IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Communication-Efficient Module-Wise Federated Learning for Grasp Pose Detection in Cluttered Environments

Woonsang Kang, Joohyung Lee, Seungjun Kim, Jungchan Cho, and Yoonseon Oh

***Abstract*—Grasp pose detection (GPD) is a fundamental capability for robotic autonomy, but its reliance on large, diverse datasets creates significant data privacy and centralization challenges. Federated Learning (FL) offers a privacy-preserving solution, but its application to GPD is hindered by the substantial communication overhead of large models, a key issue for resource-constrained robots. To address this, we propose a novel module-wise FL framework that begins by analyzing the learning dynamics of the GPD model's functional components. This analysis identifies slower-converging modules, to which our framework then allocates additional communication effort. This is realized through a two-phase process: a standard full-model training phase is followed by a communication-efficient phase where only an adaptively identified subset of slower-converging modules is trained and their partial updates are aggregated. Extensive experiments on the GraspNet-1B dataset demonstrate that our method outperforms standard FedAvg and other baselines, achieving higher accuracy for a given communication budget. Furthermore, real-world experiments on a physical robot validate our approach, showing a superior grasp success rate compared to baseline methods in cluttered scenes. Our work presents a communication-efficient framework for training robust, generalized GPD models in a decentralized manner, effectively improving the trade-off between communication cost and model performance.**

***Index Terms*—Deep learning in grasping and manipulation, deep learning methods.**

I. Introduction

Grasp pose detection (GPD) is a fundamental capability for autonomous robotic systems, with applications ranging from industrial automation to assistive robotics. In recent years, learning-based approaches have become the dominant paradigm for this task [1], [2]. This success, however, is critically dependent on large-scale, diverse datasets capable of fostering generalization across various objects, scenes, and environmental conditions [2], [3], [4], [5].

However, acquiring such data presents substantial challenges. The collection process is not only laborious but, more critically, raises significant privacy concerns when data originates from private settings like homes or proprietary industrial environments [2], [6], [7]. The traditional requirement of centralizing sensitive sensor information thus becomes a major obstacle, restricting the ability to pool datasets from multiple sources. This limitation ultimately hinders the training of more robust, generalized models and creates a *data island* problem that impedes the development of universally applicable GPD.

Federated Learning (FL) has emerged as a compelling paradigm for addressing privacy and data governance challenges, enabling multiple clients to collaboratively train a model without exchanging their raw data [7]. This decentralized approach is highly pertinent to robotics, allowing for the use of diverse, distributed datasets for applications like autonomous navigation [8] and manipulator control [9] while preserving data privacy. However, a significant bottleneck in FL is the *communication* overhead, particularly for the large-parameter models used in GPD [7]. The frequent transmission of model updates consumes substantial bandwidth and energy, a critical issue for resource-constrained robots. Although prior works like [10] have aimed to address data heterogeneity by clustering robots (or clients), their approach can intensify this resource burden.

To address the resource burden of FL for GPD, we observe that contemporary models in this domain feature *multi-modular* architectures [2], [11], [12], [13]. To analyze these architectures in an FL setting, we measured the cosine similarity of model updates on a module-by-module basis [14], [15]. Our analysis reveals that these modules exhibit heterogeneous learning dynamics, allowing us to distinguish between faster-converging modules and the slower-converging ones that act as training bottlenecks. This finding motivates our proposal of a novel, two-phase module-wise FL algorithm, as illustrated in Fig. 1. The first phase consists of standard full-model training and full aggregation. The second phase, in contrast, employs a parameter-efficient approach by strategically concentrating resources: only the subset of modules identified as slower-converging undergoes partial-model training, and subsequently, only their updated parameters are aggregated. This approach maximizes the

Received 7 July 2025; accepted 4 November 2025. Date of publication 5 December 2025; date of current version 15 December 2025. This article was recommended for publication by Associate Editor C. Choi and Editor J. B. Sol upon evaluation of the reviewers' comments. This work was supported by the National Research Foundation of Korea (NRF) funded by the Korea Government (MSIT) under Grant RS-2024-00409492 and Grant RS-2025-22442984, Beyond-G Global Innovation Center. *(Corresponding authors: Yoonseon Oh; Jungchan Cho.)*

Woonsang Kang and Yoonseon Oh are with the Department of Electronic Engineering, Hanyang University, Seoul 04763, South Korea (e-mail: monni1729@hanyang.ac.kr; yoh21@hanyang.ac.kr).

Joohyung Lee and Jungchan Cho are with the Department of Computing, Gachon University, Seongnam 13120, South Korea (e-mail: j17.lee@gachon.ac.kr; thinkai@gachon.ac.kr).

Seungjun Kim is with the Department of Artificial Intelligence, Hanyang University, Seoul 04763, South Korea (e-mail: rlatmdwnseo@gachon.ac.kr).

Digital Object Identifier 10.1109/LRA.2025.3641101

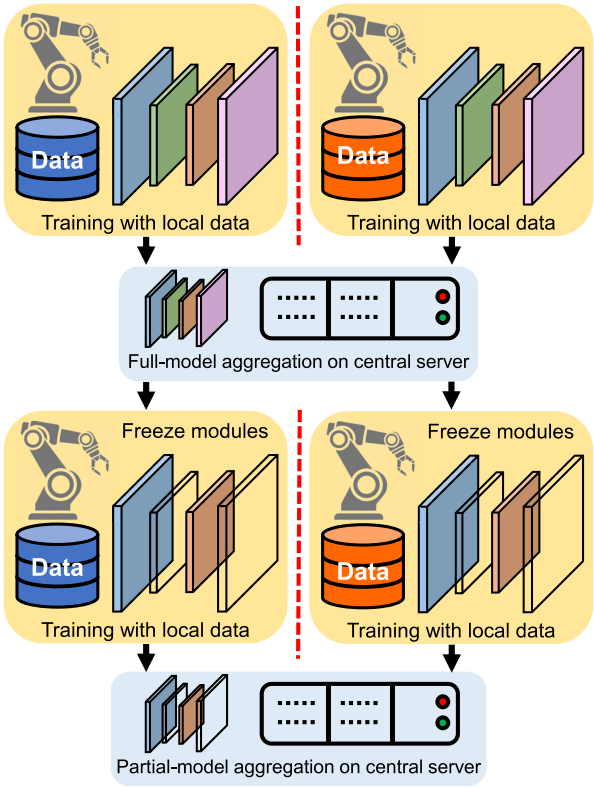

Fig. 1. An overview of our proposed two-phase module-wise FL algorithm. Phase 1 consists of standard full-model training and aggregation. In contrast, Phase 2 enhances communication efficiency with a partial-update strategy: based on our module-wise analysis, the faster-converging modules (visualized as transparent plates) are frozen. Only the remaining, slower-converging modules are trained, and their partial updates are aggregated by the server. The red dotted lines represent the privacy boundary, ensuring no raw data is shared. The aggregation step involves sharing only model weights.

performance return on a given communication budget, enhancing the feasibility of FL for complex robotics tasks like GPD. Our primary contributions are as follows:

- We identify communication overhead as a critical bottleneck for FL in GPD. We introduce a module-wise similarity analysis that reveals heterogeneous learning dynamics and, crucially, enables the identification of specific slower-converging modules that act as training bottlenecks.
- Motivated by our analysis, we propose a novel module-wise FL algorithm featuring a two-phase process. This algorithm allocates additional training and communication resources exclusively to the slower-converging modules adaptively, enhancing communication efficiency and maximizing performance for a given budget.
- Through extensive experiments on the GraspNet-1B dataset [2] and in the real world, we demonstrate the superiority of our approach over existing baselines and validate its practical effectiveness.

II. Related Work

A. Data-Driven Grasp Pose Detection

GPD aims to identify a stable 6-DoF (Degree of freedom) gripper pose from sensory inputs like RGB-D images or 3D point clouds. Historically, this problem was often addressed through model-based approaches [1], which relied on having precise 3D models of the objects. However, these methods faced significant limitations, such as difficulty handling novel objects and sensitivity to sensor noise. The advent of deep learning has since spurred a paradigm shift towards data-driven grasping methodologies that learn directly from data. The field's progress is driven by advanced datasets [2], [3], [5] and novel model architectures [12], [16]. Early data-driven works focused on representing grasps as 2D rectangles in the image plane [4], [17]. However, these 2D approaches could not represent the full spatial orientation required for manipulation in *cluttered scenes*. This limitation spurred a paradigm shift towards methods that detect full 6-DoF poses directly from 3D point clouds. These modern approaches have progressed from sampling-based techniques that evaluate numerous candidates [2] to more efficient, single-stage frameworks that directly predict grasp poses [12], [16], [18], [19]. This advancement is strongly supported by the development of large-scale datasets, which are crucial for training and evaluation. These range from early synthetic datasets with millions of grasp examples [3] to recent real-world benchmarks focusing on cluttered environments and object diversity [2], [5]. However, the very success built upon these diverse, real-world datasets underscores a critical challenge: the logistical and privacy-related difficulties in centralizing such large-scale, sensitive data.

B. Federated Learning for Robotic Grasping

FL has emerged as a machine learning paradigm that enables multiple decentralized clients to collaboratively train a model without sharing their raw local datasets [7]. In this framework, exemplified by the canonical FedAvg algorithm, clients locally train a shared model, and a central server aggregates only their resulting updates to produce an improved global model. The inherent privacy preservation of this framework makes it highly suitable for robotics, where applications often involve sensitive data from private residences or proprietary industrial processes. Reflecting its growing importance, FL is now being applied to a range of robotic tasks to scaling robot learning. It includes multi-robot trajectory prediction [20], visual robotic navigation [8], and robotic manipulation [9], [21].

However, applying FL to GPD is still in its early stages. While a prior work by [10] did explore this direction by applying a clustering-based FL approach to 2D grasping, it suffers from two critical limitations. First, from a task perspective, it is confined to simplified 2D rectangular grasps, which are insufficient for dexterous manipulation in cluttered 3D scenes. Second, from a methodological standpoint, its clustering process not only incurs significant communication overhead but also risks harming generalization by permanently excluding clients with unique data distributions. Therefore, applying FL to the more practical and challenging GPD problem, while simultaneously addressing the communication-efficiency of the FL algorithm itself, remains a significant open challenge. Our work aims to fill this critical gap by proposing a novel, communication-efficient FL strategy specifically designed for multi-modular GPD models.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

C. Communication-Efficient Federated Learning

A primary bottleneck in FL is the substantial communication overhead required for the iterative transmission of large-scale models between clients and a central server. This challenge is exacerbated as model complexity and the number of participating clients increase, creating a critical trade-off between model performance and communication cost. Consequently, a significant body of research has focused on developing communication-efficient FL strategies. Early approaches centered on reducing communication frequency by allowing for more local computation on client devices before aggregation [7]. Another major line of work involves compressing the weight updates themselves through techniques like quantization [22], [23], which reduces the precision of weight updates, and sparsification [24], which transmits only a subset of the most significant updates. More recently, sophisticated adaptive methods have been introduced. An example is FedLAMA [25], which employs a layer-wise adaptive aggregation strategy. Motivated by the observation that different layers converge at different rates, this approach adjusts the aggregation frequency of each layer based on its contribution to model discrepancy. However, these methods fundamentally assume a monolithic model architecture, and consequently, their approach to efficiency is restricted to the layer level. In contrast, our work is motivated by the functionally distinct, modular structures inherent in advanced robotics models. Instead of viewing the model as a simple stack of layers, we leverage the heterogeneous learning dynamics of its functional components. This allows us to propose a novel communication-efficient strategy tailored to multi-component systems, enabling a more granular and effective allocation of communication resources.

III. Preliminaries and Framework Overview

This section introduces the background for our work. We start by explaining the standard FL framework and the 6-DoF grasp pose representation used for GPD. Next, we highlight a key opportunity for improvement that comes from the modular structure of GPD models. This opportunity is the motivation for the high-level overview of our proposed framework, which is presented at the end of the section.

Federated Learning Framework. FL is a decentralized machine learning paradigm for settings where training data is distributed across n clients (indexed by $c \in \{1, \dots, n\}$) and cannot be centrally aggregated due to privacy or communication constraints. The objective is to collaboratively train a global model with parameters w by minimizing a global loss function $L(w)$, which is the weighted average of each client's local loss $L_c(w; D_c)$ on their local dataset D_c:

$$\min_w L(w) = \sum_{c=1}^{n} k_c L_c(w; D_c)$$

The weight $k_c = |D_c|/|D|$ (where $|D| = \sum_{i=1}^{n} |D_i|$) typically represents the relative size of each client's dataset. The canonical algorithm to solve this objective is FedAvg [7]. The training process operates in communication rounds (indexed by r), where the server first distributes the current global model w^r to all clients. Each client then performs local training to compute an updated model w_c^{r+1} and, to preserve privacy, transmits only the model update $\Delta w_c^{r+1} = w_c^{r+1} - w^r$ back to the server. The server aggregates these updates to form the next global model w^{r+1}:

$$w^{r+1} = w^r + \sum_{c=1}^{n} k_c \Delta w_c^{r+1}$$

While more advanced optimizers exist [26], [27], we use FedAvg [7] as it provides a clear baseline for validating our module-wise strategy, which is orthogonal to the choice of optimizer.

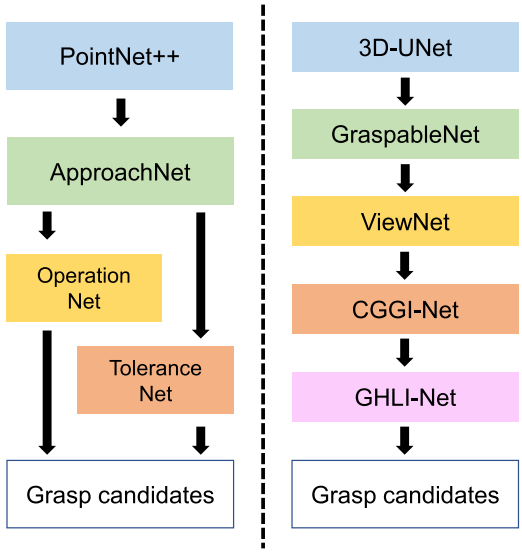

Fig. 2. Examples of multi-module architectures for GPD, from [2] (left) and [13] (right). These diagrams are simplified to show only learnable neural network modules, omitting other operations. The black arrows represent the main forward pass, though additional residual connections may exist in the actual models.

Grasp Pose Representation. Directly learning the 6-DoF grasp pose matrix is challenging for neural networks due to the complex constraints of rotation matrices. Consequently, many state-of-the-art approaches [2], [11], [13] adopt a decoupled representation. Following the methodology in [13], we define a 6-DoF grasp $\mathbf{G}$ as a tuple $[\mathbf{c}, v, a, d, w, s]$. In this formulation, $\mathbf{c} \in \mathbb{R}^3$ is the grasp center point. The integers v, a, and d represent the discretized approach direction, in-plane rotation, and grasp depth, respectively. Finally, $w \in \mathbb{R}$ is the grasp width and $s \in \mathbb{R}$ is the predicted grasp quality score.

An Opportunity in Multi-Modular GPD Models for FL. A key optimization opportunity in FL arises from the inherent structure of modern GPD models. As illustrated in Fig. 2, these models are typically not monolithic but are composed of multiple, functionally distinct neural network modules. Given that each module is specialized for a different sub-task, we posit that they will exhibit heterogeneous learning dynamics when trained under FL. For instance, some modules may converge faster, while others might require additional training.

Overview of Our Proposed Framework. Based on this premise, our work introduces a novel *module-wise FL framework* designed to exploit these dynamics for communication-efficient learning. The core idea, which will be detailed in

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Section IV, is to supplement the standard full-model update with a second, resource-efficient phase focused only on a specific subset of modules. By strategically concentrating training and communication resources on specific components, our framework aims to maximize model performance for a given communication budget.

IV. Federated Learning Framework for Grasp Pose Detection

A. Analysis of Module-Wise Learning Dynamics in FL

Modern deep learning models for GPD often feature multi-modular architectures [2], [11], [12], [13], as illustrated in Fig. 2. This modularity is a notable characteristic of robotics models, stemming from the inherent difficulty of training a neural network to interpret complex 3D spatial information [2]. Consequently, a common architectural approach is to employ a combination of multiple specialized modules. For instance, the model presented in [2] consists of `PointNet++` [28], `ApproachNet`, `OperationNet`, and `ToleranceNet`. Similarly, the architecture in [13] is composed of `3D-UNet`, `GraspableNet`, `ViewNet`, `CGGI-Net`, and `GHLI-Net`. These modules are tasked with specific functions, such as extracting 3D features, determining approaching vectors, and estimating grasp quality. Motivated by this observation, we investigate whether these functionally distinct modules exhibit heterogeneous learning dynamics during FL training.

To quantify these dynamics, we analyze the alignment between local client updates and the aggregated global update for each module using cosine similarity. This metric has been employed in clustered FL research [14], [15] to assess biases in client data. For each module $m \in M$, where M is the set of all modules in the model, in communication round r, we compute the average module-wise similarity S_m^r as:

$$S_m^r = \frac{1}{n} \sum_{c=1}^{n} \frac{\Delta w_{g,m}^r \cdot \Delta w_{c,m}^r}{\|\Delta w_{g,m}^r\| \|\Delta w_{c,m}^r\|}$$

Here, $\Delta w_{c,m}^r$ is the update for module m from client c, and $\Delta w_{g,m}^r$ is the aggregated global update for that module ($\Delta w_{g,m}^r = \sum_{c=1}^{n} k_c \Delta w_{c,m}^r$).

Our analysis, conducted by applying FedAvg to a GPD model with a multi-modular architecture analogous to that in [13], reveals heterogeneous learning dynamics at the module level. As illustrated in Fig. 3 (see Section V for experimental details), these modules exhibit distinct similarity trajectories. For example, the module analogous to `GraspableNet` consistently shows high cosine similarity that diminishes gradually. Conversely, the module analogous to `3D-UNet` shows a lower initial similarity and a steady decline. These similarity values offer insight into the module's convergence state, showing the difference between learning *general* and *personalized* features, a key idea in [29], [30]. Specifically, if a module is far from its stationary point, it is primarily learning general features, leading to high similarity between local and global updates. As a module approaches convergence, however, it begins to learn local-data-specific features, causing the local updates to diverge and thus lowering their similarity to the global update. A low similarity suggests that the module is close to a stationary point, and thus can be considered fast-converging. In contrast, a module that maintains a high similarity value is interpreted as being relatively far from its stationary point, signifying a slow-converging module. This leads to our core design principle: to maximize the performance return on a given communication budget, additional training and aggregation efforts should be directed towards the slower-converging modules, as they are the primary bottlenecks to overall model performance.

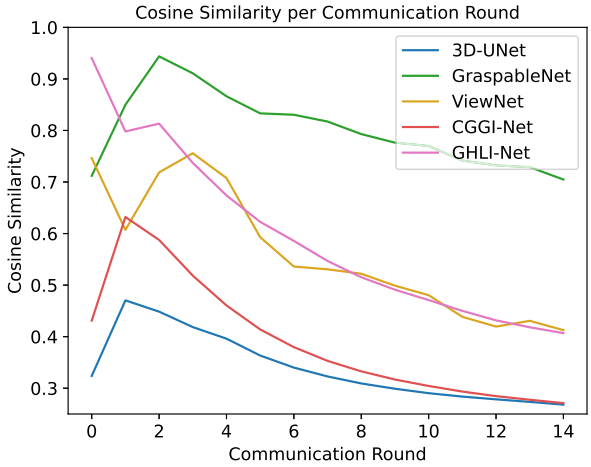

Fig. 3. Cosine similarity between local client updates and the global update for different modules across communication rounds. The plotted values are averaged over three independent experiments, and the line colors are aligned with the corresponding modules in the right-hand architecture of Fig. 2. The plot reveals that distinct modules exhibit heterogeneous learning dynamics; for example, `GraspableNet` shows a higher similarity compared to `3D-UNet`.

B. Our Two-Phase Module-Wise FL Algorithm

Based on our core design principle that additional effort should be focused on slower-converging modules, we introduce a novel **two-phase module-wise FL algorithm**. In this framework, the set of slower-converging modules, $\tilde{M}$, is adaptively identified within each communication round. This intra-round adaptation allows the algorithm to immediately focus on the modules that are the current learning bottlenecks. Our framework, detailed in Algorithm 1, is strategically designed to resolve the fundamental trade-off between full-model updates for robust generalization and partial-model updates for communication efficiency.

Phase 1: Full-Model Training and Aggregation. As detailed in lines 3-4 of Algorithm 1, each round begins with clients training the full model (M) on their local data for τ_1 epochs. The resulting updates, Δw_c, are transmitted to the server. The server then performs a weighted aggregation to compute the global update, Δw_g, and applies it to the current model (w^r) to produce an intermediate model, $w^{r+0.5}$. Crucially, the full-model updates collected in this phase provide a real-time snapshot of the learning dynamics across all modules.

Phase 2: Adaptive Module-Selective Training and Aggregation. Following Phase 1, the server immediately uses the

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Algorithm 1: Module-wise FL (Server).

Input: Initial global model parameters w^0; Total communication rounds R; Number of clients n; Client weights $\{k_c\}_{c=1}^n$; Local epochs τ_1, τ_2; Number of slow-converging modules to select ζ

Output: Final global model parameters w^R

1 Server distributes w^0 to all clients;
2 **for** $r \leftarrow 0$ **to** $R-1$ **do**
3 **for** $c \leftarrow 1$ **to** n **do in parallel**
4 $\Delta w_c \leftarrow \texttt{ClientUpdate}(c, M, \tau_1)$;
5 $\Delta w_g \leftarrow \sum_{c=1}^n k_c \Delta w_c$;
6 $w^{r+0.5} \leftarrow w^r + \Delta w_g$;
7 Calculate S_m^r using updates from Phase 1;
8 $\tilde{M}^r \leftarrow$ Select top-ζ modules with highest S_m^r;
9 Server distributes $w^{r+0.5}$ and $\tilde{M}^r$ to all clients;
10 **for** $c \leftarrow 1$ **to** n **do in parallel**
11 $\Delta w_c[\tilde{M}^r] \leftarrow \texttt{ClientUpdate}(c, \tilde{M}^r, \tau_2)$;
12 $\Delta w_g[\tilde{M}^r] \leftarrow \sum_{c=1}^n k_c \Delta w_c[\tilde{M}^r]$;
13 $w^{r+1} \leftarrow w^{r+0.5}$;
14 $w^{r+1}[\tilde{M}^r] \leftarrow w^{r+0.5}[\tilde{M}^r] + \Delta w_g[\tilde{M}^r]$;
15 Server distributes $w^{r+1}[\tilde{M}^r]$ to all clients;
16 **return** w^R;

collected updates (Δw_c) to calculate the module-wise cosine similarity S_m^r for all modules. It then identifies the top-ζ modules with the highest similarity as the current set of slow-converging modules, $\tilde{M}^r$ (lines 7-8). This newly identified set $\tilde{M}^r$, along with the intermediate model $w^{r+0.5}$, is then distributed to the clients.

In the second phase (lines 10-11), clients apply the `ClientUpdate` procedure exclusively to this just-in-time identified subset $\tilde{M}^r$ for τ_2 epochs. Clients transmit only the resulting partial updates, $\Delta w_c[\tilde{M}^r]$, back to the server. The server aggregates these to get the global partial update $\Delta w_g[\tilde{M}^r]$ and selectively applies it to form the final model for the round, w^{r+1}. For the next round, the server efficiently distributes only the updated partial model, $w^{r+1}[\tilde{M}^r]$, which clients use to reconstruct the full model locally. This intra-round adaptive mechanism ensures that the additional training effort in Phase 2 is always directed at the most relevant modules based on the most current learning dynamics.

In essence, our two-phase algorithm systematically separates the general, full-model update from the efficient, targeted refinement of slow-converging modules. This synergistic approach creates a more communication-efficient path to achieving high-performance GPD in a FL framework.

V. Experiments

A. Experimental Setup

Dataset and Model. Our experiments are conducted on the large-scale GraspNet-1B dataset [2], using data captured by the Kinect sensor. Our implementation builds upon the off-the-shelf codebase of EconomicGrasp [13]. We chose this framework as it achieves state-of-the-art performance among frameworks that leverage *less supervision*, making it an efficient baseline for our study. We adopt the standard architecture settings from the original work, with the feature dimensions of the 3D-UNet set to $[12, 24, 48, 96, 72, 72, 72, 72]$ and a fixed learning rate of 0.001. All other hyperparameters and data processing steps follow the original implementation unless otherwise specified. All experiments were performed on a single NVIDIA RTX 4000 GPU.

Federated Learning Protocol. To simulate a federated environment, we distributed the GraspNet-1B dataset across $n = 20$ clients. In a single experimental run, a total of 100 unique scenes were used, with each client being randomly assigned data from 5 distinct scenes. To ensure the reliability of our results, we repeated all experiments 5 times.

Evaluation Metrics and Cost Quantification. The performance metric is Average Precision (AP), calculated across a range of friction coefficients and averaged over the top-k predictions (we set $k = 10$). We utilize the standard `Seen`, `Unseen`, and `Novel` scene splits to assess generalization, and report an `Overall` score calculated as the average AP across these three splits.

To quantify resource utilization, we define communication cost as the total volume of model parameters transmitted between the server and clients. For our proposed method, this can be expressed as:

$$C_{\text{comm}} = 2nR\left(|w| + |w[\tilde{M}]|\right) + n|w|$$

where $|w|$ is the size of the full model parameters and $|w[\tilde{M}]|$ is the size of the module-wise updates. Computation cost is the cumulative GPU processing time for a single client for the entire training process.

B. Methods for Comparison

Our investigation is based on the EconomicGrasp model, which we treat as a composition of five distinct modules: m_1 (`3D-UNet`), m_2 (`GraspableNet`), m_3 (`ViewNet`), m_4 (`CGGI-Net`), and m_5 (`GHLI-Net`). Our proposal is an adaptive two-phase algorithm that adaptively identifies and focuses on slow-converging modules. To validate its effectiveness, we compare it against several baselines and conduct extensive ablation studies. These studies include methods with pre-identified, static module sets for Phase 2. For all two-phase experiments, we set the local epochs to $\tau_1 = 1$ and $\tau_2 = 1$.

Baselines:

- `iso`: This baseline represents an *isolated client* where each client trains a model using only its local data without any communication or aggregation. This establishes a performance lower bound, representing the outcome without collaboration.
- `t1` and `t2`: These represent standard FedAvg where the full model is trained in each communication round. We test with the number of local epochs set to $\tau = 1$ (`t1`) and $\tau = 2$ (`t2`) to evaluate the impact of additional local computation.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

TABLE I
QUANTITATIVE COMPARISON OF OUR PROPOSED METHOD WITH BASELINES AND ABLATION STUDIES. ALL EVALUATION RESULTS ARE AVERAGED OVER AT LEAST FIVE RUNS. THE BEST EVALUATION RESULTS ARE MARKED IN **BOLD**

Method		Evaluation Set (AP, %)				Communication Cost (GB, ↓)	Computation Cost (sec., ↓)
		Seen (↑)	Unseen (↑)	Novel (↑)	Overall (↑)		
Baseline	iso	44.44 ± 4.50	36.57 ± 2.58	13.46 ± 1.77	31.49 ± 2.89	0	1000++
	t1	62.09 ± 1.15	45.29 ± 0.61	17.92 ± 1.11	41.76 ± 0.55	10.48	2365
	t2	61.72 ± 0.91	46.67 ± 1.13	18.44 ± 0.65	42.28 ± 0.67	7.417	3295
	FedLAMA [25]	62.10 ± 0.56	46.31 ± 1.45	18.59 ± 1.10	42.33 ± 0.90	7.765	3121
Ablation	m1	63.20 ± 1.51	44.21 ± 3.74	17.65 ± 1.75	41.69 ± 2.24	10.41	2718
	m14	62.71 ± 0.83	46.32 ± 0.71	19.08 ± 1.38	42.70 ± 0.92	9.683	2324
	m35	63.68 ± 1.35	48.37 ± 1.90	19.37 ± 1.70	43.80 ± 1.64	7.947	2688
	m45	64.04 ± 1.14	49.40 ± 1.01	19.34 ± 0.417	44.26 ± 0.83	7.714	2463
	m135	61.75 ± 2.50	44.47 ± 1.26	17.27 ± 0.70	41.17 ± 1.46	10.44	2455
	m235	63.91 ± 1.17	49.82 ± 1.35	19.22 ± 1.62	44.32 ± 1.38	7.358	2588
	m245	61.69 ± 2.01	48.19 ± 2.65	18.08 ± 1.58	42.65 ± 1.84	7.717	2488
	m2345	62.78 ± 0.41	49.38 ± 1.56	19.31 ± 1.27	43.82 ± 0.91	7.828	2435
Ours	a3	63.17 ± 0.58	50.11 ± 1.61	19.17 ± 1.18	44.15 ± 0.69	7.358	2521
	a2	**64.42 ± 0.48**	**51.89 ± 1.64**	**20.59 ± 1.48**	**45.63 ± 0.87**	6.206	2295

*The iso method represents isolated training and has zero communication cost by definition. Methods prefixed with m denote static ablations with pre-identified modules, while a denotes our adaptive approach where the number indicates the count of selected modules.

TABLE II
COMPARISON OF GRASP SUCCESS RATES FOR DIFFERENT FL METHODS IN REAL-WORLD EXPERIMENTS. THE COMMUNICATION AND COMPUTATION COSTS FOR EACH METHOD ARE IDENTICAL TO THOSE REPORTED IN TABLE I

Method	Scene 1	Scene 2	Scene 3	Scene 4	Scene 5	Overall
t1	70% (10)	63% (11)	70% (10)	63% (11)	62% (13)	**65% (55)**
t2	88% (8)	63% (11)	63% (11)	70% (10)	73% (11)	**71% (51)**
m14	88% (8)	70% (10)	58% (12)	63% (11)	67% (12)	**68% (53)**
m235	88% (8)	70% (10)	63% (11)	63% (11)	73% (11)	**71% (51)**
m2345	77% (9)	77% (9)	63% (11)	70% (10)	62% (13)	**69% (52)**
a2	88% (8)	88% (8)	63% (11)	70% (10)	67% (12)	**73% (49)**

- FedLAMA [25]: An existing FL algorithm known for its layer-wise model aggregation strategy, included for a comprehensive comparison.

Pre-identified $\tilde{M}$ (Ablation): These experiments test our hypothesis by using various pre-identified sets of modules ($\tilde{M}$) for Phase 2. This serves to validate the strategy of targeting specific modules and motivates our adaptive approach. Each variant is denoted by m followed by the indices of the chosen modules (e.g., m14 uses the set $\tilde{M} = \{m_1, m_4\}$). The combinations tested in our ablation study are m1, m14, m35, m45, m135, m235, m245, and m2345.

Proposed Method: This is our main proposal, featuring the adaptive identification of slow-converging modules for Phase 2. For clarity, we denote our adaptive methods as aζ, where ζ indicates the number of modules adaptively selected in each round. The set $\tilde{M}$ is adaptively chosen in each round based on the highest cosine similarity scores. Accordingly, a2 and a3 correspond to selecting the top $\zeta = 2$ and $\zeta = 3$ modules, respectively.

C. Results and Analysis

Our extensive experiments demonstrate that the proposed adaptive two-phase algorithm, particularly a2, achieves a superior balance of performance and resource efficiency for federated GPD. As summarized in Table I, our adaptive method a2 achieves the highest Overall AP of 45.63%, markedly outperforming all other approaches at a comparable communication budget. This trend is also clearly illustrated by the learning curve in Fig. 4.

The Necessity of Federated Learning. First, the poor performance of the iso (isolated client) baseline, with an Overall AP of only 31.49%, confirms the necessity of FL. Without collaboration, models fail to generalize beyond their limited local data, underscoring the importance of a collaborative framework for developing robust GPD models.

Superiority Over Standard FL Baselines. Our adaptive approach, a2, showed improved performance over the standard FL baselines. With an Overall AP of 45.63%, it outperformed t1 (41.76%), t2 (42.28%), and FedLAMA (42.33%). This performance gain was also achieved with greater resource efficiency. The a2 method required 6.206 GB of communication and 2295 s of computation, making it more efficient than all baselines: t1 (10.48 GB, 2365 s), t2 (7.417 GB, 3295 s), and FedLAMA (7.765 GB, 3121 s). This result is notable as it suggests our method does not represent a typical accuracy-cost trade-off. Instead, it achieves a higher AP with a smaller resource budget, indicating that our adaptive strategy provides an efficient path for training high-performance models.

The Critical Role of Adaptive Identification. A key insight from our experiments is the importance of adaptively identifying the slower-converging modules. While several pre-identified module sets in the ablation studies, such as m45 and m235, performed well (44.26% and 44.32% Overall AP respectively), our adaptive a2 method outperformed them, achieving the highest Overall AP of 45.63%. This was also achieved with

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

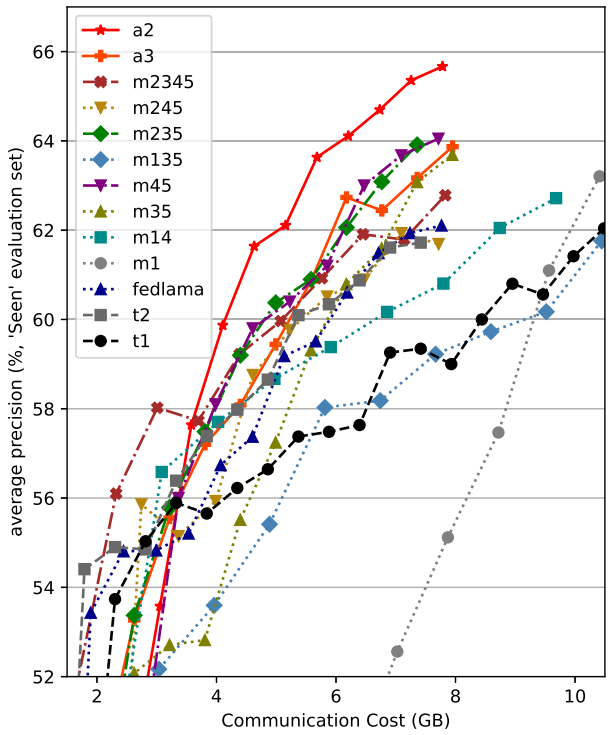

Fig. 4. Performance evaluation of methods on the `Seen` test set. This plot compares model accuracy against the total **communication cost** (in Gigabytes) for several methods.

less communication overhead, as `a2` required only 6.206 GB, compared to 7.714 GB for `m45` and 7.358 GB for `m235`.

This performance gap points to a limitation of the static approach: the set of modules that act as a learning bottleneck is not fixed and can evolve during training. Static methods, which pre-identify a set $\tilde{M}$, rely on a fixed assumption about these learning dynamics. This can lead to inefficient resource allocation, as evidenced by the varied performance across the ablation studies. For instance, the `m1` and `m135` methods used more communication resources (10.41 GB and 10.44 GB) and similar levels of computation than the better-performing `m45` (7.714 GB), yet achieved lower accuracy (41.69% and 41.17% vs. 44.26%). In contrast, our adaptive mechanism re-evaluates the learning dynamics in each communication round. This allows the additional training effort in Phase 2 to be directed toward the modules identified as most critical at that time, which explains the method's improved efficiency and performance.

In summary, these results show that our adaptive module-wise FL strategy offers an efficient path for training high-performance GPD models in a federated setting. By adaptively focusing updates on specific modules, our method achieves higher accuracy with lower computational and communication overhead compared to both standard FL methods and the static ablation approaches.

D. Real-World Experiments

To further validate the practical applicability and performance of our proposed FL framework, we conduct grasping experiments on a physical robotic system.

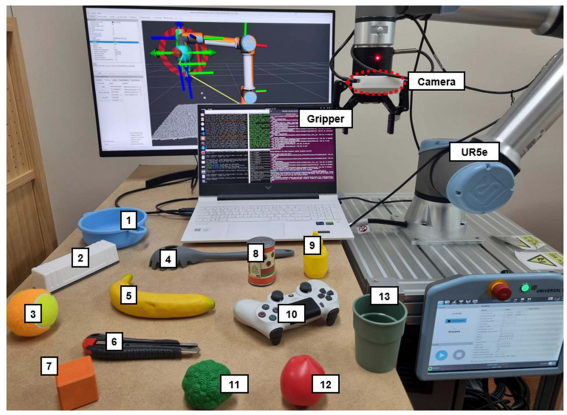

Fig. 5. The experimental setup for real-world grasp validation. The robotic platform consists of a Universal Robots UR5e arm, a Robotiq 2F-85 parallel gripper, and a wrist-mounted Intel RealSense D435 camera. The 13 numbered objects, were used to create cluttered test scenes.

Experimental Setup. Our robotic setup consisted of a Universal Robots UR5e collaborative robot arm equipped with a Robotiq 2F-85 parallel gripper and an eye-in-hand Intel RealSense D435 depth camera for point cloud capture. A host PC running ROS2 served as the control station for communication and grasp execution. To create challenging grasping scenarios, we used a variety of common household and workshop objects, placing 7 to 8 of them in a cluttered arrangement for each experimental run. The complete setup and object set are depicted in Fig. 5. The object sets are as follows: **Scene 1**: {1, 2, 4, 9, 10, 11, 13}; **Scene 2**: {3, 5, 6, 7, 8, 11, 12}; **Scene 3**: {1, 2, 5, 8, 9, 11, 12}; **Scene 4**: {2, 3, 4, 5, 6, 7, 13}; **Scene 5**: {1, 2, 3, 4, 6, 8, 10, 13}.

Methods and Evaluation. We compared the performance of `t1`, `t2`, `m14`, `m235`, `2345`, and `a2`. The evaluation task is to clear a cluttered scene of all its objects. For each run, we measured the total number of grasp attempts required to complete this task. A grasp was deemed successful if an object was securely lifted from the surface. The success rate for each run was then calculated by dividing the number of objects in the scene by the total number of grasp attempts required to clear them.

Results and Discussion. The results of our real-world experiments, summarized in Table II, are consistent with our simulation findings. The primary evaluation metric was the total number of grasp attempts required to clear each cluttered scene. The baseline `t1` model resulted in a 65% success rate. Other methods, such as `t2` and `m235`, performed well, requiring 51 attempts to complete the task (71% success rate). Our proposed `a2` method proved to be the most effective, clearing all scenes with the fewest attempts (49), which corresponds to the highest overall success rate of 73%. This efficiency in grasp attempts suggests that the model trained with our adaptive strategy generates more reliable grasp poses, leading to fewer failures. These results support the practical applicability of our framework in a physical, cluttered environment.

VI. Conclusion

To address the high communication cost of FL on resource-constrained robots, we proposed an adaptive module-wise

framework that reduces data transmission by selectively updating model components. We validated our framework for GPD, where it outperformed standard and static FL methods in both accuracy and resource efficiency. Since modular architectures are common in models for other robotics tasks, our approach has the potential to be applied more broadly as an efficient training strategy for other decentralized robotic systems.

REFERENCES

[1] J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-driven grasp synthesis—A survey," *IEEE Trans. Robot.*, vol. 30, no. 2, pp. 289–309, Apr. 2014.

[2] H.-S. Fang, C. Wang, M. Gou, and C. Lu, "GraspNet-1billion: A large-scale benchmark for general object grasping," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11444–11453.

[3] J. Mahler et al., "DEX-Net 1.0: A cloud-based network of 3D objects for robust grasp planning using a multi-armed bandit model with correlated rewards," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2016, pp. 1957–1964.

[4] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *Int. J. Robot. Res.*, vol. 37, no. 4/5, pp. 421–436, 2018.

[5] A. D. Vuong et al., "Grasp-anything: Large-scale grasp dataset from foundation models," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2024, pp. 14030–14037.

[6] L. Pinto and A. Gupta, "Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2016, pp. 3406–3413.

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

[8] S. Gummadi, M. V Gasparino, D. Vasisht, and G. Chowdhary, "Fed-EC: Bandwidth-efficient clustering-based federated learning for autonomous visual robot navigation," *IEEE Robot. Automat. Lett.*, vol. 9, no. 12, pp. 11841–11848, Dec. 2024.

[9] Y. Wang, S. Zhong, and T. Yuan, "Grasp control method for robotic manipulator based on federated reinforcement learning," in *Proc. 7th Int. Conf. Adv. Algorithms Control Eng.*, 2024, pp. 1513–1519.

[10] S.-K. Kang and C. Choi, "FOGL: Federated object grasping learning," in *Proc. 2023 IEEE Int. Conf. Robot. Automat.*, 2023, pp. 5851–5857.

[11] C. Wang, H.-S. Fang, M. Gou, H. Fang, J. Gao, and C. Lu, "Graspness discovery in clutters for fast and accurate grasp detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 15964–15973.

[12] H. Wang, W. Niu, and C. Zhuang, "GraNet: A multi-level graph network for 6-DoF grasp pose generation in cluttered scenes," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2023, pp. 937–943.

[13] X.-M. Wu, J.-F. Cai, J.-J. Jiang, D. Zheng, Y.-L. Wei, and W.-S. Zheng, "An economic framework for 6-DoF grasp detection," in *Proc. Eur. Conf. Comput. Vis.*, 2024, pp. 357–375.

[14] F. Sattler, K.-R. Müller, and W. Samek, "Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 8, pp. 3710–3722, Aug. 2021.

[15] J. Ma, G. Long, T. Zhou, J. Jiang, and C. Zhang, "On the convergence of clustered federated learning," 2022, *arXiv:2202.06187*.

[16] H. Zhang, X. Lan, X. Zhou, Z. Tian, and N. Zheng, "ROI-based robotic grasp detection for object overlapping scenes," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 4768–4775.

[17] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *Int. J. Robot. Res.*, vol. 34, no. 4-5, pp. 705–724, 2015.

[18] Z. Liu, Z. Chen, S. Xie, and W.-S. Zheng, "TransGrasp: A multi-scale hierarchical point transformer for 7-DoF grasp detection," in *Proc. Int. Conf. Robot. Automat.*, 2022, pp. 1533–1539.

[19] H. Ma and D. Huang, "Towards scale balanced 6-DoF grasp detection in cluttered scenes," in *Proc. Conf. Robot Learn.*, 2023, pp. 2004–2013.

[20] N. Majcherczyk, N. Srishankar, and C. Pinciroli, "Flow-FL: Data-driven federated learning for spatio-temporal predictions in multi-robot systems," in *Proc. 2021 IEEE Int. Conf. Robot. Automat.*, 2021, pp. 8836–8842.

[21] C. Miao et al., "FedVLA: Federated vision-language-action learning with dual gating mixture-of-experts for robotic manipulation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2025, pp. 6904–6913.

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

[23] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, "QSGD: Communication-efficient SGD via gradient quantization and encoding," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 1707–1718.

[24] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Robust and communication-efficient federated learning from non-i.i.d. data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 9, pp. 3400–3413, Sep. 2020.

[25] S. Lee, T. Zhang, and A S. Avestimehr, "Layer-wise adaptive model aggregation for scalable federated learning," in *Proc. AAAI Conf. Artif. Intell.*, 2023, vol. 37, pp. 8491–8499.

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

[27] P. Sai et al., "Scaffold: Stochastic controlled averaging for federated learning," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 5132–5143.

[28] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet : Deep hierarchical feature learning on point sets in a metric space," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5105–5114.

[29] T. Wang et al., "Personalized federated learning via heterogeneous modular networks," in *Proc. IEEE Int. Conf. Data Mining*, 2022, pp. 1197–1202.

[30] L. Yi et al., "pFedMoE: Data-level personalization with mixture of experts for model-heterogeneous personalized federated learning," 2024, *arXiv:2402.01350*.