)$ storage cost is manageable in practice through lightweight server-side compression.

Unreliable Client Robustness. To evaluate robustness under unreliable participants, we simulate malicious clients by corrupting local training labels. Specifically, 1–4 out of 10 clients are trained with shifted ground-truth labels to produce misleading updates. As shown in Tab. 7, Sum maintains substantially higher accuracy and exhibits smaller performance degradation than LoRM [41]. This behavior arises because Sum refines adaptation vectors at aggregation time, suppressing incompatible directions introduced by corrupted client updates.

6 Conclusion

We introduced Surgery & Merge (Sum), a unified framework for Federated Class Incremental Learning (FCIL) that models federated and continual updates as spatio-temporal interactions among adaptation vectors. By performing projection-based surgery at aggregation time, Sum removes redundant and conflicting update components, enabling stable integration of heterogeneous client and task updates. Extensive experiments show that Sum consistently improves performance across FCIL benchmarks while maintaining lightweight communication and computation overhead. Importantly, this improvement is achieved without modifying client-side training, adding replay buffers, or requiring additional client communication, making Sum compatible with standard federated deployment protocols. These results highlight the importance of modeling directional interactions among adaptation vectors for resolving spatio-temporal interference in FCIL.

7 Acknowledgements

This work was supported in part by the National Research Foundation of Korea (NRF) grants (RS-2025-00560762, RS-2024-00414981), the Institute of Information & Communications Technology Planning & Evaluation (IITP) grants (RS-2025-25442338, RS-2025-25455059, RS-2021-II211343), and Samsung Electronics Co., Ltd. (IO251215-14535-01). J. Do is affiliated with ASRI, SNU.

References

1. Babakniya, S., Fabian, Z., He, C., Soltanolkotabi, M., Avestimehr, S.: A data-free approach to mitigate catastrophic forgetting in federated class incremental learning for vision tasks. Proc. of Neural Information Processing Systems (NeurIPS) **36**, 66408–66425 (2023)
2. Bagwe, G., Yuan, X., Pan, M., Zhang, L.: Fed-cprompt: Contrastive prompt for rehearsal-free federated continual learning. arXiv preprint arXiv:2307.04869 (2023)
3. Bakman, Y.F., Yaldiz, D.N., Ezzeldin, Y.H., Avestimehr, S.: Federated orthogonal training: Mitigating global catastrophic forgetting in continual federated learning. arXiv preprint arXiv:2309.01289 (2023)
4. Caccia, L., Aljundi, R., Asadi, N., Tuytelaars, T., Pineau, J., Belilovsky, E.: New insights on reducing abrupt representation change in online continual learning. arXiv preprint arXiv:2104.05025 (2021)
5. Chen, S., Zhou, T., Long, G., Jiang, J., Zhang, C.: Fedmerge: Federated personalization via model merging. arXiv preprint arXiv:2504.06768 (2025)
6. Dong, J., Wang, L., Fang, Z., Sun, G., Xu, S., Wang, X., Zhu, Q.: Federated class-incremental learning. In: Proc. of Computer Vision and Pattern Recognition (CVPR). pp. 10164–10173 (2022)
7. Dosovitskiy, A.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
8. Guo, H., Zhu, F., Liu, W., Zhang, X.Y., Liu, C.L.: Pilora: Prototype guided incremental lora for federated class-incremental learning. In: Proc. of European Conf. on Computer Vision (ECCV). pp. 141–159. Springer (2024)
9. Helber, P., Bischke, B., Dengel, A., Borth, D.: Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing **12**(7), 2217–2226 (2019)
10. Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T., Parajuli, S., Guo, M., et al.: The many faces of robustness: A critical analysis of out-of-distribution generalization. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 8340–8349 (2021)
11. Hendrycks, D., Zhao, K., Basart, S., Steinhardt, J., Song, D.: Natural adversarial examples. In: Proc. of Computer Vision and Pattern Recognition (CVPR). pp. 15262–15271 (2021)
12. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al.: Lora: Low-rank adaptation of large language models. Proc. of Int'l Conf. on Learning Representations (ICLR) **1**(2), 3 (2022)
13. Huang, C., Ye, P., Chen, T., He, T., Yue, X., Ouyang, W.: Emr-merging: Tuning-free high-performance model merging. Proc. of Neural Information Processing Systems (NeurIPS) (2024)
14. Huang, W., Ye, M., Du, B.: Learn from others and be yourself in heterogeneous federated learning. In: Proc. of Computer Vision and Pattern Recognition (CVPR). pp. 10143–10153 (2022)
15. Ilharco, G., Ribeiro, M.T., Wortsman, M., Gururangan, S., Schmidt, L., Hajishirzi, H., Farhadi, A.: Editing models with task arithmetic. Proc. of Int'l Conf. on Learning Representations (ICLR) (2023)
16. Jin, X., Ren, X., Preotiuc-Pietro, D., Cheng, P.: Dataless knowledge fusion by merging weights of language models. Proc. of Int'l Conf. on Learning Representations (ICLR) (2023)

17. Karimireddy, S.P., Kale, S., Mohri, M., Reddi, S., Stich, S., Suresh, A.T.: Scaffold: Stochastic controlled averaging for federated learning. In: International conference on machine learning. pp. 5132–5143. PMLR (2020)
18. Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A.A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al.: Overcoming catastrophic forgetting in neural networks. Proceedings of the national academy of sciences **114**(13), 3521–3526 (2017)
19. Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3d object representations for fine-grained categorization. In: Proceedings of the IEEE international conference on computer vision workshops. pp. 554–561 (2013)
20. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images.(2009) (2009)
21. Lang, K.: Newsweeder: Learning to filter netnews. In: Machine learning proceedings 1995, pp. 331–339. Elsevier (1995)
22. Larson, S., Mahendran, A., Peper, J.J., Clarke, C., Lee, A., Hill, P., Kummerfeld, J.K., Leach, K., Laurenzano, M.A., Tang, L., et al.: An evaluation dataset for intent classification and out-of-scope prediction. arXiv preprint arXiv:1909.02027 (2019)
23. Li, Y., Wang, H., Qi, Y., Liu, W., Li, R.: Re-fed+: A better replay strategy for federated incremental learning. IEEE Transactions on Pattern Analysis and Machine Intelligence (2025)
24. Li, Y., Wang, X., Xu, W., Wang, H., Qi, Y., Dong, J., Li, R.: Feature distillation is the better choice for model-heterogeneous federated learning. Advances in Neural Information Processing Systems **38**, 104726–104744 (2026)
25. Li, Y., Wang, Y., Wang, H., Qi, Y., Xiao, T., Li, R.: Rehearsal-free continual federated learning with synergistic synaptic intelligence. arXiv preprint arXiv:2412.13779 (2024)
26. Li, Z., Hoiem, D.: Learning without forgetting. IEEE transactions on pattern analysis and machine intelligence **40**(12), 2935–2947 (2017)
27. Liu, B., Liu, X., Jin, X., Stone, P., Liu, Q.: Conflict-averse gradient descent for multi-task learning. Proc. of Neural Information Processing Systems (NeurIPS) **34**, 18878–18890 (2021)
28. Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S.: A convnet for the 2020s. In: Proc. of Computer Vision and Pattern Recognition (CVPR). pp. 11976–11986 (2022)
29. Lopez-Paz, D., Ranzato, M.: Gradient episodic memory for continual learning. Proc. of Neural Information Processing Systems (NeurIPS) **30** (2017)
30. Luo, M., Chen, F., Hu, D., Zhang, Y., Liang, J., Feng, J.: No fear of heterogeneity: Classifier calibration for federated learning with non-iid data. Proc. of Neural Information Processing Systems (NeurIPS) **34**, 5972–5984 (2021)
31. Ma, Y., Xie, Z., Wang, J., Chen, K., Shou, L.: Continual federated learning based on knowledge distillation. In: Int'l Joint Conf. on Artificial Intelligence (IJCAI). pp. 2182–2188 (2022)
32. Marczak, D., Twardowski, B., Trzciński, T., Cygert, S.: Magmax: Leveraging model merging for seamless continual learning. In: Proc. of European Conf. on Computer Vision (ECCV). pp. 379–395. Springer (2024)
33. Matena, M.S., Raffel, C.A.: Merging models with fisher-weighted averaging. Proc. of Neural Information Processing Systems (NeurIPS) **35**, 17703–17716 (2022)
34. McCloskey, M., Cohen, N.J.: Catastrophic interference in connectionist networks: The sequential learning problem. In: Psychology of learning and motivation, vol. 24, pp. 109–165. Elsevier (1989)

35. McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Artificial intelligence and statistics. pp. 1273–1282. PMLR (2017)
36. Mhanna, E., Assaad, M.: Countering the communication bottleneck in federated learning: A highly efficient zero-order optimization technique. Journal of Machine Learning Research **25**(418), 1–53 (2024)
37. Nguyen, D.M., Nguyen, L.T., Pham, Q.V.: Spatio-temporal gradient matching for federated continual learning. In: ICML 2025 Workshop on Collaborative and Federated Agentic Workflows (2025)
38. Phan, H., Tran, L., Tran, Q., Tran, N., Truong, T., Lei, Q., Ho, N., Phung, D., Le, T.: Beyond losses reweighting: Empowering multi-task learning via the generalization perspective. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2440–2450 (2025)
39. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of machine learning research **21**(140), 1–67 (2020)
40. Saha, G., Garg, I., Roy, K.: Gradient projection memory for continual learning. arXiv preprint arXiv:2103.09762 (2021)
41. Salami, R., Buzzega, P., Mosconi, M., Bonato, J., Sabetta, L., Calderara, S.: Closed-form merging of parameter-efficient modules for federated continual learning. Proc. of Int'l Conf. on Learning Representations (ICLR) (2024)
42. Salami, R., Buzzega, P., Mosconi, M., Verasani, M., Calderara, S.: Federated class-incremental learning with hierarchical generative prototypes. arXiv preprint arXiv:2406.02447 (2024)
43. Sener, O., Koltun, V.: Multi-task learning as multi-objective optimization. Proc. of Neural Information Processing Systems (NeurIPS) **31** (2018)
44. Senushkin, D., Patakin, N., Kuznetsov, A., Konushin, A.: Independent component alignment for multi-task learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20083–20093 (2023)
45. Shi, C., Zhao, H., Zhang, B., Zhou, M., Guo, D., Chang, Y.: Fedawa: Adaptive optimization of aggregation weights in federated learning using client vectors. In: Proc. of Computer Vision and Pattern Recognition (CVPR). pp. 30651–30660 (2025)
46. Shin, H., Lee, J.K., Kim, J., Kim, J.: Continual learning with deep generative replay. Proc. of Neural Information Processing Systems (NeurIPS) **30** (2017)
47. Smith, J.S., Karlinsky, L., Gutta, V., Cascante-Bonilla, P., Kim, D., Arbelle, A., Panda, R., Feris, R., Kira, Z.: Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. In: Proc. of Computer Vision and Pattern Recognition (CVPR). pp. 11909–11919 (2023)
48. Tan, Y., Long, G., Liu, L., Zhou, T., Lu, Q., Jiang, J., Zhang, C.: Fedproto: Federated prototype learning across heterogeneous clients. In: Proceedings of the AAAI conference on artificial intelligence. vol. 36, pp. 8432–8440 (2022)
49. Tziolas, G., Siniosoglou, I., Sarigiannidis, P., Argyriou, V.: A contemporary survey of federated continual learning (2025). `https://doi.org/10.5281/zenodo.14768027`, `https://doi.org/10.5281/zenodo.14768027`
50. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The caltech-ucsd birds-200-2011 dataset. Tech. Rep. CNS-TR-2011-001, California Institute of Technology (2011)
51. Wang, Z., Zhang, Z., Lee, C.Y., Zhang, H., Sun, R., Ren, X., Su, G., Perot, V., Dy, J., Pfister, T.: Learning to prompt for continual learning. In: Proc. of Computer Vision and Pattern Recognition (CVPR). pp. 139–149 (2022)

52. Worsham, J., Kalita, J.: Multi-task learning for natural language processing in the 2020s: Where are we going? Pattern Recognition Letters **136**, 120–126 (2020)
53. Yadav, P., Tam, D., Choshen, L., Raffel, C.A., Bansal, M.: Ties-merging: Resolving interference when merging models. Proc. of Neural Information Processing Systems (NeurIPS) **36** (2023)
54. Yang, E., Wang, Z., Shen, L., Liu, S., Guo, G., Wang, X., Tao, D.: Adamerging: Adaptive model merging for multi-task learning. Proc. of Int'l Conf. on Learning Representations (ICLR) (2024)
55. Yang, X., Yu, H., Gao, X., Wang, H., Zhang, J., Li, T.: Federated continual learning via knowledge fusion: A survey. IEEE Transactions on Knowledge and Data Engineering **36**(8), 3832–3850 (2024)
56. Yang, Z., Zhang, Y., Zheng, Y., Tian, X., Peng, H., Liu, T., Han, B.: Fedfed: Feature distillation against data heterogeneity in federated learning. Proc. of Neural Information Processing Systems (NeurIPS) **36**, 60397–60428 (2023)
57. Yoon, J., Jeong, W., Lee, G., Yang, E., Hwang, S.J.: Federated continual learning with weighted inter-client transfer. In: Proc. of Int'l Conf. on Machine Learning (ICML). pp. 12073–12086. PMLR (2021)
58. Yu, H., Yang, X., Gao, X., Feng, Y., Wang, H., Kang, Y., Li, T.: Overcoming spatial-temporal catastrophic forgetting for federated class-incremental learning. In: Proceedings of the 32nd ACM International Conference on Multimedia. pp. 5280–5288 (2024)
59. Yu, H., Yang, X., Zhang, L., Gu, H., Li, T., Fan, L., Yang, Q.: Handling spatial-temporal data heterogeneity for federated continual learning via tail anchor. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 4874–4883 (2025)
60. Yu, T., Kumar, S., Gupta, A., Levine, S., Hausman, K., Finn, C.: Gradient surgery for multi-task learning. Proc. of Neural Information Processing Systems (NeurIPS) **33**, 5824–5836 (2020)
61. Zhang, J., Chen, C., Zhuang, W., Lyu, L.: Target: Federated class-continual learning via exemplar-free distillation. In: Proc. of Int'l Conf. on Computer Vision (ICCV). pp. 4782–4793 (2023)