Personalized Federated Neural Architecture Search for Edge-Based Monocular Depth Estimation

1st Zhilong He
School of Artificial Intelligence
Hebei University of Technology
Tianjin, China
202422802049@stu.hebut.edu.cn

2nd Haoyu Zhang*
School of Information Science and Technology
Hangzhou Normal University
Hangzhou, China
haoyu.zhang@hznu.edu.cn

3rd Qiqi Liu*
Westlake University
Westlake Institute for Advanced Study
Hangzhou, China
qiqi6770304@gmail.com

4th Xinjie Wang
School of Information
Science and Technology
Hangzhou Normal University
Hangzhou, China
xinjie.wang@hznu.edu.cn

5th Junhua Gu
School of Artificial Intelligence
Hebei University of Technology
Tianjin, China
jhgu@hebut.edu.cn

6th Ruyu Liu
School of Information
Science and Technology
Hangzhou Normal University
Hangzhou, China
ruyu.liu@ieee.org

***Abstract*—Monocular depth estimation (MDE) is a key component of 3D reconstruction and plays an essential role in applications such as autonomous driving, augmented reality, and robotic navigation. In practical scenarios, MDE models are often required to be deployed on edge devices to support real-time inference. However, deploying MDE models on edge devices faces two major challenges. First, data collected by different devices exhibit significant heterogeneity, while computational and storage resources vary widely across clients, making a unified fixed architecture difficult to meet diverse deployment requirements. Second, existing high-accuracy MDE models usually involve heavy computational costs, which limits their applicability on resource-constrained edge devices. To address these challenges, we propose PTF-EMDE, a personalized federated learning (PFL) framework based on bi-objective evolutionary neural architecture search. By designing a lightweight encoder search space and employing a bi-objective optimization strategy, PTF-EMDE is able to significantly reduce computational complexity while maintaining high estimation accuracy, thereby automatically discovering efficient and client-adaptive model architectures for heterogeneous edge devices. Experimental results on the KITTI dataset demonstrate that PTF-EMDE achieves higher prediction accuracy and lower computational cost than baseline methods, providing an efficient and scalable solution for edge deployment of MDE in privacy-sensitive scenarios.**

***Index Terms*—Monocular Depth Estimation, Federated Learning, Neural Architecture Search, Multi-objective Evolutionary Algorithms.**

I. Introduction

MDE is a key task in computer vision that predicts pixel-wise depth from a single RGB image. It supports various applications such as 3D reconstruction, scene understanding, and autonomous driving [1]. Recent advances in deep neural networks have significantly improved the accuracy and robustness of MDE under real-world conditions. To enable real-time perception, MDE models are increasingly deployed on edge devices, including mobile platforms, in-vehicle systems, and medical robots, highlighting the need for models that are both accurate and lightweight for efficient on-device inference.

However, deploying MDE models on edge devices presents two major challenges. First, edge device application environments are highly diverse, leading to significant data heterogeneity between clients in terms of scene structure, lighting conditions, viewpoint distribution, and camera parameters. In addition, edge devices vary widely in computational power and memory capacity, making a single fixed architecture insufficient to meet all deployment needs. Second, most edge devices are resource-constrained, while state-of-the-art MDE models often involve high computational complexity, hindering real-time inference.

To address the first challenge, recent studies have used PFL frameworks, such as FedRep [2] and FedALA [3], to mitigate data heterogeneity between clients. However, these PFL frameworks still require that all clients share the same model architecture, while in the context of MDE, it is highly desired to design personalized models for each client. This is due to the complex interaction between scene semantics and depth cues, which requires a task-specific architectural adaptation. Moreover, achieving both high accuracy and computational efficiency still relies heavily on manual architecture design, which is time-consuming and resource-intensive. These limitations highlight the need for automated methods to discover lightweight yet high-performing architectures tailored to individual clients.

Neural architecture search (NAS) provides an effective approach for automatically discovering high-performing network architectures, and has recently been introduced into federated learning (FL) to enable personalized architecture optimiza-

*Corresponding authors: Qiqi Liu and Haoyu Zhang

This work was supported in part by the National Natural Science Foundation of China under Grant No. 62302147, 62306097, 62306114, and in part by Selected Funding Program for Zhejiang Province Postdoctoral Research Projects.

tion [4]. For example, SPIDER [5] performs client-specific adaptation by alternating global and local optimization steps, while PerFedRLNAS [6] employs reinforcement learning to search for architectures that balance accuracy and efficiency. In practical applications, personalized federated NAS has been applied to tasks such as MRI image segmentation [7], [8], showing improved performance across heterogeneous clients.

Although personalized federated NAS has shown promising results in various computer vision tasks, directly applying existing methods to the design of personalized MDE models for edge deployment presents several key challenges. First, deploying MDE models on edge devices remains difficult due to the wide variability and limited capacity of computational resources. It is essential to jointly consider both depth estimation accuracy and computational efficiency during model design. However, these two objectives often conflict, making it challenging to manually balance performance and resource usage. A principled approach is therefore required to explore the trade-offs between these two objectives and to automatically generate architectures that are tailored to each client's hardware constraints. Second, MDE models typically consist of a large number of parameters. In federated settings, frequent transmission of model updates between clients and the central server leads to prohibitively high communication costs, which severely limits practical deployment.

To address these aforementioned challenges, we propose a bi-objective personalized federated NAS framework that jointly optimizes depth estimation accuracy and computational efficiency. By incorporating both objectives into the search process, the proposed method enables automatic discovery of resource-adaptive architectures that are tailored to the heterogeneous constraints of edge devices. To make the architecture search more efficient, we design a lightweight supernet as the underlying search space. This supernet significantly reduces computational complexity during training and inference, while maintaining competitive estimation accuracy. As a result, the proposed framework supports efficient and scalable deployment of personalized MDE models in resource-constrained federated environments. The main contributions of this work are summarized as follows:

- We propose PTF-EMDE, a PFL framework based on multi-objective evolutionary optimization. It enables the automatic search for lightweight and high-performing model architectures tailored to heterogeneous edge devices.
- We design a task-specific lightweight encoder search space within the supernet, optimized for edge deployment. This search space effectively reduces both the computational cost and communication overhead of federated NAS, while preserving sufficient representational capacity for accurate depth estimation.
- We demonstrate through extensive experiments on the KITTI benchmark that the proposed method outperforms the comparison methods in terms of estimation accuracy and computational efficiency under both independent and identically distributed (IID) and non-independent and identically distributed (non-IID) data partition settings.

II. Related Work

A. Monocular Depth Estimation

MDE aims to predict the per-pixel depth values from a single RGB image by learning a mapping from two-dimensional visual input to the three-dimensional structure of a scene. This task plays a critical role in various applications, including autonomous driving, robotic perception, and medical imaging. With the rise of deep learning, Eigen et al. [9] first introduced convolutional neural networks to directly estimate depth from single images. Laina et al. [10] enhanced prediction accuracy using multi-scale residual modules. NeWCRFs [11] incorporated a Transformer-based encoder to improve global context modeling, while MambaDepth [12] leveraged a state-space model to boost computational efficiency without sacrificing accuracy. Most state-of-the-art MDE models adopt an encoder-decoder architecture, where the encoder is responsible for extracting high-level semantic and spatial features. The encoder plays a crucial role in determining model performance. However, most existing encoders are manually designed and often adapted from image classification backbones. Since classification networks emphasize global semantic abstraction, they may fail to capture the fine-grained geometric cues required for accurate depth estimation, leading to feature mismatches in the encoder design.

B. Federated Neural Architecture Search

MDE often involves large-scale distributed data collected from diverse sources, such as autonomous vehicles, sensors, and medical devices. Directly uploading such data to a centralized server for model training poses significant privacy and security risks. FL addresses this issue by enabling clients to train models locally on their private data, while only sharing model parameters or gradients for global aggregation [13]. Recent studies, such as FedSCDepth [14] and BOFedSCDepth [15], have extended FL to the MDE task, allowing collaborative training without compromising data privacy. However, most existing federated MDE methods focus on learning a single global model, which may not perform well across clients with heterogeneous data distributions. PFL has emerged as a promising direction to address this limitation by tailoring models to each client's local data. This is particularly beneficial for edge devices with constrained resources and diverse application scenarios. Hence, we propose a bi-objective evolutionary NAS framework that automatically searches for personalized architectures under a federated setting, optimizing both depth estimation accuracy and computational efficiency for heterogeneous edge clients.

III. Proposed Method

In this section, we present a comprehensive overview of the proposed PTF-EMDE framework for personalized federated MDE, as illustrated in Fig. 1. We begin by formally defining the problem setting of federated MDE. Next, we introduce a lightweight encoder search space specifically designed for

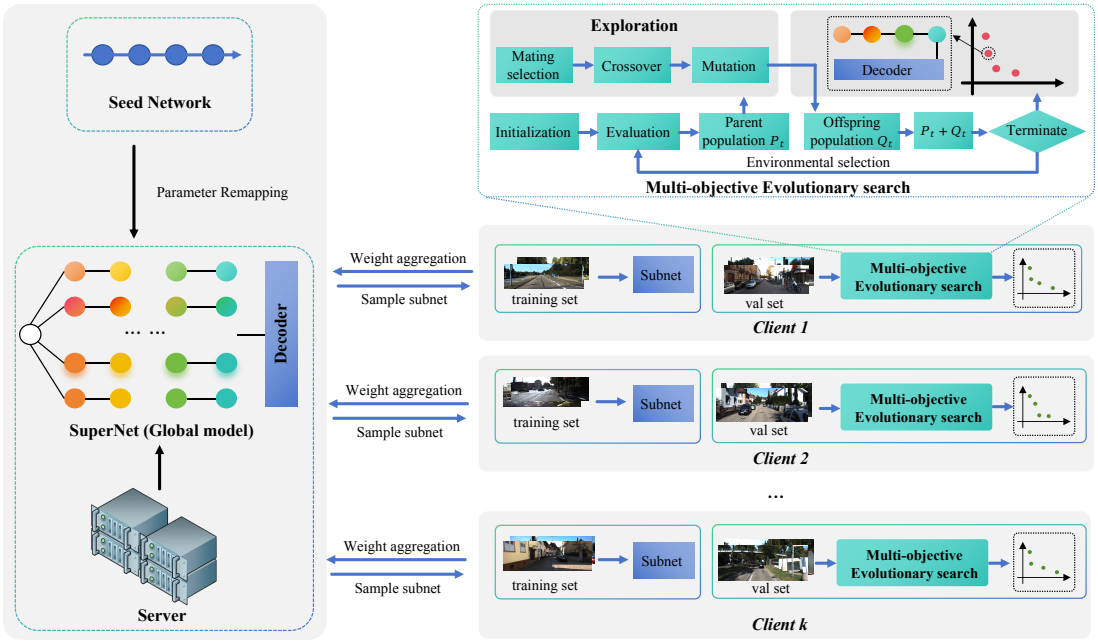


Fig. 1. Overview of the proposed PTF-EMDE framework. During federated supernet training, the server samples sub-networks from the supernet and sends them to clients for local optimization. The locally updated parameters are then uploaded and aggregated to update the supernet. After training converges, the complete supernet is broadcast to all clients only once. Based on this supernet, clients conduct local evolutionary search to derive personalized MDE models, which are subsequently employed for full federated training.

deployment on resource-constrained edge devices. Finally, we describe the evolutionary bi-objective search process, which enables heterogeneous clients to discover personalized MDE architectures that meet their individual requirements while preserving data privacy.

A. Problem Definition

This work addresses the challenge of deploying MDE models in resource-constrained edge environments under FL settings. As a dense per-pixel regression task, MDE is sensitive to both feature representation and spatial resolution. Existing MDE models are typically encoder-heavy with large parameter sizes and high computational cost, limiting their deployment on edge devices. Our goal is to automatically discover personalized MDE architectures for heterogeneous clients that achieve high prediction accuracy and low computational complexity while preserving data privacy. Since high prediction accuracy and low computational complexity are conflicting objectives, we formulate the model design problem as a bi-objective optimization task.

We consider a FL system with K heterogeneous clients, each holding a private dataset $\mathcal{D}_k$. The task is to search for a neural architecture $\mathcal{A}_k \in \mathcal{S}$ for each client that adapts to its data distribution and resource constraints, Absolute Relative Error (Abs Rel) $f_{1,k}(\mathcal{A}_k)$ and simultaneously minimizing computational cost $f_{2,k}(\mathcal{A}_k)$, i.e., $\min_{\mathcal{A}_k \in \mathcal{S}}(f_{1,k}(\mathcal{A}_k), f_{2,k}(\mathcal{A}_k))$, where $\mathcal{S}$ denotes the predefined architecture search space.

To optimize this bi-objective problem, we adopt a multi-objective optimization strategy that explores the Pareto frontier, enabling client-specific architectures that balance accuracy and efficiency for practical deployment on heterogeneous edge devices.

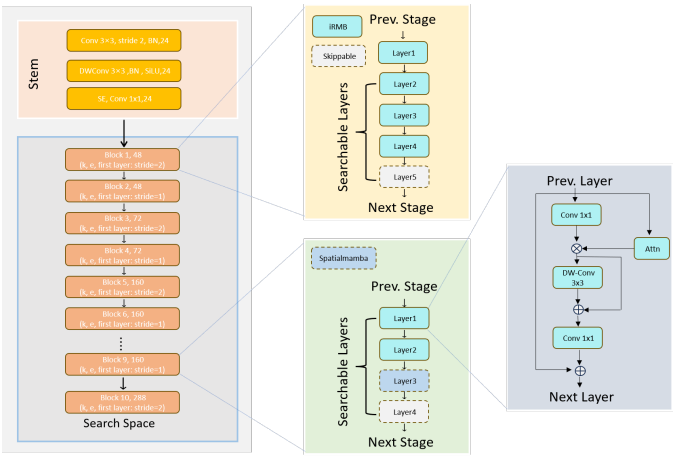
B. Encoder Search Space Design

To support efficient and accurate depth estimation under federated settings, we design a lightweight encoder search space that balances feature representation capacity with computational efficiency. The search space is constructed based on the core building blocks of the EMO model [16]. An overview of the encoder search space is illustrated in Fig. 2.

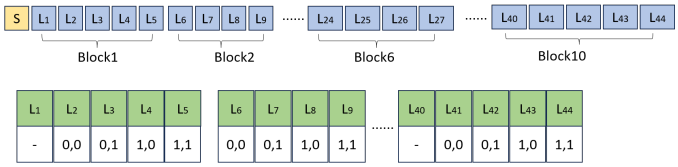
Specifically, the encoder search space is defined as a supernetwork composed of ten sequential choice blocks, denoted as Block 1 through Block 10. These blocks are categorized into two types: downsampling blocks and non-downsampling blocks, in accordance with standard architectural design principles of deep convolutional networks.

Blocks 1, 3, 5, and 10 serve as downsampling stages and each contains five branches. Among them, Layer 1 is a fixed downsampling operation that reduces spatial resolution and adjusts channel dimensions. This layer remains active in all instantiated architectures, while the remaining four branches are candidate operations included in the architecture search process. The remaining blocks are non-downsampling blocks, which do not involve changes in spatial resolution. Each of these blocks includes four fully searchable branches without any fixed operations, thereby allowing greater flexibility in feature extraction under varying computational constraints.

To accommodate the specific requirements of MDE in FL, the encoder search space is designed to balance archi-



(a) Search space



(b) Encoding

Fig. 2. Overview of the encoder search space and its encoding scheme. (a) The encoder search space is extended from an EMO-style architecture and consists of ten stages. At each stage, four extended candidate modules are considered, including an iRMB block with kernel sizes $\{3\times3, 5\times5, 7\times7\}$ and expansion ratios $\{3, 4, 6\}$, a skip-connection block, and a Spatial-Mamba block. (b) Candidate architectures are encoded as binary sequences. For each stage, the four candidate blocks are represented using a two-bit binary code, and a ten-stage sub-network is constructed using a 20-bit encoding. This encoding scheme enables efficient sampling and evolutionary search in federated settings.

tectural expressiveness with computational efficiency. As a dense regression task, MDE requires both fine-grained local feature extraction and effective modeling of long-range spatial dependencies. To support these complementary objectives, the candidate branch configurations are adapted according to the depth of each encoder block.

In Blocks 1 through 6, the four candidate branches include three improved residual mobile bottleneck (iRMB) blocks with kernel sizes $\{3\times3, 5\times5, 7\times7\}$ and expansion ratios $\{3, 4, 6\}$ using depthwise separable convolutions. These variants allow flexible adjustment of receptive fields and channel capacity with minimal computational cost. The fourth branch is an Identity operation, which enables dynamic control of network depth and facilitates computational savings when necessary. Blocks 7 to 9 are designed to enhance global feature modeling. Each block includes two iRMB branches with distinct configurations, one Spatial-Mamba [17] branch that efficiently captures long-range spatial context in 2D feature maps, and one Identity branch. This design enhances global context modeling in deeper layers while preserving parameter efficiency. In Block 10, all four branches are implemented as iRMB blocks with varying kernel sizes and expansion ratios. This configuration strengthens high-level feature representation before the decoder stage. Overall, the search space is tailored to the unique characteristics of MDE by promoting efficient local feature extraction in shallow layers and adaptive global context modeling in deeper layers. This hierarchical design enables each federated client to discover personalized encoder architectures that align with both their computational budgets and the spatial complexity of their local data.

During the search process, sub-networks are sampled from the super-network using a 20-dimensional binary encoding composed of $\{0, 1\}$. Each pair of bits specifies the selection of one candidate branch from a corresponding choice block. The sampled sub-networks are then evaluated in terms of both depth estimation accuracy and computational complexity. Fig. 3 illustrates the instantiation process of a complete architecture from a given binary encoding.

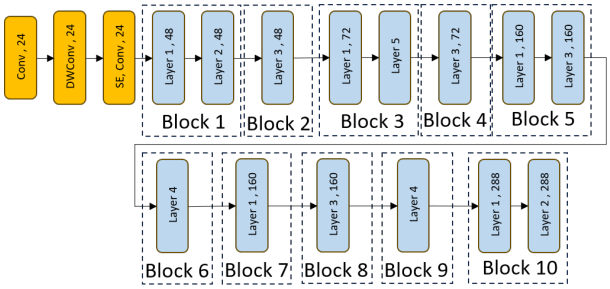


Fig. 3. Illustration of a candidate individual network decoded from the encoding [0,0,1,0,1,1,1,0,0,1,1,1,0,0,1,0,1,1,0,0]. Each pair of elements corresponds to a specific layer in the network and determines which one of the four candidate branches is selected for that layer.

C. PTF-EMDE Pipeline

The overall pipeline of the proposed PTF-EMDE framework is illustrated in the pseudocode of Algorithm 1 and the flowchart shown in Fig. 1. The process begins with adopting the EMO model pre-trained on ImageNet as the seed network. By optionally varying its kernel sizes, expansion ratios, and module types, a lightweight encoder architecture search space is constructed in the form of a super-network.

During the supernet initialization phase (Line 1), all candidate iRMB branches are initialized using a parameter remapping strategy based on the seed network $\mathcal{W}_0$. For iRMB branches located in the same stage and sharing identical configurations (i.e., kernel size and expansion ratio) with $\mathcal{W}_0$, parameters are directly inherited from the corresponding layers of the seed network. For branches with mismatched configurations, alignment is performed as follows: (1) If the number of channels differs, the parameters from the seed network are adjusted via channel cropping or zero-padding to match the target dimensions. (2) If the kernel sizes differ, spatial alignment is achieved through kernel center cropping or interpolation. All non-iRMB branches in the supernet are randomly initialized. After this process, the initialized supernet $\mathcal{W}$ is obtained. Subsequently, the full MDE supernetwork $\mathcal{W}_{\text{mde}}$ is constructed by attaching the decoder architecture from IDisc [18].

As illustrated in Algorithm 1 and Fig. 1, the proposed framework consists of two main stages: (1) federated supernet training and (2) personalized bi-objective evolutionary search. In the federated supernet training stage (Lines 3–11), the server first samples M architecture encodings using Latin Hypercube Sampling [19] and instantiates M sub-networks $\{\mathcal{A}_1, \ldots, \mathcal{A}_M\}$ from the MDE supernet $\mathcal{W}_{\text{mde}}$ accordingly (Line 4). These sub-networks are distributed to a selected set of clients $\mathcal{K}$. For each client $k \in \mathcal{K}$ (Line 5), the corresponding sub-network $\mathcal{A}_i$ and its initial parameters θ_i are transmitted (Line 6). Each client performs local training on its private MDE dataset $D_{\text{train}}^{\text{mde}}$ for a few steps and returns the updated weights $\hat{\theta}_i$ to the server (Lines 7–9). Due to architectural variations among sub-networks, the server reconstructs a full supernet copy for each received $\hat{\theta}_i$ using the current supernet $\mathcal{W}_{\text{mde}}$ as a template. The received parameters are inserted into the corresponding branches, while the remaining branches are filled with weights from $\mathcal{W}_{\text{mde}}$. A weighted aggregation is then performed across all supernet copies, with weights proportional to the clients' data volumes, yielding updated parameters θ_r for the global supernet (Line 10). After R rounds of federated training, the supernet $\mathcal{W}_{\text{mde}}$ is fully pre-trained on decentralized MDE data. In the personalized bi-objective evolutionary search stage (Lines 12–24), each client performs a local architecture search using the NSGA-II [20] algorithm to jointly optimize prediction accuracy and computational cost. Given the pre-trained supernet $\mathcal{W}_{\text{mde}}$, each client $k \in \mathcal{K}$ initializes a parent population $P^{(0)}$ of N binary-encoded individuals (Line 13). In each generation, an offspring population $Q^{(t)}$ of size N is generated via crossover and mutation operators (Line 15). The parent and offspring populations are merged into a combined population $R^{(t)}$ of size $2N$ (Line 16). For each individual $\gamma \in R^{(t)}$, a sub-network $\mathcal{A}$ is instantiated from the supernet based on its encoding (Line 18), and corresponding weights are inherited directly from $\mathcal{W}_{\text{mde}}$ (Line 19). The client evaluates $\mathcal{A}$ on its local validation dataset $D_{\text{val}}^{\text{mde}}$ to obtain the objective values: Abs Rel (f_1) and computational complexity (f_2) (Line 20). Non-dominated sorting and crowding distance are then applied to select the next parent population $P^{(t)}$ (Line 22). After T generations, the final client-specific Pareto front $\mathcal{P}_k$ is extracted (Line 24), representing a set of architectures that achieve optimal trade-offs between accuracy and efficiency for deployment on client k.

Algorithm 1 PTF-EMDE Pipeline

Input: Clients $\mathcal{K}$, population size N, evolutionary iterations T, federated rounds R, ImageNet-pretrained EMO [16] seed network $\mathcal{W}_0$, local training and validation datasets $D_{\text{train}}^{\text{mde}}$, $D_{\text{val}}^{\text{mde}}$

Output: Client-specific Pareto fronts $\{\mathcal{P}_k\}_{k \in \mathcal{K}}$

// Initialize supernet $\mathcal{W}$

1: Initialize the weights of the supernet $\mathcal{W}$ by Parameter remapping [21] from the ImageNet-pretrained EMO seed network $\mathcal{W}_0$

2: $\mathcal{W}_{\text{mde}} \leftarrow$ Construct MDE model based on $\mathcal{W}$ and IDisc [18] decoder

// Federated supernet pre-training

3: **for** $r = 1$ **to** R **do**

4: $\{\mathcal{A}_1, \ldots, \mathcal{A}_M\} \leftarrow$ Server samples M sub-networks from $\mathcal{W}_{\text{mde}}$ (Latin Hypercube Sampling)

5: **for** each selected client $k \in \mathcal{K}$ **do**

6: $\{\mathcal{A}_i, \theta_i\} \leftarrow$ Client k receives sub-network and parameters

7: $\{\mathcal{A}_i, \hat{\theta}_i\} \leftarrow$ Train $\mathcal{A}_i$ on local $D_{\text{train}}^{\text{mde}}$

8: $\{\hat{\theta}_i\} \leftarrow$ Uploads to server

9: **end for**

10: $\{\mathcal{W}_{\text{mde}}, \theta_r\} \leftarrow$ Aggregate all client-uploaded $\hat{\theta}_i$ via method in [22]

11: **end for**

// Personalized evolutionary search

12: **for** each client $k \in \mathcal{K}$ **do**

13: $P^{(0)} \leftarrow$ Initialize parent population (N individuals);

14: **for** $t = 1$ to T **do**

15: $Q^{(t)} \leftarrow$ Generate offspring via crossover & mutation operators

16: $R^{(t)} \leftarrow P^{(t-1)} \cup Q^{(t)}$

17: **for** each $\gamma \in R^{(t)}$ **do**

18: $\mathcal{A} \leftarrow$ Derive architecture by γ from $\mathcal{W}_{\text{mde}}$

19: $\{\mathcal{A}, \theta\} \leftarrow$ Inherit weights from $\mathcal{W}_{\text{mde}}$

20: $(f_1, f_2) \leftarrow$ Evaluate $\mathcal{A}$ on local $D_{\text{val}}^{\text{mde}}$ (f_1: Abs Rel, f_2: computational complexity)

21: **end for**

22: $P^{(t)} \leftarrow$ Select N individuals via non-dominated sorting and crowding distance

23: **end for**

24: $\mathcal{P}_k \leftarrow$ Extract Pareto front from $P^{(T)}$

25: **end for**

IV. EXPERIMENTAL RESULTS

To evaluate the effectiveness and efficiency of the proposed method, we conduct experiments on widely used benchmark datasets for MDE. This section provides an overview of the datasets, implementation details, and evaluation protocols used in our experiments.

A. Benchmark Datasets

We evaluate the proposed method on the widely used outdoor benchmark dataset KITTI [23]. KITTI was collected using a suite of sensors mounted on a moving vehicle and includes stereo image pairs along with corresponding Velodyne LiDAR scans from diverse outdoor driving scenarios, such as urban and rural roads. Both the RGB images and ground-truth depth maps have a resolution of 1241×376 pixels. Following the standard Eigen split [9], we use 23,158 left-view images for training and 652 images for testing. To ensure fair comparison with prior work, all test images are center-cropped according to the protocol described in [24].

B. Federated Data Partitioning

To simulate FL scenarios with different data heterogeneity, we construct IID and non-IID data partitions. In the IID setting, all samples are randomly shuffled and uniformly distributed across clients. In the non-IID setting, data is partitioned by driving routes, where each client is assigned a disjoint subset of routes and only observes samples from its own routes. Furthermore, each client splits its local dataset into 90% training data and 10% validation data. The training subset is used for supernet optimization and subnet fine-tuning, while the validation subset is exclusively used for architecture evaluation during the evolutionary search. These two configurations allow us to comprehensively assess the convergence behavior and generalization performance of PTF-EMDE under varying levels of data heterogeneity.

C. Baselines

To assess the effectiveness of the proposed PTF-EMDE framework, we evaluated it against five FL baselines. FedAvg [13] serves as a reference baseline, while FedProx [25], FedAS [26], and FedALA [3] provide partial personalization at the parameter level. PerFedRLNAS [6] represents a personalized federated NAS method that performs adaptation at the architecture level. For FedAvg, FedProx, FedAS, and FedALA, the encoder is set to EfficientNet-B4 [27], and all methods use the same decoder from IDisc [18]. Training hyperparameters are kept identical across all methods.

D. Implementation Details

All experiments are implemented in PyTorch and conducted on an NVIDIA RTX L40s GPU. To train the supernet efficiently, we adopt the DeepSpeed framework with ZeRO-2 optimization enabled. The batch size is set to 8 with 2 steps of gradient accumulation, and optimization is performed using AdamW [28]. The learning rate follows a warm-up schedule, increasing linearly from 0 to 1×10^{-4} over the first 1,000 iterations, with a weight decay of 1×10^{-4}. Federated training is conducted over 8 clients for 50 communication rounds.

For the evolutionary architecture search, we use a population size of 30 and perform 30 generations of evolution. New candidate architectures are generated using three-point crossover with a probability of 0.9 and polynomial mutation with an exponent of 0.3, enabling effective exploration of the search space. Each candidate inherits weights from the federated pre-trained supernet and is evaluated on the client's local validation set to compute its fitness. After obtaining the Pareto front for each client, the best-performing architecture is selected and retrained in a federated manner for 20 additional rounds. During retraining, we use a learning rate of 3×10^{-4}, a batch size of 8, 2 gradient accumulation steps, and 600 local iterations per client, again using AdamW.

For evaluation, we adopt seven commonly used depth estimation metrics, including δ_1, δ_2, δ_3, Abs Rel, RMSE, $\text{RMSE}_{\log}$ and $\log_{10}$.

V. RESULTS

In this section, we present the evaluation results of PTF-EMDE, covering seven depth estimation metrics and the associated computational and storage costs measured in terms of floating-point operations (FLOPs) and model parameters.

A. Results on KITTI under IID Settings

Under the IID setting, quantitative results are summarized in Table I. FedAvg achieves competitive accuracy but incurs high computational cost due to its full architecture, limiting its suitability for resource-constrained edge devices. PFL methods (FedAS, FedALA, FedProx) provide only marginal accuracy improvements while retaining substantial computation and parameter overhead. PerFedRLNAS improves the accuracy–efficiency trade-off by discovering personalized architectures. In comparison, PTF-EMDE further reduces computational cost (average 170.25 GFLOPs per client) while achieving superior performance on key metrics, including δ_1, Abs Rel, and RMSE, demonstrating a better balance between accuracy and efficiency. Fig. 4 provides qualitative comparisons. To enable objective analysis, we focus on the dashed-box regions. Baseline methods exhibit three consistent failure modes: (i) incomplete or discontinuous depth responses for small-scale structures (e.g., tree trunks and traffic signs in row 1 and row 3), (ii) over-smoothed depth that removes structural gaps (e.g., the rider–bicycle gap in row 1), and (iii) blurred foreground–background transitions with merged object boundaries (e.g., multiple rider–bicycle pairs in row 2 and distant two pedestrians in row 4). In contrast, PTF-EMDE preserves more continuous structural details, sharper depth discontinuities, and clearer foreground–background separation, indicating more reliable depth modeling in challenging regions.

B. Results on KITTI under Non-IID Settings

Under the non-IID setting, the quantitative results are reported in Table I. Compared to the IID scenario, non-IID partitioning introduces significant statistical heterogeneity among clients, posing a greater challenge for federated model optimization. As shown in the table, baseline methods suffer noticeable performance degradation. For example, metrics such as δ_1 decrease considerably, while computational cost (175.55 GFLOPs) and parameter count (23.23M) remain high. In contrast, the architectures discovered by the proposed PTF-EMDE framework maintain a much lower computational footprint, with only 170.07 GFLOPs and 13.36M parameters on average. At the same time, δ_1 decreases by merely 0.0015 compared to the IID setting, indicating that our method preserves estimation performance despite data heterogeneity. These results clearly demonstrate the ability of PTF-EMDE to balance model accuracy and efficiency under challenging non-IID conditions, confirming its suitability for deployment on edge devices.

Figure 5 presents qualitative results under the non-IID setting. Compared to the IID case, data heterogeneity further amplifies the limitations of baseline methods (e.g., FedAvg,

TABLE I
QUANTITATIVE RESULTS ON KITTI UNDER IID AND NON-IID SETTINGS.

Method	$\delta_1 \uparrow$	$\delta_2 \uparrow$	$\delta_3 \uparrow$	Abs Rel $\downarrow$	Log10 $\downarrow$	RMSE $\downarrow$	$\text{RMSE}_{\log} \downarrow$	Params (M)	FLOPs (G)
IID Setting									
Fedavg [13]	0.9421	0.9918	0.9984	0.074	0.0328	2.8582	0.112	23.23	175.55
Fedprox [25]	0.9435	0.9921	0.9984	0.0721	0.0316	2.8372	0.11	23.23	175.55
Fedala [3]	0.9489	0.9927	0.9985	0.0695	0.0302	2.7574	0.106	23.23	175.55
Fedas [26]	0.9517	0.9933	0.9986	0.0679	0.0297	2.7422	0.1036	23.23	175.55
PerFedRLNAS [6]	0.9542	0.9935	0.9985	0.0637	0.0284	2.5955	0.0993	**13.39**	171.19
PTF-EMDE	**0.9601**	**0.994**	**0.9987**	**0.0609**	**0.0266**	**2.4775**	**0.0952**	13.41	**170.25**
Non-IID Setting									
Fedavg [13]	0.9352	0.9911	0.9983	0.0771	0.0335	2.9577	0.116	23.23	175.55
Fedprox [25]	0.937	0.9912	0.9982	0.0752	0.0331	2.9676	0.115	23.23	175.55
Fedala [3]	0.944	0.992	0.9984	0.0726	0.0318	2.8349	0.1103	23.23	175.55
Fedas [26]	0.9485	0.9928	0.9985	0.0699	0.0306	2.8019	0.1064	23.23	175.55
PerFedRLNAS [6]	0.9467	0.9924	0.9983	0.0661	0.0301	2.6712	0.1063	13.43	170.92
PTF-EMDE	**0.9586**	**0.9937**	**0.9986**	**0.0618**	**0.027**	**2.5013**	**0.0959**	**13.36**	**170.07**

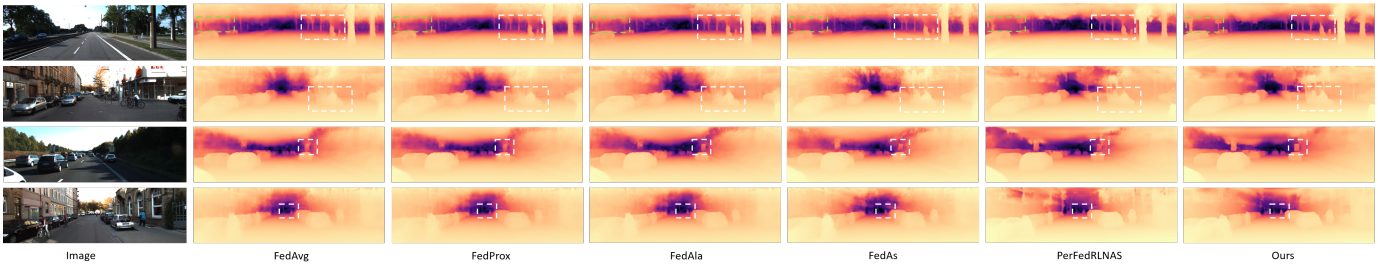


Fig. 4. Qualitative comparisons on KITTI under the IID setting. The first column displays the input raw images, while the subsequent columns present the depth prediction maps of four federated methods (employing the fixed EfficientNetB4 encoder), a personalized federated NAS method, and our proposed model, respectively.

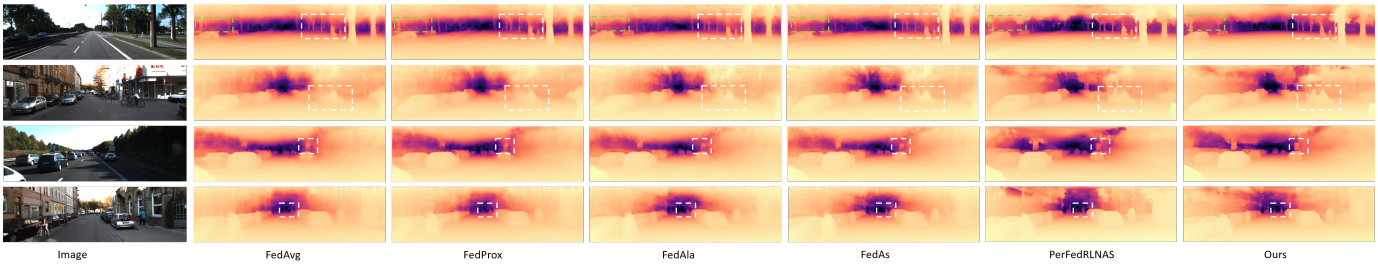


Fig. 5. Qualitative comparisons on KITTI under the non-IID setting. The first column displays the input raw images, while the subsequent columns present the depth prediction maps of four federated methods (employing the fixed EfficientNetB4 encoder), a personalized federated NAS method, and our proposed model, respectively.

FedProx, FedALA, FedAS), leading to more severe boundary blurring and incomplete object predictions in complex regions. In the dashed-box regions across all four rows, baseline methods still fail to capture complete structural details (e.g., tree trunks, traffic signs), produce over-smoothed depth that merges adjacent objects (e.g., rider and bicycle), and struggle to separate distant pedestrians from the background. In contrast, our PTF-EMDE maintains sharper depth discontinuities and more stable foreground-background separation, demonstrating stronger robustness to data heterogeneity. In summary, the proposed PTF-EMDE framework achieves superior depth estimation accuracy and significantly lower computational cost under both IID and non-IID settings. These results highlight its robustness, efficiency, and strong potential for real-world federated deployment on resource-constrained edge devices. Furthermore, in the federated supernet retraining stage with 8 clients over 50 rounds, the total communication amounts to approximately 11,416M parameters (42.53 GB under FP32 precision). During the architecture search stage, the supernet is broadcast only once (176.72M parameters, 0.66 GB). Overall, the total communication cost of the entire algorithm is approximately 43.19 GB.

VI. Conclusion

In this work, we propose PTF-EMDE, a bi-objective federated NAS framework tailored for MDE. By introducing a lightweight encoder search space and formulating the search process as a bi-objective evolutionary optimization problem, PTF-EMDE efficiently discovers personalized architectures that balance depth estimation accuracy and computational efficiency. Experimental results on the KITTI benchmark demonstrate that our method achieves competitive performance while significantly reducing computational cost, highlighting its potential for practical deployment on resource-constrained edge devices.

References

[1] J. Zheng, C. Lin, J. Sun, Z. Zhao, Q. Li, and C. Shen, "Physical 3d adversarial attacks against monocular depth estimation in autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 24 452–24 461.

[2] L. Collins, H. Hassani, A. Mokhtari, and S. Shakkottai, "Exploiting shared representations for personalized federated learning," in *International Conference on Machine Learning*, vol. 139. PMLR, 2021, pp. 2089–2099.

[3] J. Zhang, Y. Hua, H. Wang, T. Song, Z. Xue, R. Ma, and H. Guan, "Fedala: Adaptive local aggregation for personalized federated learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 9, 2023, pp. 11 237–11 244.

[4] A. Garg, A. K. Saha, and D. Dutta, "Direct federated neural architecture search," *arXiv preprint arXiv:2010.06223*, 2020.

[5] E. Mushtaq, C. He, J. Ding, and S. Avestimehr, "Spider: Searching personalized neural architecture for federated learning," *arXiv preprint arXiv:2112.13939*, 2021.

[6] D. Yao and B. Li, "Perfedrlnas: One-for-all personalized federated neural architecture search," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 15, 2024, pp. 16 398–16 406.

[7] B. Cao, H. Deng, Y. Hao, and X. Luo, "Multi-view information fusion based on federated multi-objective neural architecture search for mri semantic segmentation," *Information Fusion*, vol. 123, p. 103301, 2025.

[8] H. R. Roth, D. Yang, W. Li, A. Myronenko, W. Zhu, Z. Xu, X. Wang, and D. Xu, "Federated whole prostate segmentation in mri with personalized neural architectures," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2021, pp. 357–366.

[9] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[10] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper depth prediction with fully convolutional residual networks," in *2016 Fourth International Conference on 3D Vision (3DV)*. IEEE, 2016, pp. 239–248.

[11] W. Yuan, X. Gu, Z. Dai, S. Zhu, and P. Tan, "Neural window fully-connected crfs for monocular depth estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 3916–3925.

[12] I. Grigore and C.-A. Popa, "Mambadepth: Enhancing long-range dependency for self-supervised fine-structured monocular depth estimation," *arXiv preprint arXiv:2406.04532*, 2024.

[13] 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.

[14] E. F. d. S. Soares and C. A. V. Campos, "Privacy-preserving training of monocular depth estimators via self-supervised federated learning," in *2024 IEEE 100th Vehicular Technology Conference (VTC2024-Fall)*. IEEE, 2024, pp. 1–6.

[15] F. D. S. Elton, E. V. Brazil, and C. A. V. Campos, "Leveraging bayesian optimization to enhance self-supervised federated learning of monocular depth estimators," in *2024 IEEE 27th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2024, pp. 2420–2425.

[16] J. Zhang, X. Li, J. Li, L. Liu, Z. Xue, B. Zhang, Z. Jiang, T. Huang, Y. Wang, and C. Wang, "Rethinking mobile block for efficient attention-based models," in *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE Computer Society, 2023, pp. 1389–1400.

[17] C. Xiao, M. Li, Z. Zhang, D. Meng, and L. Zhang, "Spatial-mamba: Effective visual state space models via structure-aware state fusion," *arXiv preprint arXiv:2410.15091*, 2024.

[18] L. Piccinelli, C. Sakaridis, and F. Yu, "idisc: Internal discretization for monocular depth estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 21 477–21 487.

[19] M. D. McKay, R. J. Beckman, and W. J. Conover, "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics*, vol. 42, no. 1, pp. 55–61, 2000.

[20] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[21] J. Fang, Y. Sun, Q. Zhang, K. Peng, Y. Li, W. Liu, and X. Wang, "Fna++: Fast network adaptation via parameter remapping and architecture search," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 9, pp. 2990–3004, 2020.

[22] H. Zhu and Y. Jin, "Real-time federated evolutionary neural architecture search," *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 2, pp. 364–378, 2021.

[23] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[24] R. Garg, V. K. Bg, G. Carneiro, and I. Reid, "Unsupervised cnn for single view depth estimation: Geometry to the rescue," in *European Conference on Computer Vision*. Springer, 2016, pp. 740–756.

[25] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine Learning and Systems*, vol. 2, pp. 429–450, 2020.

[26] X. Yang, W. Huang, and M. Ye, "Fedas: Bridging inconsistency in personalized federated learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 11 986–11 995.

[27] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International Conference on Machine Learning*, vol. 97. PMLR, 2019, pp. 6105–6114.

[28] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.