

Privacy-Preserving Efficient Federated Learning for Internet of Medical Things under Data Heterogeneity

Abstract—With the rapid development of the Internet of Medical Things (IoMT), healthcare institutions are collecting large volumes of data from medical devices and using this data to train machine learning models. Federated learning (FL) based on IoMT devices has emerged as a prominent research focus. FL enables distributed training of global machine learning models across multiple IoMT devices without accessing clients' private data. However, data heterogeneity across clients significantly reduces model accuracy. Recent studies indicate that sharing client information can help address this issue. However, because such information often contains private data, these strategies may lead to privacy breaches. Therefore, achieving a trade-off between privacy protection and model accuracy has become a critical challenge. We propose a federated learning solution—FedTPF—that achieves this privacy-accuracy trade-off under data heterogeneity. FedTPF extracts the minimal feature information from raw data that meets the global model's generalization requirements while filtering out redundant information. This minimal feature information is then shared with clients using privacy-preserving techniques. Subsequently, each client trains its local model using both local and shared data to mitigate data heterogeneity. By discarding a large amount of redundant features, FedTPF significantly reduces the privacy cost of the shared data, enabling the model to achieve higher accuracy without compromising client privacy. Theoretical analysis and extensive experiments demonstrate that FedTPF improves test accuracy by 3.7% compared to state-of-the-art methods.

Index Terms—Federated learning, Feature Distillation, Prototype learning, Privacy protection, Model regularization.

I. INTRODUCTION

IoMT is a rapidly growing field, with intelligent healthcare systems generating medical data at an exponential rate [1], covering nearly every aspect of the healthcare domain [2], [3]. Today, medical sensors are embedded in wearable devices, home appliances, and hospital equipment within smart healthcare systems. These intelligent medical devices perform functions such as data collection, real-time monitoring, diagnosis, treatment, and disease management.

The vast amount of data collected through the IoMT creates opportunities for the development and integration of advanced computational technologies to analyze, process, and store data—ultimately improving treatment and care quality [4], [5]. Because the IoMT consists of data from a wide range of sensors and devices, it is inherently decentralized. A large amount of healthcare data is often scattered across different institutions and individuals, with various stakeholders, such as hospitals, laboratories, and physicians having access to critical patient information. These medical records contain highly sensitive personal health information, requiring secure access and minimal communication delays [6]. This poses numerous challenges for various advanced machine learning data processing techniques, which typically require vast amounts of training data [7].

To address this challenge, FL provides an effective solution [8], [9]. FL offers a sophisticated mechanism for collaborative training of high-quality shared models. This collaborative approach involves aggregating local computational updates contributed by IoMT devices. It consists of two main steps: (1) clients train local models and upload the updated local models to a central server, and (2) the server aggregates these updates to create a global model, which is then redistributed to clients for further training. The core principle of FL is to preserve the privacy of sensitive medical data, which remains on the client side and is not shared directly with the server or other clients.

However, Federated Learning (FL) faces a fundamental issue: significant data heterogeneity among clients can substantially reduce model accuracy [10], [11]. For instance, when multiple smart medical devices—such as wearable heart rate monitors, remote blood pressure monitors, portable glucometers—collaboratively train a unified global model, the data collected by each device can vary widely [12]. These devices are often produced by different manufacturers, with differences in sampling frequencies, sensor precision, and data formats. As a result, the local datasets are typically non-independent and identically distributed (non-IID), with imbalanced class distributions. Some devices may be unable to capture certain physiological indicators or may only activate sensors under specific scenarios, leading to gaps in key feature data. This causes a misalignment between the local training objectives of individual devices and the optimization objectives of the global model. Consequently, after model aggregation, performance may degrade, reducing the global model's generalization and accuracy across multi-source medical data. To address the challenges of data heterogeneity, sharing client information is considered an effective solution. The shared information may include raw data, statistical information, or synthetically generated data [13]–[16]. For example, [17] proposed alleviating the heterogeneity issue in FL by sharing a small amount (e.g., 5%) of raw data. This approach improves model accuracy to some extent, but it introduces significant privacy risks—especially in IoMT settings, where raw medical data often contains highly sensitive personal health information. As such, the potential for privacy leakage remains a major concern.

Similarly, [18] re-examined the issue of data heterogeneity in FL from a new perspective, proposing a method that combines local learning with the sharing of statistical information to address heterogeneity challenges. This method significantly enhances model accuracy in heterogeneous data environments while effectively preventing the leakage of raw data. However, their limitation lies in the fact that statistical information inherently represents only a simple characterization of data

distributions, making it challenging to capture the rich relationships and diversity present in the data. [19] examined various synthetic data generation techniques and proposed strategies to enhance the generalization capability of synthetic data, improving its applicability in heterogeneous data conditions and boosting global model accuracy. Synthetic data retains more distribution structures, class-specific characteristics, and other critical information, making it beneficial for calibrating models across diverse client data distributions. Sharing synthetic data achieves the dual goals of preventing raw data leakage and improving model accuracy in heterogeneous data environments. Unfortunately, because shared synthetic data often carries substantial original information, it is vulnerable to inference attacks [20], which could reconstruct the raw data. To address this issue, [21] proposed a synthetic data generation method incorporating differential privacy. This method applies differential privacy noise to the statistical information of local client data before sharing it. The server then uses this protected statistical information to generate synthetic data, which is subsequently used by clients in conjunction with their local data to train local models.

Although this method introduces noise to protect shared data, it fails to strike a balance between model accuracy and privacy protection. We identify a key limitation of existing solutions: while sharing client information (such as statistics or synthetic data) proves effective in mitigating data heterogeneity across medical devices, this sharing inevitably triggers a trade-off between privacy and accuracy, as IoMT device data involves highly sensitive personal health information. This trade-off becomes a bottleneck for further improving model performance, especially in medical environments that require fine-grained modeling and real-time responses. To address this issue, we propose a new approach: can we achieve a more efficient federated learning framework while protecting the privacy of IoMT device user data? Inspired by information bottleneck theory [22], we suggest using a distillation model to filter shared information, removing redundant data from raw or intermediate features to reduce the overhead required for privacy protection. In IoMT applications, reducing privacy overhead means less noise needs to be added, allowing the model to more accurately capture the true distribution of data from various medical devices while maintaining privacy, thus improving the final training performance. To achieve this, we employ a competitive mechanism between the generator and the classifier to extract the minimal generalized information (see Section 3.2). Additionally, we address two key challenges: the reconstruction constraint of the generator leads to insufficient feature filtering (see Section 3.3), and the large differences in data distribution between different medical devices result in inconsistent convergence speeds of the feature distillation model across clients, which can cause overfitting issues (see Section 3.4). Experimental results show that FedTPF performs excellently in terms of accuracy across multiple datasets while ensuring the privacy and security of user medical data.

Our main contributions are as follows:

- We propose a federated learning framework combining feature distillation and prototype learning to address the

privacy-accuracy trade-off under data heterogeneity. By leveraging the competition between the generator and classifier, we effectively remove redundant features, retaining those beneficial for global generalization.

- We introduce a novel generator latent space regularization strategy by applying local prototypes to the feature distillation model. This addresses the issue of insufficient feature filtering caused by reconstruction constraints, avoiding prototype aggregation and preventing privacy leakage at the source.
- We propose a threshold-based adaptive weight decay aggregation mechanism to mitigate model overfitting caused by varying convergence speeds of feature distillation models across clients. For early-converging models, parameters are frozen, and adaptive weight decay is applied during aggregation based on global model accuracy changes.
- We use three types of distributions (Dirichlet distribution $\text{Dir}(\alpha)$, $\#C = k$ distribution, Subset distribution) to simulate client data heterogeneity, our experiments demonstrate that FedTPF improves test accuracy by 3.7% compared to state-of-the-art methods while preserving client privacy, showcasing its effectiveness in reducing model bias caused by data heterogeneity.

II. RELATED WORK

Federated Learning models typically perform poorly when faced with severely Non-independent and identically distributed (non-IID) data [23]. To mitigate data heterogeneity, advanced work often introduces additional auxiliary loss functions or improves model aggregation schemes to address cross-client data heterogeneity in FL.

To prevent local models from converging to local minima instead of the global minimum, study have focused on controlling local updates from the perspective of weight space. Adap-FedITK [24] alleviates the negative impact of differential privacy noise on model accuracy by dynamically adjusting gradient clipping thresholds, achieving a trade-off between privacy protection and model accuracy. Another study have developed alternative aggregation schemes on the server side to address the issue of data heterogeneity in federated learning. FedKF [25] leveraged local knowledge generated by each client and dynamically fuses global and local models to improve generalization and personalization [26], effectively mitigating the challenges posed by uneven data distribution. By controlling weight divergence and improving aggregation schemes, these methods avoid the direct transmission of raw client data, making them relatively effective in preserving individual privacy. However, their limitation lies in the inability to fully exploit the information contained in the raw data, resulting in model updates that fail to fully utilize the generalizable features of each client.

Recent studies have identified feature inconsistencies among clients from the perspective of feature space. To mitigate model divergence, several contrastive learning techniques, such as feature alignment and feature distillation, have been adopted. FedDr+ [27] enhanced feature alignment in local

models by freezing the classifier as a simplex ETF, incorporates a point regression loss to address client drift, and employs feature distillation to retain information about unseen classes, thereby improving both global and personalized model accuracy. FedTGP [28] proposed trainable global prototypes and adaptive margins to enhance contrastive learning and align client feature distributions, addressing data heterogeneity issues. Methods addressing feature inconsistency involve sharing the feature representations learned by clients with the server, which aggregates and redistributes them to all clients. However, the processes of sharing and redistribution may introduce potential privacy leakage risks, and without proper safeguards, they could indirectly reveal certain data distribution information.

Sharing feature representations falls under the category of client information-sharing methods, which are considered direct and promising approaches for mitigating data heterogeneity [29]. Research has demonstrated that sharing a limited amount of generalizable information can significantly enhance model accuracy. However, the conflicting goals of protecting data privacy and improving model accuracy hinder the practical effectiveness of information-sharing strategies. While the security guarantees of injecting random noise into data for privacy protection have been theoretically proven [24], [30], the challenge lies in addressing the decline in global model accuracy caused by applying noise to shared data [31]. Inspired by the strategy of separating spurious and robust features [32], we divide data into generalizable features beneficial for the global model and redundant features irrelevant to global generalization. By sharing only the generalizable features and retaining most redundant client data features locally, the required noise intensity for protecting shared global features is significantly reduced. This approach effectively resolves the privacy-accuracy trade-off.

TABLE I: Notation table of the FedTPF Algorithm

Symbol	Description
α	heterogeneity degree
T_d	communication round of feature distillation
T_r	communication round of classifier training
E_d	local epochs of feature distillation
E/E_r	local epochs of classifier training
X_s	Features promoting global model generalization
X_p	Shared features (distributed to each client)
σ_s^2	DP noise level, added to x_s
$ C_t / C_r $	selected clients every communication round
K	clients of federated system
θ_f	Feature extractor parameters
f	Embedding function of the feature extractor
\mathbf{w}^0	Parameters of the Distillation Model Generator
θ^0	Parameters of the Distillation Model Classifier
n_m	Total number of samples of class m in all clients
n_m^k	Number of samples of class m in client k
c_k	Label counter for client k
$\omega_k^{t,m}$	Local prototype for class m obtained by client k in round t
D^s	Global shared dataset
D^k	Client local dataset
n	Add noise (Laplacian noise or Gaussian noise)

III. METHODS

A. Proposed Solution Overview

To address the privacy-accuracy trade-off in mitigating client data heterogeneity using shared client information, we propose a federated framework based on feature distillation and prototype learning with adaptive weighted aggregation(FedTPF). FedTPF consists of two stages training, with the transition from the first stage to the second determined by pre-defined threshold conditions. The overall architecture of the FedTPF framework is illustrated in Figure 1, algorithmic logic is detailed in Algorithm 1.

In the Stage 1, Client 1 to N client are training the local model while simultaneously computing and statistically analyzing local prototypes (step ① and ②). A local prototype comprises multiple class prototypes, where the number of class prototypes corresponds to the number of categories in the local dataset. Each class prototype is derived by aggregating feature vectors of all samples belonging to the same class, serving as the “feature center” that represents the feature distribution of the class). The generated local prototypes, along with local data, are input into the distillation model for training (step ③). Subsequently, the parameters of both the local model and the distillation model are uploaded to the server for aggregation (step ④). The aggregation of the distillation model employs an adaptive weight decay aggregation mechanism to mitigate training issues caused by varying convergence speeds under different data distributions). Each dispatched distillation model(step ⑤) is evaluated on the local test set of each client. When the distillation model on every client meets the predefined condition (achieving 90% classification accuracy on the classifier using generalized features obtained from the distillation model), the training of the distillation model and the computation of local prototypes are halted, transitioning into the second phase. Otherwise, the first phase process is iterative (as shown in the middle).

In the first round of the Stage 2, each client uses the global distillation model received from the server to distill features from its local data. The generalized features generated by all clients (protected generalized features; details in Section 3.4) are collected to form a globally shared dataset(illustrated in the bottom-left corner). In subsequent rounds, the client’s local models are trained using both the globally shared dataset and local data to address data heterogeneity issues(step ① ② ③).

B. Feature Distillation

To extract the minimal features required for global generalization from each client, we propose a feature distillation model to filter the raw image x . The ideal filtering result retains only the minimal sufficient information necessary for global model generalization [22], [33], while all other data features are discarded and remain local to the client. For clarity, we denote local features as x_r and generalizable features as x_s . A careful analysis of the above objective indicates that minimizing the quantity of generalizable features implies the inability to directly use traditional distillation models

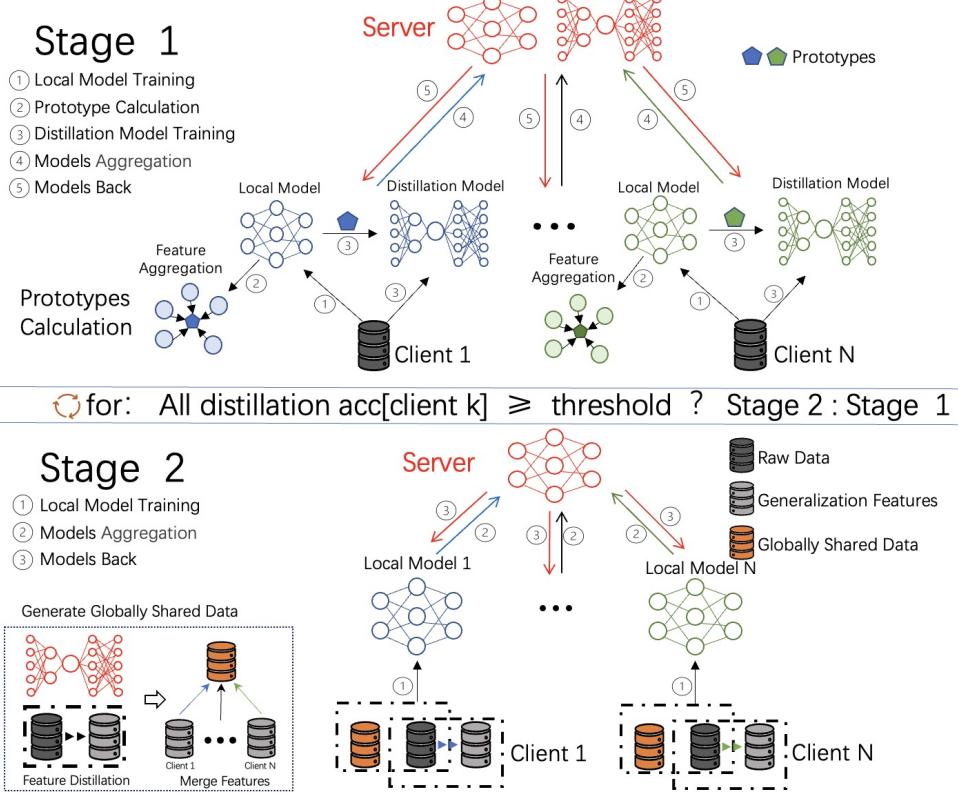


Fig. 1: Overview of FedTPF

for feature filtering to remove redundant features effectively. This is because the feature filtering efficiency of existing distillation models for raw images falls far short of meeting the required objectives. If private information contained in the raw image—particularly privacy-sensitive details unrelated to generalization—is excessively included in the generalizable features x_s , even when x_s is protected with differential privacy(DP) [34] , the inclusion of excessive private information may still lead to privacy leakage risks.

In contrast to direct distillation, our proposed method introduces a competitive mechanism [35], where a generator and a classifier are adversarially trained (with opposing loss functions) to indirectly achieve the desired feature distillation effect. For the distillation model,

The adversarial training between the generator and the classifier enables feature distillation through the following mechanism: the generator aims to reconstruct the input image as completely as possible, while the classifier attempts to perform classification using only the difference between the input and the reconstructed image. These two objectives are inherently antagonistic—the generator strives to retain as much information as possible for perfect reconstruction, whereas the classifier forces the generator to preserve only the most critical discriminative features. This adversarial relationship creates a dynamic balance via backpropagation: when the generator retains excessive redundant features, the classifier can easily achieve high accuracy by leveraging the extra information, thereby encouraging the generator to further compress the features; conversely, when excessive compression degrades

classification performance, gradient signals from the classifier guide the generator to retain essential features. As training progresses, the difference between the input and the reconstructed image (i.e., the distilled generalizable feature x_s) gradually converges to the minimal sufficient representation—containing enough discriminative information while filtering out privacy-related, non-generalizable details. The training process of the distillation model can be formally expressed as follows:

Specifically, we train the generator model to reconstruct an image \hat{x} that closely approximates the original image x . The optimization objective can be formulated as:

$$g^* = \arg \min_{\theta} \|x - q(x; \theta)\|_2^2 \quad (1)$$

where x is the raw image, $q(x; \theta)$ is reconstructed image, generated by the generator.

The classifier is subsequently trained on the residual difference between the raw image and its reconstructed counterpart, $x - \hat{x}$, aiming to maximize its ability to accurately predict the original raw image x . The precise optimization objective is defined as:

$$w_k^* = \arg \min_{w_k} \mathcal{L}(f(x - \hat{x}; w_k), y) \quad (2)$$

The term $x - \hat{x}$ represents the difference between the raw image x and the reconstructed image \hat{x} . The function $f(x - \hat{x}; w_k)$ denotes the output of the classifier applied to the difference image $x - \hat{x}$, where the classifier is parameterized by w_k . The variable y represents the true label of the raw image x .

By performing a certain amount of training on the generator and classifier using the client's local data, allowing both the

Algorithm 1 FedTPF Algorithm

Server Input: initial global model ϕ^0 , communication round T_r , communication round T_d .
Client k's input: $E_r, E_d, \mathcal{D}^k, \sigma_s^2$, learning rate η_k .
Initialization: server distributes the initial model $\phi^0, \mathbf{w}^0, \theta^0$ to all clients.

```

if all accuracy[k] ≥ threshold then
    Distribute  $\mathcal{D}^s$  to all clients
    Server Executes:
        for  $t = T_d, T_d + 1, \dots, T_r$  do
            server samples a subset of clients  $C_r \subseteq \{1, \dots, K\}$ 
            server communicates  $\phi^r$  to selected clients
            for each client  $k \in C_r$  in parallel do
                 $\phi_k^{r+1} \leftarrow \text{ClientTraining}(k, \phi^r, \mathcal{D}^k \cup \mathcal{D}^s)$ 
            end for
             $\phi^{r+1} \leftarrow \text{AGG}(\phi_k^{r+1})$ 
        end for
    else
        Server Executes:
        for  $t = 1, \dots, T_d$  do
            server samples a subset of clients  $C_t \subseteq \{1, \dots, K\}$ 
            server communicates  $\phi^r, \mathbf{w}^t, \theta^t$  to selected clients
            for each client  $k \in C_t$  in parallel do
                 $\phi_k^{r+1} \leftarrow \text{ClientTraining}(k, \phi^r, \mathcal{D}^k)$ 
                local prototype  $m$  class  $\omega_k^{t,m} = \sum_{i=1}^{n_k^m} F_{\Pi_k}(x_i^m) / n_k^m$ 
                traverse all classes to get  $\omega_k^{t+1}$ .
                 $\mathbf{w}_k^{t+1}, \theta_k^{t+1} \leftarrow \text{LocalDisTraining}(\mathbf{w}^t, \theta^t, \sigma_s^2, \omega_k^t)$ 
            end for
             $\mathbf{w}^{t+1}, \theta^{t+1} \leftarrow \text{AGG}(\mathbf{w}_k^t, \theta_k^t, k \in C_t)$ 
             $\phi^{r+1} \leftarrow \text{AGG}(\phi_k^{r+1}, k \in C_t)$ 
        end for
    end if

```

Training function

LocalDisTraining($\mathbf{w}^t, \theta^t, \sigma_s^2, \omega_k^t$ **:**
for each local epoch $e = 1, \dots, E_d$ **: do**
 $\mathbf{w}_k^{t+1}, \theta_k^{t+1} \leftarrow \text{SGD update use Eq (1)(2)}$.
end for
Return $\mathbf{w}_k^{t+1}, \theta_k^{t+1}$ to server

ClientTraining($k, \phi^r, \mathcal{D}_t^k$ **:**
 ϕ^r initialize local model ϕ_k^r
for each local epoch $e = 1, 2, \dots, E_r$ **: do**
 $\phi_k^{r+1} \leftarrow \text{SGD update with } \mathcal{D}_t^k$.
end for
Return ϕ_k^{r+1} to server

generator and the classifier to reach a convergent state, we observe that the difference between the raw image and the reconstructed image, $x - \hat{x}$, approximates the minimal sufficient information required for global model generalization. Specifically, using $x - \hat{x}$ (the difference between the raw image and the reconstructed image) for label prediction achieves an accuracy comparable to directly using the raw image x for

prediction.

$$\min_{w_k} \mathcal{L}(f(x - \hat{x}; w_k), y), \quad \text{s.t. } \mathcal{A}(f(x - \hat{x}; w_k)) \approx \mathcal{A}(f(x; w_k)) \quad (3)$$

Thus, we use the distillation model, composed of a generator and a classifier, to obtain the generalized features x_s . The overall optimization objective for the competition between the generator and the classifier is expressed as:

$$\min_{\theta, w_k} \mathcal{L}(f(x - q(x; \theta); w_k), y), \quad \text{s.t. } \|x - q(x; \theta)\|_2^2 \leq \rho \quad (4)$$

$f(x - q(x; \theta); w_k)$: A classifier that takes the residual $x - q(x; \theta)$ as input, parameterized by w_k . The goal is to classify the input data based on the residual information to identify y . $\|x - q(x; \theta)\|_2^2 \leq \rho$: The reconstruction constraint for the generator, indicating that the reconstructed sample $q(x; \theta)$ generated by the generator should approximate the original data x as closely as possible, with the reconstruction error not exceeding the threshold ρ .

From the above formulas, it is evident that the reconstructed image \hat{x} essentially represents the local features x_r , while the difference between the raw image and the reconstructed image, $x - \hat{x}$, effectively represents the generalized features x_s . We apply random noise n to the generalized features x_s , enabling the secure sharing of protected features x_p between clients, where $x_p = x_s + n$. Each client contributes its protected features x_p to construct the global shared dataset. Subsequently, each client trains its local model using both its local data and the global shared dataset, thereby addressing the data heterogeneity problem. The workflow of the feature distillation model is shown in Figure III-B. The original data x is passed through the generator, which, guided by the local prototype, applies latent space regularization [36] during the reconstruction process to produce reconstructed data x_r . On one hand, x and x_r are fed into the discriminator to obtain the output and the discriminator loss. On the other hand, the generalized data x_s , defined as $x_s = x - x_r$, is perturbed with noise n to yield protected generalized data x_p , i.e., $x_p = x_s + n$. The protected data x_p is then input into the classifier to obtain predictions and compute the classification loss. In addition to these, the generator's reconstruction loss and the Euclidean distance constraint between the latent variable z and the local prototype are also considered. These four loss components collectively guide the model's parameter updates.

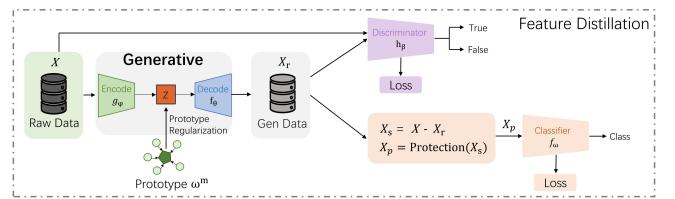


Fig. 2: The Workflow of Feature Distillation

C. Generator Latent Space Regularization Strategy Based on Local Prototypes

The competition between the generator and the classifier addresses the problem of minimizing generalizable features.

Through the filtering of the distillation model, we obtained a sufficiently small set of features, which preliminarily meet the requirements for global generalization. However, a limitation lies in the reconstruction constraint of the generator, which causes a considerable portion of features beneficial for global model generalization to be excluded from global sharing. Whether these features can be effectively separated from the raw image directly impacts the generalization accuracy of the global model.

The feature distillation model mentioned in the previous section uses a competitive mechanism (competition between training the generator and training the classifier) to separate local features and generalizable features. Since the generator's objective is to reconstruct the image x_r to maximize its similarity to the raw image x , this indirectly causes a significant portion of features relevant to global model generalization but non-determinative for classifier predictions, denoted as x_s ($x_s = x - x_r$), to be retained locally as part of the reconstructed image x_r .

To thoroughly filter out irrelevant features, we introduce a prototype mechanism [37] applied to the generative model (VAE-WGAN-GP [38]) to prevent the reconstructed image x_r from overfitting [39]. Our design involves aligning the latent vector z , sampled through the reparameterization technique [40] in the VAE-WGAN-GP model, with the local prototype ω_k^t generated in the current round. Specifically, the Euclidean distance between the latent variable z of each sample and its corresponding local prototype ω_k^t is minimized. During each round of generator training, the generator receives prototypes statistically calculated from the local model after training (Since prototypes do not participate in aggregation, the number of prototypes theoretically has no impact on privacy):

$$\omega_k^{t,m} = \frac{1}{n_k^m} \sum_{i=1}^{n_k^m} F_{\Pi_k}(x_i) \quad (5)$$

$\omega_k^{t,m}$ represents the local prototype of class m computed by client k ; x_i represents a sample of class m ; n_k^m represents the number of samples belonging to class m on client k . Π denotes the parameters of the feature extractor, and F_{Π} is the embedding function of the feature extractor.

The VAE-WGAN-GP is trained using local data and prototypes. The encoder maps the input data x_i to the latent space, generating the distribution parameters mean μ_i and variance σ_i^2 . Using the reparameterization trick, noise ϵ is sampled from the standard normal distribution $\mathcal{N}(0, I)$, and the latent variable is calculated as $z_i = \mu_i + \sigma_i \odot \epsilon$. Subsequently, the Euclidean distance between z_i and the local prototype ω_k^t is computed to define the constraint loss function in the latent space, expressed as follows:

$$L_{po} = \frac{1}{B_k} \sum_{i=1}^{B_k} \| (f_\mu(x_i; \phi) + f_\sigma(x_i; \phi) \odot \epsilon) - \omega_k^t \|_2^2 \quad (6)$$

$f_\mu(x_i; \phi)$: The mean of the latent distribution output by the encoder based on the input x_i . $f_\sigma(x_i; \phi)$: The standard deviation of the latent distribution output by the encoder based on the input x_i . ϵ : Noise vector sampled from the standard normal distribution $\mathcal{N}(0, I)$, used for reparameterization.

Combining the loss of the generator (VAE) [41], the loss of the discriminator (WGAN-GP) [42], and the cross-entropy loss of the classifier, the overall loss function of the feature distillation model can be expressed as follows:

$$Loss_{total} = l_1 Loss_{VAE} + l_2 Loss_{WGAN-GP} + l_3 Loss_{po} + l_4 Loss_{ce} \quad (7)$$

l_1, l_2, l_3 , and l_4 represent the coefficients.

D. Federated Training Framework with Threshold Conditions

In our design, the primary objective of the feature distillation model is to separate local features from generalized features. During the first stage, the local model and the feature distillation model are trained simultaneously, due to the varying convergence speeds of the global distillation model across different client distributions, failing to handle the prematurely converged model parameters could lead to overfitting of the global distillation model on certain client data. Therefore, determining an appropriate timing to stop training the client-side feature distillation model and enabling its meaningful participation in subsequent model aggregation is crucial. Our approach is to freeze the local distillation model parameters on a client once it reaches a pre-defined efficiency threshold (The classifier achieves 90% accuracy in classifying the input images, and the generator reaches a convergent state. This indicates that the client distillation model has achieved its main objective: separating local features from generalized features. Continuing training and aggregation beyond this point may lead to performance degradation due to overfitting. To determine the appropriate timing for freezing the parameters of the distillation model, we adopt a comprehensive multi-dimensional evaluation criterion. When local testing indicates that the classifier achieves a classification accuracy of $90\% \pm 2\%$ on the generalizable features x_s (with a 95% confidence interval), and the cosine similarity between the generator's reconstructed features x_r and the generalizable features x_s remains consistently below 0.15, with both indicators fluctuating by no more than 5% over three consecutive training epochs, we consider the client to have completed the core task of feature disentanglement and freeze the model parameters accordingly.) and notify the server. Subsequently, the client stops training the distillation model and halts prototype computation. For clients that have ceased distillation training, the server employs an adaptive weight decay aggregation strategy: Starting from the next round, the influence of this client on the global model is no longer fixed, but gradually decays. The rate of decay depends on the accuracy difference between the global model's performance in the previous and current rounds: if the accuracy difference is large, the aggregation weight calculated from the accuracy difference will increase, slowing down the decay; if the global model's accuracy stabilizes and continues to improve, the decay will accelerate.

$$\alpha_k(t) = \begin{cases} 1, & t < t_s(k) \\ \exp(-\lambda \cdot \max(0, \Delta_{perf}(t))), & t \geq t_s(k) \end{cases} \quad (8)$$

$\alpha_k(t)$ denote the decay coefficient of client k at round t , $t_s(k)$ represent the round when client k stops distillation, λ

be the decay intensity hyperparameter, and $\Delta_{\text{perf}}(t)$ denote the relative accuracy change of the global distilled model on the test set compared to when distillation stops.

$$\Delta_{\text{perf}}(t) = \frac{1}{|\mathcal{C}_t|} \sum_{k \in \mathcal{C}_t} [p_k(t) - p_k(t-1)] \quad (9)$$

\mathcal{C}_t represents the set of clients participating in aggregation during the current round that are still undergoing distillation training. $p_k(t)$ denotes the accuracy of the global distillation model on the test set of client k after the completion of aggregation in round t .

Greater weight is assigned to these models during aggregation to retain the essential knowledge from early convergence. The decay rate is then dynamically adjusted based on the real-time accuracy of the global distillation model on various test sets. Specifically, if the global distillation model exhibits significant fluctuations on a test set, the decay slows down to preserve more knowledge from early-stopping clients. Conversely, when global accuracy stabilizes and improves, the decay accelerates, allowing distillation models from clients that have not yet met the threshold to play a more dominant role in subsequent iterations. Until the transition to the second stage of training.

Using the relative accuracy change of the global distillation model as the core variable for adjusting the decay rate can adapt to the global model's learning process. It dynamically adjusts the contribution of clients at different stages, ensuring stability in the early stages of training, accelerating optimization convergence in the later stages, and balancing the influence of different clients. This effectively improves the convergence efficiency of federated learning and the generalization performance of the model. From the perspective of dynamic model evolution, the change in relative accuracy can intuitively reflect the learning state of the global model [43]. When the accuracy fluctuates significantly (e.g., changes exceeding 5% in the first two rounds), it indicates that the model is still in a rapid knowledge acquisition phase. In this case, slowing the decay rate helps retain the stable features extracted by already converged clients. Conversely, when the accuracy change becomes mild (fluctuation less than 2%), it suggests that the model is approaching convergence. Accelerating the decay at this stage facilitates quicker integration of new knowledge.

E. Privacy Protection

Since the globally shared dataset we construct inevitably contains personal privacy information, the purpose of introducing protection methods is to safeguard the privacy-related features within the globally shared dataset. This requires the adopted protection method to possess robustness against privacy attacks [44], [45]. Based on the above analysis, differential privacy (DP) [reference] is naturally suited for the feature distillation scenario in our design. Therefore, we employ differential privacy (DP) to protect the generalization-related features before globally sharing them. Specifically, noise (e.g., Gaussian noise or Laplace noise). Laplace noise, due to its sharp peak and heavy-tailed distribution, is particularly well-suited for static data-sharing scenarios with stringent privacy

requirements. This distribution effectively conceals extreme feature values in the data, significantly reducing the risk of reconstructing original data through statistical inference, and thereby provides stronger protection for outliers commonly found in medical datasets. In contrast, within the dynamic training environment of federated learning, Gaussian noise demonstrates distinct advantages. Its mathematical stability and composability make it well-equipped to handle the cumulative effect of privacy leakage across multiple training rounds, ensuring that the overall privacy budget remains within a controllable range as the number of iterations increases.) is applied to the generalizable features x_s ,

Then transforming them into protected shared generalizable features x_p [46], [47], defined as $x_p = x_s + n$. The protected features x_p are then collected to construct the global dataset, which is subsequently shared across all clients.

Compared to sharing raw features x directly, we decompose it into two parts: the generalizable features x_s (with proportion ρ , a small fraction, where x_s is derived from the distillation model described in Section 3.1) and the local features x_r (with proportion $1-\rho$), such that $x = x_s + x_r$. When sharing the raw features x , both x_s and x_r must be perturbed with noise to meet privacy requirements, resulting in a combined privacy budget ε' that is affected by both components. However, in FedTPF, since x_r is retained locally and inaccessible to external adversaries, it is effectively protected with infinitely large noise ($\sigma_r \rightarrow \infty$), making the corresponding privacy cost $\varepsilon_r \approx 0$. Therefore, only the privacy budget for x_s needs to be considered. This implies that, under the same privacy budget constraint, FedTPF allows the use of a lower noise level to protect x_s compared to directly sharing the full raw feature x . Moreover, the noise level σ_s applied to x_s is proportional to the magnitude of x_s . This means that the more effective the distillation model is (the smaller x_s is), the lower the required noise σ_s will be. As a result, FedTPF can preserve more useful information while still ensuring privacy protection. In other words, FedTPF adopts a strategy of "protecting only the necessary sensitive information", thereby avoiding the performance degradation caused by indiscriminately adding noise to all features. Consequently, FedTPF achieves better model training performance under the same level of privacy protection.

IV. EXPERIMENT

A. Experiment Settings

We validate our method on three datasets and compare it with six methods. For the federating scenario, our experiments use Latent Dirichlet Allocation (LDA) [48] to simulate non-IID data distributions, and we set $\alpha = 0.1$, $\alpha = 0.3$ and $\alpha = 1.0$ in the LDA configuration. LDA allows flexible adjustment of the data class distribution on each client, ranging from highly imbalanced to near non-IID, to simulate the variability in patient data collection across different healthcare institutions or devices in an IoMT environment. For instance, large comprehensive hospitals may cover a wide range of categories and a substantial amount of disease data, while certain community clinics or specialized departments may primarily handle specific types of diseases with smaller datasets.

Additionally, we employ two other widely adopted partitioning strategies, label skew and quantity skew, to evaluate FedTPF. As shown in Figure 3, different colors represent different labels, and the length of each row indicates the amount of data for each client. We illustrated three types of non-IID data distributions: (a) an LDA distribution with $\alpha = 0.1$, where each client holds only a few classes and exhibits quantity imbalance; (b) an extreme label skew distribution where each client contains data from only $C = 2$ classes; and (c) an extreme quantity skew (Subset), where over 90% of the data on each client belongs to a single class.

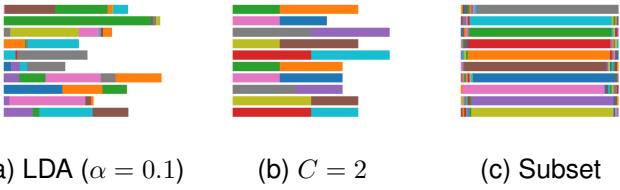


Fig. 3: Data Distribution in Various FL Heterogeneity Scenarios

The characteristic of label skew is that each client samples data only from a fixed subset of the total classes, with other classes entirely absent. For instance, in IoMT scenarios, certain telemedicine devices may only collect specific types of physiological indicators or disease data, lacking coverage of the full spectrum of disease categories. Quantity skew refers to the scenario where the total amount of data varies significantly across clients, while the class distribution may remain similar. For example, specialized hospitals may accumulate a large number of cases for a particular disease, whereas general hospitals may have a broader range of data but with fewer cases for each individual category.

The non-IID characteristics of medical data in IoMT are particularly prominent. Different hospitals may use different medical devices and data standards, leading to variations in measurement methods. The distribution of patient characteristics may be biased due to regional or departmental differences. Specialized hospitals may possess a large volume of cases for a single disease (corresponding to quantity skew), while grassroots clinics or wearable devices may only collect a limited amount of physiological health data [49]. Therefore, conducting experiments with the three types of non-IID partitions mentioned above helps evaluate the adaptability and generalization ability of FedTPF in IoMT scenarios.

Although LDA allocation, label skew, and quantity skew effectively simulate the data heterogeneity between different healthcare institutions and devices, these methods still have certain limitations, particularly when dealing with extreme non-IID (Non-Independent and Identically Distributed) data. In real-world healthcare environments, some specialized hospitals or telemedicine devices may exclusively collect data from specific patient categories over extended periods, entirely missing information from certain disease categories, which is a more extreme case than traditional label skew. Additionally, the imbalance in data quality (e.g., comparing data from large hospitals to that from personal wearable devices)

and the geographical and demographic differences in patient populations further exacerbate the data imbalance. Therefore, we also evaluate the adaptability of FedTPF to these extreme non-IID phenomena. For the LDA distribution, we discuss three cases: $\alpha = 0.1$, $\alpha = 0.3$, and $\alpha = 1$ (where α represents the uniformity of the topic distribution). A smaller α results in a more uneven topic distribution. Specifically, $\alpha = 0.1$ typically represents extreme data heterogeneity). For the label skew distribution, we simulate extreme label skew by assigning only 2 classes to each client. For the quantity skew distribution, we simulate extreme quantity skew by making a single topic account for more than 90% of the client’s data.

Datasets: We evaluate the effectiveness of FedTPF on three datasets.

(1) The LC25000 (Lung and Colon) dataset [50] consists of 25,000 histopathological images with a resolution of 224×224 pixels. It is divided into five categories: colon adenocarcinoma, benign colon tissue, lung adenocarcinoma, lung squamous cell carcinoma, and benign lung tissue, with 5,000 images per class. All images have been preprocessed with standard normalization techniques. (2) The PathMNIST [51] dataset comprises 107,180 histological images of colorectal cancer tissue, each with a resolution of 28×28 pixels. It is split into 89,996 training images, 10,004 validation images, and 7,180 test images. The dataset covers nine tissue types: adipose tissue, background, debris, lymphocytes, mucus, smooth muscle, normal colon mucosa, cancer-associated stroma, and colorectal adenocarcinoma epithelium. All images have been standardized through preprocessing. (3) The CelebA [52] dataset contains 202,599 facial images at a resolution of 178×218 pixels, covering more than 10,000 distinct identities. The images exhibit significant variation in pose, expression, background, and illumination, and have been preprocessed with face alignment and standard normalization.

Benchmarks: We compared the accuracy of FedTPF with six benchmarks: FedAvg [8], FedKF [25], FedDr+ [27], FedFA [53], FedTGP [28], and FedFed [54]. FedAvg, as a canonical aggregation algorithm, serves as a baseline for evaluating the foundational performance of federated learning. FedKF mitigates client model bias through a global-local knowledge fusion strategy and enhances model consistency under heterogeneous data distributions via data-free knowledge distillation. FedDr+ introduces a point-wise regression loss and a feature distillation mechanism to effectively alleviate client drift, improving local model alignment while preserving information from unseen classes, thereby enhancing global model performance in non-i.i.d. settings. FedFA aligns features and calibrates classifiers via functional anchors, breaking the vicious cycle among clients and improving model consistency and cross-client generalization under data heterogeneity. FedTGP enhances prototype separability while preserving semantic meaning by introducing adaptive margin-augmented contrastive learning (ACL) for prototype training. FedFed addresses data heterogeneity by distilling partial features from shared data in a federated manner. These methods are selected to comprehensively evaluate the performance advantage and practical applicability of FedTPF, encompassing diverse perspectives on privacy-preserving modeling, feature sharing,

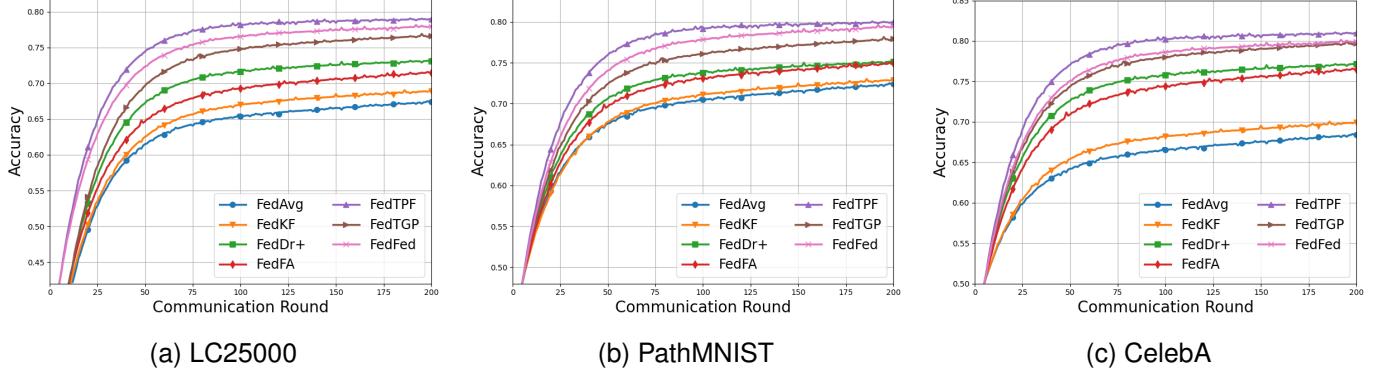


Fig. 4: Accuracy Curves on LC25000, PathMNIST, and CelebA Datasets Under the LDA ($\alpha = 1$) Data Distribution Condition.

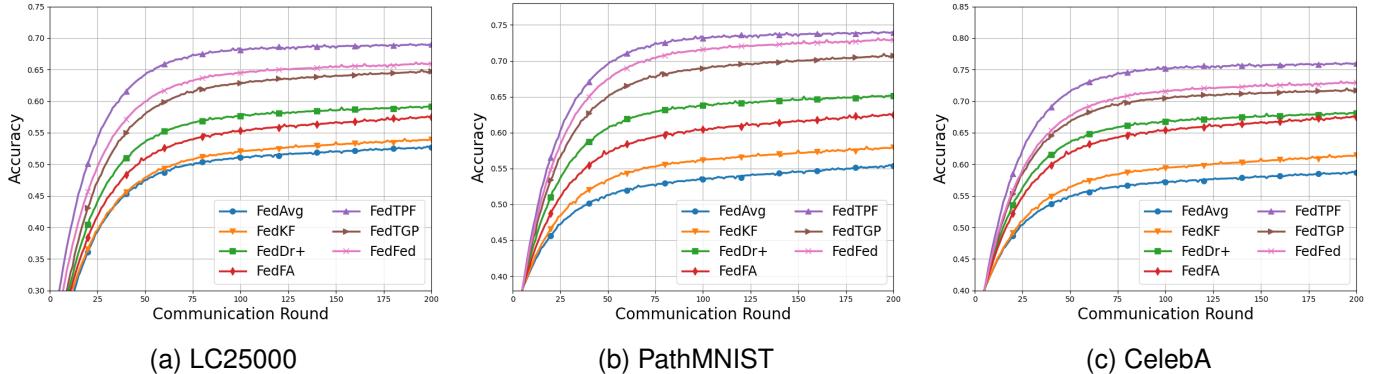


Fig. 5: Accuracy Curves on LC25000, PathMNIST, and CelebA Datasets Under Label-skewed($\#C = 2$) Data Distribution Conditions.

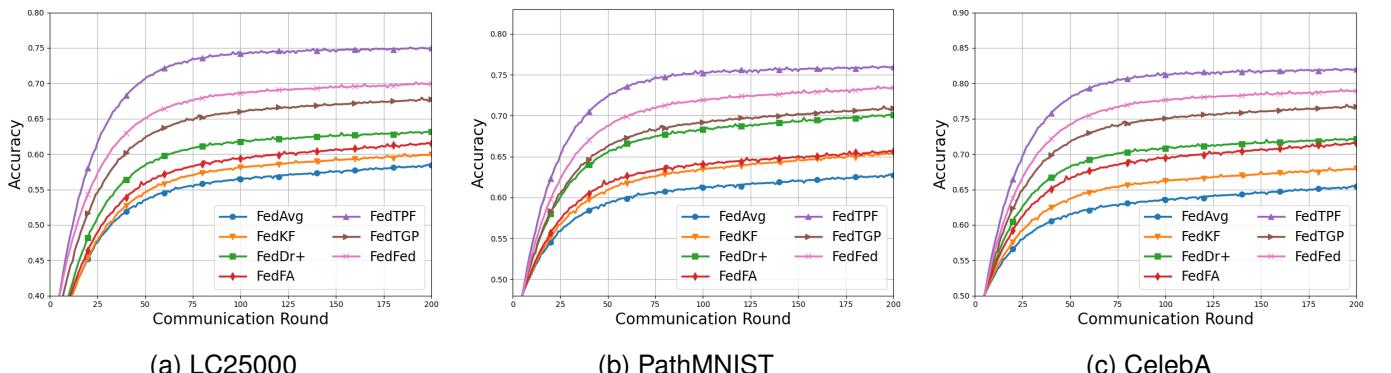


Fig. 6: Accuracy Curves on LC25000, PathMNIST, and CelebA Datasets Under Quantity Skew (Subset) Data Distribution Conditions

and distribution-aware learning mechanisms. We compared FedTPF with other approaches across five different attributes: Privacy Protection, Differential Privacy, Feature Normalization, Data Sharing, and Heterogeneity. The results are presented in Table II.

B. Experiment Results

Accuracy Analysis: To validate the effectiveness of FedTPF, we evaluate its accuracy under three different data heterogeneity settings across three datasets, as shown in Tables III, IV, V. To better visualize the learning process of each

method, we plot the test accuracy curves of various algorithms on different datasets(See Figures 4,5,6 for details). Clearly, FedTPF exhibits outstanding accuracy across all datasets under most data heterogeneity levels.

Client Participation Rate Analysis: To further validate whether FedTPF maintains its advantage under different client participation rates, we considered three values for the client participation rate C : 0.1, 0.4, and 0.8. We calculated the model accuracy under the Dirichlet sampling distribution (LDA $\alpha = 0.1$). The results show that FedTPF consistently delivers the best accuracy across all client participation rates(as shown in

TABLE II: Benchmarks Comparison in Privacy-preserving Attributes

Method	Privacy Protection	Differential Privacy	Feature Normalization	Data Sharing	Heterogeneity
FedAvg	✗ No mechanism	✗	✗	✗	✗
FedKF	Partial protection	✗	✗	✓ Shares distilled logits	✓
FedDr+	Partial protection	✗	✓	✓ Uses feature representations for distillation	✓
FedFA	Personalized protection	✗	✗	✗	✓
FedTGP	✗ No explicit mechanism	✗	✓	✓ Shares global prototype	✓
FedFed	✓ Differential privacy protection	✓	✓	✓ Shares partial features	✓
FedTPF	✓ Differential privacy protection	✓	✓	✓ shares minimal generalizable representation	✓

TABLE III: Prediction Accuracy of Different Dirichlet Sampling (LDA) aAlgorithms on LC25000, PathMNIST, and CelebA Dataset

	LC25000		PathMNIST		CelebA
	$\alpha = 0.1$	$\alpha = 1$	$\alpha = 0.3$	$\alpha = 1$	iid
FedAvg	61.74±0.95	64.65±0.90	67.43±0.72	69.51±0.86	65.72±0.84
FedKF	62.58±0.87	65.99±0.72	68.61±0.86	70.93±0.79	67.92±0.99
FedFA	65.95±0.82	68.43±0.79	70.55±0.65	72.11±0.77	73.93±0.81
FedDr+	67.69±0.77	71.84±0.58	71.90±0.83	73.95±0.81	75.51±0.91
FedTGP	70.47±0.55	74.79±0.50	74.89±0.64	76.48±0.59	77.74±0.96
FedFed	72.87±0.49	76.76±0.32	75.78±0.29	77.34±0.52	78.91±0.79
FedTPF (ours)	73.67±0.22	78.24±0.19	77.34±0.62	79.23±0.69	80.77±0.86

TABLE IV: Prediction Accuracy of Various Label Tilt Algorithms on LC25000, PathMNIST, and CelebA Dataset

	LC25000		PathMNIST		CelebA	
	#C = 2	#C = 3	#C = 2	#C = 3	#C = 2	#C = 3
FedAvg	50.75±0.89	53.98±0.92	52.62±0.59	54.69±0.85	55.94±0.78	58.92±2.01
FedKF	51.67±0.81	56.76±0.87	54.19±0.61	58.55±0.72	58.64±0.68	60.77±2.52
FedFA	54.81±0.75	63.70±0.74	59.27±0.44	62.31±0.66	64.57±0.56	65.34±0.12
FedDr+	57.46±0.66	66.92±0.59	63.89±0.90	66.34±0.79	66.65±0.75	70.62±2.38
FedTGP	62.83±0.46	69.32±0.43	68.83±0.22	70.85±0.31	70.75±0.91	72.34±2.60
FedFed	64.91±0.33	68.57±0.41	71.72±0.78	73.44±0.40	72.26±0.74	74.89±2.43
FedTPF (ours)	68.62±0.29	72.99±0.39	73.35±0.32	76.24±0.21	75.45±1.35	78.97±1.26

Tables VI).

Privacy Verification: We conducted tests on the privacy protection effectiveness of FedTPF to provide an empirical analysis of the privacy-utility trade-off methods. Our goal is to test whether certain attack methods can infer the global shared data. Therefore, we identified the widely used model inversion attack [55] in recent literature to reconstruct the global shared data. The attack effect is reflected in Figure 7. Specifically, x_s is the transparent testing attack generalizable features, x_p is globally shared data, where $x_p = x_s + n$, $n \sim \mathcal{N}(0, \sigma_s^2 \mathbf{I})$. From Figure 7b, we can see that without privacy protection, the generalizable feature x_s still causes leakage. Figure 7c visually illustrates the effect of attacking the protected generalizable feature x_p .

Additionally, we conducted membership inference attack (MIA) [44], [56] experiments, where the attacker aims to infer

TABLE V: Prediction Accuracy of the Quantity Skew Algorithm on LC25000, PathMNIST, and CelebA Dataset

	LC25000	PathMNIST	CelebA
FedAvg	55.64±0.90	60.59±0.86	64.61±0.88
FedKF	58.06±0.87	62.42±0.81	66.29±0.76
FedFA	58.87±0.64	63.89±0.72	67.18±0.53
FedDr+	60.55±0.35	67.27±0.63	70.13±0.74
FedTGP	65.97±0.36	69.37±0.49	74.93±0.67
FedFed	68.85±0.30	72.29±0.32	77.80±0.72
FedTPF (ours)	74.59±0.20	75.95±0.21	81.50±0.36

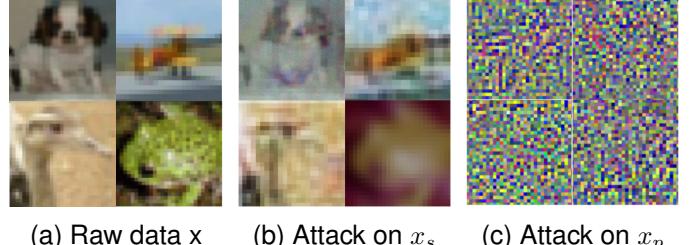


Fig. 7: Model Inversion Attack Results

whether a data sample is part of the target model's dataset. we treated the server-side global model as the target model and trained a shadow model using the globally shared data. The input to the attack model consists of the top- k vectors output by the shadow model, and the attack model produces a binary result indicating whether the sample belongs to the training set. We also compared MIA under different noise levels with experiments that shared fully differentially private (DP) feature data. As shown in Figure 8, we perform membership inference attacks on the global model every 10 communication rounds. We observe that, under the same privacy budget, x_p (shared data) requires less noise intensity, allowing FedTPF to achieve better performance while maintaining privacy protection. The recall curve for the shared partial feature data reaches approximately 54% when the attack model approaches convergence—close to the level of random guessing.

Under the same DP protection level, sharing raw data results in a higher MIA recall rate (indicating that the attack model is more effective at identifying whether a sample belongs to the original training set) compared to sharing partial data. This demonstrates that, under the same privacy budget, sharing partial feature data requires a lower noise level (α_2), enabling models sharing partial feature data to outperform those sharing all feature data in terms of performance.

Ablation Study: We investigate the impact of the prototype mechanism on the efficiency of the feature distillation model by testing feature distillation models with and without the prototype mechanism. As shown in Figure 9, the results demonstrate the effectiveness of the prototype mechanism in enhancing model accuracy.

To validate the robustness of FedTPF under different noise levels, we also considered the application of Laplace noise to the globally shared data when applying differential privacy (DP). Table VII reports the results of model accuracy under various noise levels. As shown in Figure 10, the results in the table demonstrate the robustness of FedTPF across different noise settings.

TABLE VI: Prediction Accuracy of Different Customer Engagement Rates on LC25000, PathMNIST, and CelebA Dataset

	LC25000			PathMNIST			CelebA		
	$C = 0.1$	$C = 0.4$	$C = 0.8$	$C = 0.1$	$C = 0.4$	$C = 0.8$	$C = 0.1$	$C = 0.4$	$C = 0.8$
FedAvg	54.30	58.59	60.56	57.71	60.39	62.18	60.53	63.87	65.76
FedKF	56.49	60.79	62.33	58.52	61.45	63.42	63.85	66.18	68.13
FedFA	58.55	60.71	65.68	60.59	63.58	66.68	68.64	70.09	74.03
FedDr+	59.63	62.48	69.08	61.91	64.64	68.55	71.66	76.15	76.99
FedTGP	61.57	63.82	68.36	63.67	66.93	70.12	72.57	77.49	78.82
FedFed	61.67	65.57	70.77	64.87	67.86	72.14	72.98	77.87	79.19
FedTPF (ours)	65.71	67.88	71.51	67.65	69.98	74.85	76.32	80.94	81.67

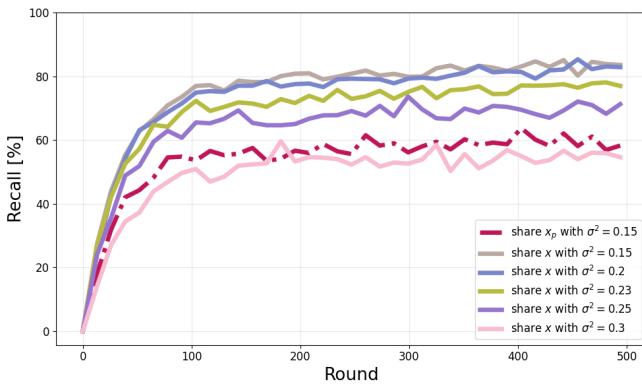


Fig. 8: Recall Curve Under Different Noise Levels

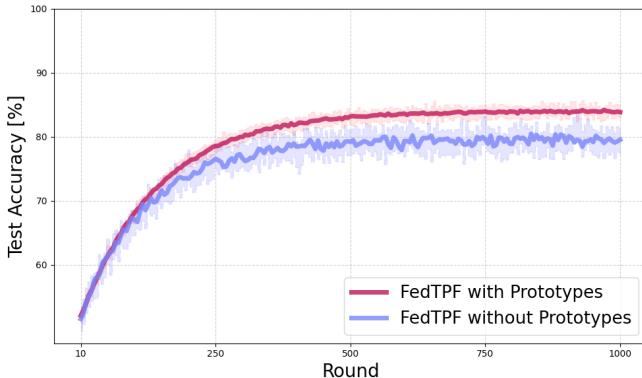


Fig. 9: Accuracy Curve of the Distillation Model With and Without Prototypes

Meanwhile, we also evaluate the privacy protection effectiveness of FedTPF under different noise levels, as shown in Table VIII. The results demonstrate that FedTPF is capable of resisting inference attacks even at relatively low noise intensities. Moreover, as the noise intensity increases, the privacy protection is further enhanced (i.e., a decrease in recall indicates a lower success rate for attackers in correctly identifying sensitive information).

TABLE VII: Test Accuracy on Different Noise with **FedTPF**

Noise Type	FedFA	FedTGP	FedTPF
Gaussian Noise	70.84	75.66	79.17
Laplacian Noise	71.30	75.97	81.32

TABLE VIII: Recall Rate of Inference Attacks on **FedTPF**

Noise Type—Noise intensity	$\sigma_s^2 = 0.15$	$\sigma_s^2 = 0.20$	$\sigma_s^2 = 0.25$
Gaussian Noise	65.42%	60.29%	57.85%
Laplacian Noise	62.58%	58.38%	56.47%

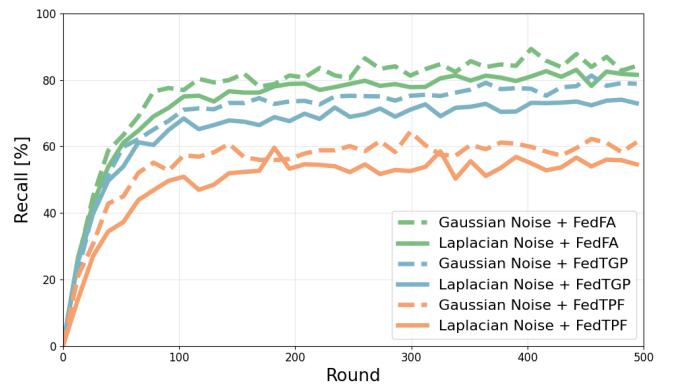


Fig. 10: Experiment Results with Different Noise on LC25000 Dataset

To explore the impact of different generators on the robustness of the feature distillation model, we considered two types of generators for reconstructing images, which serve as the basis for extracting generalizable features in the distillation model. The first is the classic ResNet Generator, commonly used in many works [57], [58]. The second is the VAE-WGAN-GP [38], which uses a VAE [41] network as the generator in the GAN adversarial generation [59] framework, leveraging the VAE's excellent feature extraction capabilities to capture the latent space distribution of complex input data. The results, as shown in Figure 11, validate the effectiveness of FedTPF with different generation models. According to the results in Figure 11, VAE-WGAN-GP outperforms the ResNet Generator in terms of performance. All generators mentioned in this paper refer to VAE-WGAN-GP.

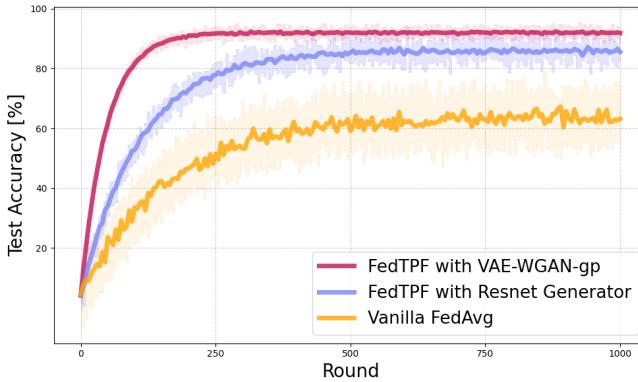


Fig. 11: Convergence Comparison: VAE-WGAN-GP (ResNet) vs. FedAvg on Three Datasets

V. CONCLUSION

In this paper, we propose FedTPF, a federated learning framework designed to mitigate client data heterogeneity in the Internet of Medical Things (IoMT). FedTPF leverages a distilled model to filter shared client data, combining privacy-preserving mechanisms with the extraction of minimal yet generalizable information. We fully consider the privacy-accuracy trade-off, experimental results show that FedTPF outperforms previous advanced methods, improving test accuracy by 3.7% while ensuring that medical privacy information remains protected. This performance improvement holds significant clinical implications. In medical imaging diagnostics, a 3.7% increase in accuracy translates to 37 fewer misdiagnosed cases per 1,000 lung cancer screenings, thereby reducing the risk of unnecessary biopsies or delayed treatments. For diabetic retinopathy detection, this improvement can boost early detection rates by 12–15%, providing patients with a critical window for timely intervention. More importantly, our method achieves this gain while preserving privacy, effectively addressing the long-standing trade-off between diagnostic accuracy and data confidentiality in conventional approaches.

The FedTPF framework points to a significant research direction for the development of the Internet of Medical Things (IoMT): deeply integrating edge computing with federated learning to enable real-time and adaptive model updates on IoMT devices will be a key breakthrough. As medical data standards become increasingly unified, FedTPF is expected to serve as a foundational framework for cross-institution and cross-regional collaboration in medical AI. It holds the potential to promote the establishment of a global medical knowledge-sharing network while ensuring that data sovereignty of all participating entities remains protected.

In future work, we plan to replace the foundational model in the FL framework with more complex large language models to explore data heterogeneity challenges in IoMT-based federated learning.

REFERENCES

- [1] J. Xu, B. S. Glicksberg, C. Su, P. Walker, J. Bian, and F. Wang, “Federated learning for healthcare informatics,” *Journal of healthcare informatics research*, vol. 5, pp. 1–19, 2021.
- [2] J.-P. A. Yaacoub, M. Noura, H. N. Noura, O. Salman, E. Yaacoub, R. Couturier, and A. Chehab, “Securing internet of medical things systems: Limitations, issues and recommendations,” *Future Generation Computer Systems*, vol. 105, pp. 581–606, 2020.
- [3] X. Zhou, W. Huang, W. Liang, Z. Yan, J. Ma, Y. Pan, and K. I.-K. Wang, “Federated distillation and blockchain empowered secure knowledge sharing for internet of medical things,” *Information Sciences*, vol. 662, p. 120217, 2024.
- [4] R. Miotto, F. Wang, S. Wang, X. Jiang, and J. T. Dudley, “Deep learning for healthcare: review, opportunities and challenges,” *Briefings in bioinformatics*, vol. 19, no. 6, pp. 1236–1246, 2018.
- [5] F. Wang and A. Preininger, “Ai in health: state of the art, challenges, and future directions,” *Yearbook of medical informatics*, vol. 28, no. 01, pp. 016–026, 2019.
- [6] Y. Xu, G. Xu, C. Ma, and Z. An, “An advancing temporal convolutional network for 5g latency services via automatic modulation recognition,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 69, no. 6, pp. 3002–3006, 2022.
- [7] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [8] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, “Communication-efficient learning of deep networks from decentralized data,” in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.
- [9] X. Zhou, W. Liang, I. Kevin, K. Wang, K. Yada, L. T. Yang, J. Ma, and Q. Jin, “Decentralized federated graph learning with lightweight zero trust architecture for next-generation networking security,” *IEEE Journal on Selected Areas in Communications*, 2025.
- [10] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, “Tackling the objective inconsistency problem in heterogeneous federated optimization,” *Advances in neural information processing systems*, vol. 33, pp. 7611–7623, 2020.
- [11] J. Wang, Z. Charles, Z. Xu, G. Joshi, H. B. McMahan, M. Al-Shedivat, G. Andrew, S. Avestimehr, K. Daly, D. Data *et al.*, “A field guide to federated optimization,” *arXiv preprint arXiv:2107.06917*, 2021.
- [12] Z. Huang, Y. Wu, N. Tempini, H. Lin, and H. Yin, “An energy-efficient and trustworthy unsupervised anomaly detection framework (eatu) for iiot,” *ACM Transactions on Sensor Networks*, vol. 18, no. 4, pp. 1–18, 2022.
- [13] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, “Scaffold: Stochastic controlled averaging for federated learning,” in *International conference on machine learning*. PMLR, 2020, pp. 5132–5143.
- [14] M. Luo, F. Chen, D. Hu, Y. Zhang, J. Liang, and J. Feng, “No fear of heterogeneity: Classifier calibration for federated learning with non-iid data,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 5972–5984, 2021.
- [15] W. Hao, M. El-Khamy, J. Lee, J. Zhang, K. J. Liang, C. Chen, and L. C. Duke, “Towards fair federated learning with zero-shot data augmentation,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 3310–3319.
- [16] Z. Huang and Y. Wu, “A survey on explainable anomaly detection for industrial internet of things,” in *2022 IEEE Conference on Dependable and Secure Computing (DSC)*. IEEE, 2022, pp. 1–9.
- [17] L. Qu, Y. Zhou, P. P. Liang, Y. Xia, F. Wang, E. Adeli, L. Fei-Fei, and D. Rubin, “Rethinking architecture design for tackling data heterogeneity in federated learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 10061–10071.
- [18] M. Mendieta, T. Yang, P. Wang, M. Lee, Z. Ding, and C. Chen, “Local learning matters: Rethinking data heterogeneity in federated learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8397–8406.
- [19] A. Mora, A. Bujari, and P. Bellavista, “Enhancing generalization in federated learning with heterogeneous data: A comparative literature review,” *Future Generation Computer Systems*, 2024.
- [20] M. Nasr, R. Shokri, and A. Houmansadr, “Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning,” in *2019 IEEE symposium on security and privacy (SP)*. IEEE, 2019, pp. 739–753.
- [21] H. Chen and H. Vikalo, “Federated learning in non-iid settings aided by differentially private synthetic data,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2023, pp. 5027–5036.
- [22] R. Shwartz-Ziv and N. Tishby, “Opening the black box of deep neural networks via information,” *arXiv preprint arXiv:1703.00810*, 2017.

- [23] Q. Li, Z. Wen, Z. Wu, S. Hu, N. Wang, Y. Li, X. Liu, and B. He, "A survey on federated learning systems: Vision, hype and reality for data privacy and protection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 4, pp. 3347–3366, 2021.
- [24] X. Sun, Z. Yuan, X. Kong, L. Xue, L. He, and Y. Lin, "Communication-efficient and privacy-preserving aggregation in federated learning with adaptability," *IEEE Internet of Things Journal*, 2024.
- [25] X. Zhou, X. Lei, C. Yang, Y. Shi, X. Zhang, and J. Shi, "Handling data heterogeneity for iot devices in federated learning: A knowledge fusion approach," *IEEE Internet of Things Journal*, 2023.
- [26] X. Zhou, Q. Yang, X. Zheng, W. Liang, K. I.-K. Wang, J. Ma, Y. Pan, and Q. Jin, "Personalized federated learning with model-contrastive learning for multi-modal user modeling in human-centric metaverse," *IEEE Journal on Selected Areas in Communications*, vol. 42, no. 4, pp. 817–831, 2024.
- [27] S. Kim, M. Jeong, S. Kim, S. Cho, S. Ahn, and S.-Y. Yun, "Feddr+: Stabilizing dot-regression with global feature distillation for federated learning," *arXiv preprint arXiv:2406.02355*, 2024.
- [28] J. Zhang, Y. Liu, Y. Hua, and J. Cao, "Fedtgp: Trainable global prototypes with adaptive-margin-enhanced contrastive learning for data and model heterogeneity in federated learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 15, 2024, pp. 16768–16776.
- [29] Y. Tan, G. Long, L. Liu, T. Zhou, Q. Lu, J. Jiang, and C. Zhang, "Fedproto: Federated prototype learning across heterogeneous clients," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 8, 2022, pp. 8432–8440.
- [30] D. A. E. Acar, Y. Zhao, R. M. Navarro, M. Mattina, P. N. Whatmough, and V. Saligrama, "Federated learning based on dynamic regularization," *arXiv preprint arXiv:2111.04263*, 2021.
- [31] Z. He, L. Wang, and Z. Cai, "Clustered federated learning with adaptive local differential privacy on heterogeneous iot data," *IEEE Internet of Things Journal*, 2023.
- [32] J. Zhang, Z. Li, B. Li, J. Xu, S. Wu, S. Ding, and C. Wu, "Federated learning with label distribution skew via logits calibration," in *International Conference on Machine Learning*. PMLR, 2022, pp. 26311–26329.
- [33] X. Zhou, J. Wu, W. Liang, K. I.-K. Wang, Z. Yan, L. T. Yang, and Q. Jin, "Reconstructed graph neural network with knowledge distillation for lightweight anomaly detection," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 9, pp. 11817–11828, 2024.
- [34] G. Barthe, M. Gaboardi, B. Grégoire, J. Hsu, and P.-Y. Strub, "Proving differential privacy via probabilistic couplings," in *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science*, 2016, pp. 749–758.
- [35] N. Tishby and N. Zaslavsky, "Deep learning and the information bottleneck principle," in *2015 ieee information theory workshop (itw)*. Ieee, 2015, pp. 1–5.
- [36] K. Akuzawa, Y. Iwasawa, and Y. Matsuo, "Information-theoretic regularization for learning global features by sequential vae," *Machine Learning*, vol. 110, pp. 2239–2266, 2021.
- [37] Z. Tao, S. Huang, and G. Wang, "Prototypes sampling mechanism for class incremental learning," *IEEE Access*, 2023.
- [38] K. Yonekura, Y. Tomori, and K. Suzuki, "Airfoil shape generation and feature extraction using the conditional vae-wgan-gp," *AI*, vol. 5, no. 4, pp. 2092–2103, 2024.
- [39] X. Zhou, X. Zheng, T. Shu, W. Liang, K. I.-K. Wang, L. Qi, S. Shimizu, and Q. Jin, "Information theoretic learning-enhanced dual-generative adversarial networks with causal representation for robust ood generalization," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.
- [40] M. Figurnov, S. Mohamed, and A. Mnih, "Implicit reparameterization gradients," *Advances in neural information processing systems*, vol. 31, 2018.
- [41] D. P. Kingma, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.
- [42] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," *Advances in neural information processing systems*, vol. 30, 2017.
- [43] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," *arXiv preprint arXiv:1611.03530*, 2016.
- [44] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE symposium on security and privacy (SP)*. IEEE, 2017, pp. 3–18.
- [45] W. Zhang, S. Tople, and O. Ohremenko, "Leakage of dataset properties in {Multi-Party} machine learning," in *30th USENIX security symposium (USENIX Security 21)*, 2021, pp. 2687–2704.
- [46] C. Dwork, A. Roth *et al.*, "The algorithmic foundations of differential privacy," *Foundations and Trends® in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014.
- [47] A. Blum, K. Ligett, and A. Roth, "A learning theory approach to noninteractive database privacy," *Journal of the ACM (JACM)*, vol. 60, no. 2, pp. 1–25, 2013.
- [48] T.-M. H. Hsu, H. Qi, and M. Brown, "Measuring the effects of non-identical data distribution for federated visual classification," *arXiv preprint arXiv:1909.06335*, 2019.
- [49] X. Zhou, X. Ye, I. Kevin, K. Wang, W. Liang, N. K. C. Nair, S. Shimizu, Z. Yan, and Q. Jin, "Hierarchical federated learning with social context clustering-based participant selection for internet of medical things applications," *IEEE Transactions on Computational Social Systems*, vol. 10, no. 4, pp. 1742–1751, 2023.
- [50] A. A. Borkowski, M. M. Bui, L. B. Thomas, C. P. Wilson, L. A. DeLand, and S. M. Mastorides, "Lung and colon cancer histopathological image dataset (lc25000)," *arXiv preprint arXiv:1912.12142*, 2019.
- [51] J. Yang, R. Shi, and B. Ni, "Medmnist classification decathlon: A lightweight automl benchmark for medical image analysis," in *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*. IEEE, 2021, pp. 191–195.
- [52] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 3730–3738.
- [53] T. Zhou, J. Zhang, and D. H. Tsang, "Fedfa: Federated learning with feature anchors to align features and classifiers for heterogeneous data," *IEEE Transactions on Mobile Computing*, 2023.
- [54] Z. Yang, Y. Zhang, Y. Zheng, X. Tian, H. Peng, T. Liu, and B. Han, "Fedfed: Feature distillation against data heterogeneity in federated learning," *Advances in Neural Information Processing Systems*, vol. 36, 2024.
- [55] Z. He, T. Zhang, and R. B. Lee, "Model inversion attacks against collaborative inference," in *Proceedings of the 35th Annual Computer Security Applications Conference*, 2019, pp. 148–162.
- [56] H. Yan, S. Li, Y. Wang, Y. Zhang, K. Sharif, H. Hu, and Y. Li, "Membership inference attacks against deep learning models via logits distribution," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 5, pp. 3799–3808, 2022.
- [57] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.
- [58] O. Poursaeed, I. Katsman, B. Gao, and S. Belongie, "Generative adversarial perturbations," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4422–4431.
- [59] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.