数据进行重训练，容易导致过拟合并降低模型性能。(ii) 时间资源开销大。设备端微调计算梯度更新模型参数的计算量大，不适合实时要求的设备应用，还会耗费大量设备计算资源，影响设备的续航。

本文提出了一种新型 DML 框架，称为“设备-云协作参数生成框架”（DUET），以解决上述问题。DUET 的核心思路是利用端侧个性化数据动态生成模型权重，从而完成学习任务。正如图 1(c) 所示，本文的框架包括以下部分：(1) 通用元网络（UMN）在云端收集来自所有设备的数据，并进行标注和训练。为提升 DMG 效果，训练好的模型被划分为静态层和动态层，静态层（骨干网络）的参数固定，而动态层（分类器）的参数则基于设备的实时数据动态生成。(2) 个性化参数生成器（PPG）利用 HyperNetworks 为每个设备生成独立的分类器参数。每个设备的实时样本输入 PPG 以生成个性化分类器权重，然后云端将该参数传送到设备端，从而实现实时个性化推理。PPG 仅需前向传播，无需额外计算，使得 DUET 具有实时性。(3) 本文还设计了稳定权重适配器（SWA），通过多个 PPG 间的自适应相关性预测最优权重，以减缓动态模型的性能波动。

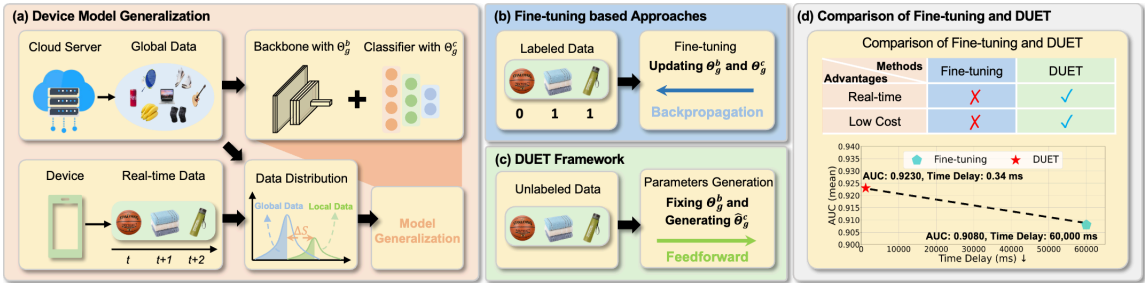

图 1. 常见设备模型泛化方法

2 相关工作

2.1 轻量级神经网络

传统神经网络 [4,14,23,24] 在各类任务中表现出色，但在设备端部署时需要考虑设备的存储空间和计算能力。因此，近年来提出了许多轻量级卷积神经网络（CNN）模型，如 SqueezeNet、MobileNet 和 ShuffleNet 等。SqueezeNet [10] 通过广泛使用 1×1 卷积的 Fire 模块减少参数数量；MobileNetV1 将传统卷积核分解为深度卷积核和点卷积核；MobileNetV2 [15] 引入反向残差和线性瓶颈结构；MobileNetV3 [9] 结合 AutoML 进行网络结构优化以提升效率；ShuffleNetV1 [25] 通过通道分组混洗增强通道信息交换，ShuffleNetV2 [13] 进一步引入通道拆分来提升推理速度；EfficientNet [18] 使用神经网络架构搜索（NAS）与混合缩放策略；GhostNet [7] 通过参数较少的线性变换层生成“幽灵”特征图。这些模型在参数量和浮点运算（FLOPs）较小的情况下仍能获得良好性能，但参数量的限制在一定程度上限制了模型的泛化能力。

2.2 超网络

超网络是一类生成其他网络参数的神经网络。Ha 等人首次提出了超网络 [6]，利用超网络生成的权重减少了训练所需的参数量。之后，关于超网络的研究逐渐增多，涵盖了参数初始化 [2]、持续学习 [19]、图网络 [22]、元学习 [20] 和联邦学习 [16] 等应用。近年来，超网络研究

逐渐集中在根据不同数据输入生成特定的网络参数。例如，HyperStyle [1] 和 HyperInverter [5] 利用超网络为不同图像生成特定的解码器参数，以提升图像重建质量。在论文的研究中，论文将超网络适配至设备-云协作系统，专门应对这一场景中的特定挑战。

3 本文方法

3.1 概述

本文方法对应模型框架图如图 2 所示，模型的核心包含三个组件：通用元网络 (Universal Meta Network,UMN)，个性化参数生成器 (Personalized Parameters Generator,PPG)，参数稳定适配器 (Stable Weight Adapter,SWA)。通用元网络 UMN 的主要作用是通过全局用户数据训练一个 backbone，backbone 被称为静态层，全局用户共享一个 backbone。个性化参数生成器 PPG 基于特定设备的实时样本生成个性化分类器（动态层）参数以增强个性化。由于直接使用超网通常会在学习过程中产生较大的波动，导致不稳定的预测，本文使用参数稳定适配器 SWA 来提高预测的稳定性。

在以下章节中会详细介绍框架中的各个组件。

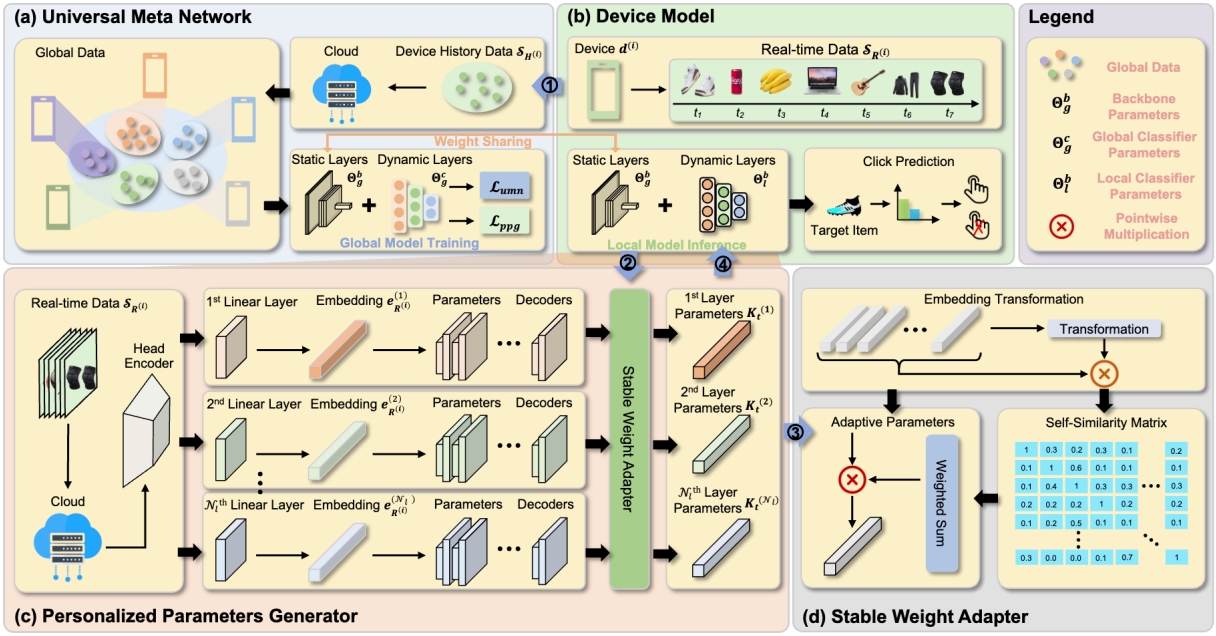

图 2. 模型框架示意图

3.2 通用元网络 UMN

在通用元网络（UMN，见图 3）中，论文训练一个带有骨干网络和分类器的基础模型，用作全局云模型。给定设备集合 $D = \{d^{(i)}\}_{i=1}^{N_d}$ 及其对应的历史数据 $S_{H^{(i)}} = \{x_{H^{(i)}}^{(j)}, y_{H^{(i)}}^{(j)}\}_{j=1}^{N_{H^{(i)}}}$，UMN 的目标可以表述为以下优化问题：

$$\min_{\Theta_g^b, \Theta_g^c} L_{umn} = \sum_{i=1}^{N_d} \sum_{j=1}^{N_{R^{(i)}}} D_{ce}(y_{H^{(i)}}^{(j)}, \Omega(x_{H^{(i)}}^{(j)}; \Theta_g^b); \Theta_g^c)) \tag{1}$$

其中 D_{ce} 表示两个概率分布间的交叉熵，$\Omega(x^{(j)}_{H^{(i)}};\Theta^b_g)$ 表示骨干网络从样本 $x^{(j)}_{H^{(i)}}$ 中提取的特征。为了实现个性化模型泛化，论文将联合训练的骨干和分类器模块解耦为“静态层”和“动态层”：

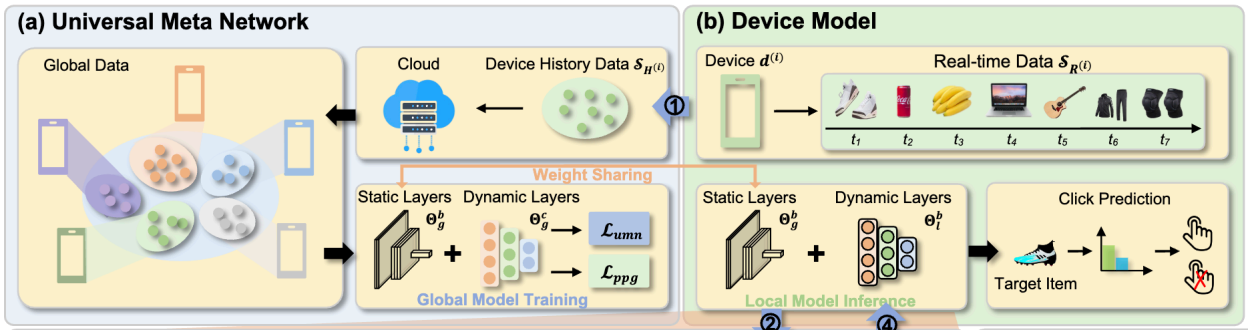

图 3. 通用元网络 UMN

- 静态层：通过全局数据训练的骨干网络参数 Θ^b_g 能够准确地将用户行为映射到特征空间中，论文将骨干网络固定为“静态层”，以生成与全局数据分布相关的泛化表示。
- 动态层：根据用户的具体行为，将个性化样本输入 PPG，生成个性化的分类器权重 Θ^c_l，通过调整分类器参数实现个性化泛化的提升。

3.3 个性化参数生成器 PPG

个性化参数生成器（PPG）基于特定设备的实时样本生成个性化分类器的动态参数 Θ^l_c，以提升对不同数据分布的泛化能力。

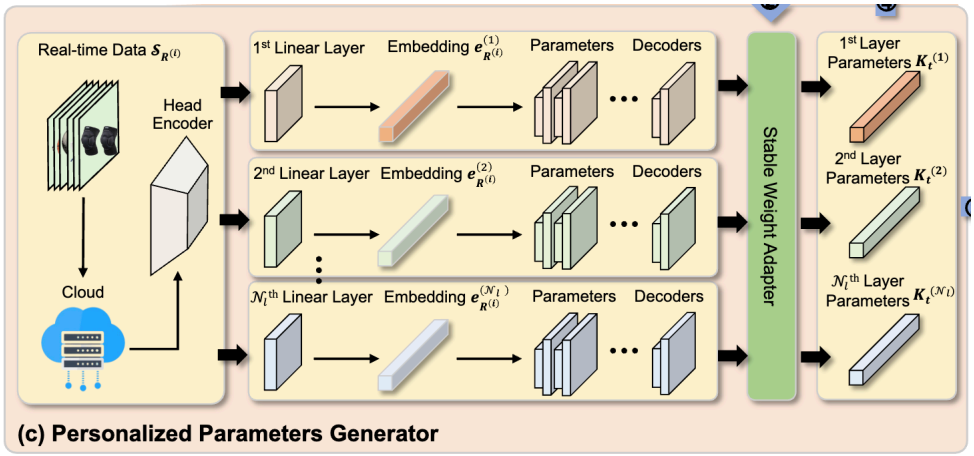

图 4. 个性化参数生成器 PPG

论文使用每个会话中的实时样本 $S_{R^{(i)}} = \{x^{(j)}_{R^{(i)}}\}_{j=1}^{N_{R^{(i)}}}$ 来生成模型参数。为了生成主模型中“动态层”第 n 层的参数，本文使用了一个层编码器，将第 n 层的参数表示为嵌入 $\mathbf{e}^{(n)}_{R^{(i)}}$。为了建模不同层的关系，而不是构建一对一的编码器层对应关系，$\mathbf{e}^{(n)}_{R^{(i)}}$ 共享一个编码器颈部，但使用不同的线性层来改变实时数据特征。

$$\mathbf{e}^{(n)}_{R^{(i)}} = L^{(n)}_{\text{layer}}(E_{\text{share}}(S_{R^{(i)}})), \quad \forall n = 1, \ldots, N_l \tag{2}$$

其中 $E_{\text{share}}(\cdot)$ 表示共享编码器颈部，$L_{\text{layer}}^{(n)}(\cdot)$ 是用于将 $E_{\text{share}}(\cdot)$ 的输出调整为第 n 个动态层特征的线性层。

然后论文使用生成器 $g(\cdot)$ 将实时数据特征转换为主模型的参数：

$$K_{R^{(i)}}^{(n)} = g^{(n)}(\mathbf{e}_{R^{(i)}}^{(n)}) \tag{3}$$

具体来说，论文将 $e_{R^{(i)}}^{(n)}$ 输入以下两个 MLP 层中，根据主模型的“动态层”一致结构生成参数：

$$\mathbf{w}_{R^{(i)}}^{(n)} = (W_1\mathbf{e}_{R^{(i)}}^{(n)} + B_1)W_2 + B_2 K_{R^{(i)}}^{(n)} = \mathbf{w}_{R^{(i)}}^{(n)} + \mathbf{b}_{R^{(i)}}^{(n)} \tag{4}$$

其中两层 MLP 的权重分别用 W_1 和 W_2 表示，B_1 和 B_2 表示偏置项。在云端训练中，PPG 的所有层和主模型的静态层将基于全局历史数据 $S_H^{(i)} = \{x_H^{(j)(i)}, y_H^{(j)(i)}\}_{j=1}^{N_H^{(i)}}$ 一同优化，而不是先优化主模型的静态层再优化 PPG。PPG 的损失函数 L_{ppg} 定义如下：

$$\min_{\Theta_g^b, \Theta_p} L_{\text{ppg}} = \sum_{i=1}^{N_d} \sum_{j=1}^{N_{R^{(i)}}} \gamma^t \, D_{\text{ce}}(y_{H^{(i)}}^{(j)}, \Omega(x_{H^{(i)}}^{(j)}; \Theta_g^b); g(\mathbf{e}_{R^{(i)}}^{(n)}; \Theta_p)) \tag{5}$$

3.4 稳定权重适配器 SWA

直接使用超网通常会在学习过程中产生较大的波动，导致不稳定的预测。为了解决这个问题，论文提出了稳定权重适配器（SWA）模块，以提高预测的稳定性。

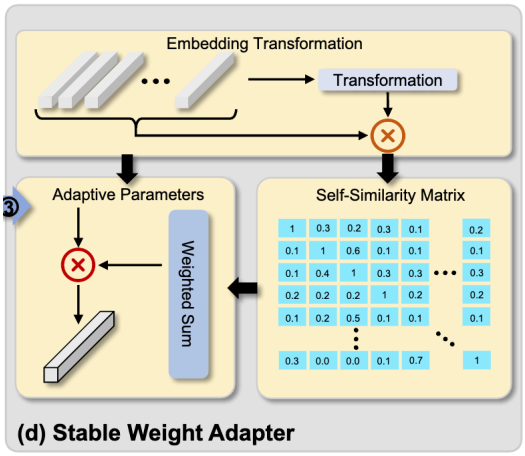

图 5. 稳定权重适配器 SWA

首先，论文开发了 N_p 个 PPG 来生成一组参数，而不是仅使用一个生成器，记作 W_1'。然后，将多个 W_1' 拼接成一个矩阵，记作 $W_1' = \{W_{1,1}', W_{1,2}', \ldots, W_{1,m}'\}$。$W_{i,j}'$ 表示第 j 个生成器的第 i 层 MLP。然后，论文可以计算 $W_{1,i}'$ 和 $W_{1,j}'$ 之间的相似性，从而得到一个维度为 $m \times m$ 的自相似矩阵 S：

$$S = W_1' \times (W_1')^T \tag{6}$$

对 S 按行求和，论文得到权重向量 $\mathbf{p}' = \{p_1', p_2', \ldots, p_m'\}$ 维度为 $m \times 1$。其中 p_i 可以视为多个生成器中 $W_{1,i}'$ 的重要性。论文还设置温度来调整最终权重向量 $\mathbf{p}$：

$$p_i = \text{Softmax}\left(\frac{p_i'/\tau}{\sum_j p_j'/\tau}\right) \tag{7}$$

然后可以计算最终的 W_1 和 W_2：

$$W_1 = \sum_{i=0}^{m} p_i \times W'_{1,i} \tag{8}$$

最后，论文使用等式（8）来获取 W_1 和 W_2，并在等式（4）中替换 W_1 和 W_2 以获得模型参数。

4 复现细节

4.1 与已有开源代码对比

本报告实现的 DUET 框架已公布部分源代码，源代码实现了基于 SASRec 模型的 DUET 框架。但数据预处理代码以及论文中展示的基于 DIN、GRU4Rec 的 DUET 框架代码并未给出。本人实现了数据预处理逻辑以及论文中展示的基于 DIN、GRU4Rec 的 DUET 框架，最终效果与论文实验部分数据仅在百分位或者千分位上有差距，基本复现了论文中的结果，本人也分析了产生差距的原因，可能是数据预处理过程中所选的用户和物品不一致以及随机种子设置不同所导致。

4.2 数据集与评价指标

在本报告中，采用原论文使用的两个 Movielens 数据集。MovieLens-100K 是一个经典的推荐系统数据集，包含 100,000 条用户评分记录，涉及 943 名用户和 1,682 部电影，评分范围为 1 到 5。数据时间跨度为 1997 年 9 月至 1998 年 4 月，提供用户、电影及评分的核心信息以及电影和用户的元数据。由于规模小、格式简单，适合快速实验和模型验证。MovieLens-1M 是一个更大规模的数据集，包含 1,000,209 条评分，涉及 6,040 名用户和 3,952 部电影，评分范围同样为 1 到 5，数据时间跨度从 2000 年 4 月至 2003 年 3 月。该数据集还包括用户的详细信息（如年龄、性别、职业）和电影的元信息，常用于推荐系统的性能评估和复杂模型的研究。

本人实现了数据集滑动窗口式的预处理逻辑，主要包含两点：固定长度序列填充、滚动更新历史序列。固定长度序列填充指数据中每条记录均为固定长度（即长度相同），使用 0 作为填充值，表示无效位置，常用于深度学习模型中的序列建模，以保证输入到模型的序列具有统一的长度，便于批处理和加速计算。滚动更新历史序列指每当用户有新的一次点击行为时，数据会在序列的末尾添加新的物品，而之前的序列记录则被左移，去掉最早的物品，以保持长度不变，这种处理方式通常用于实时性较高的场景，以确保模型捕获到最新的用户行为。

为评估模型性能，本人采用了与论文一致的评估指标，即 AUC（Area Under the Curve）指标，AUC 是一个常用的二分类模型评价指标，用于衡量模型区分正负样本的能力。

4.3 复现结果

表 1 展示了原论文中的结果。表 2 展示了本人复现的结果。

其中基于 DIN、GRU4Rec 的 base、finetune 过程以及 DUET 框架由本人实现。通过论文中实验结果与复现结果的对比，可以看到论文中结果与本人复现结果基本相似，仅在百分

Model	Method	Dataset	
		Movielens-1M	Movielens-100K
DIN	base	0.9077	0.8348
	finetune	0.9080	0.8429
	DUET	0.9230	0.8581
SASRec	base	0.9280	0.8721
	finetune	0.9279	0.8719
	DUET	0.9326	0.8723
GRU4Rec	base	0.9279	0.8723
	finetune	0.9286	0.8711
	DUET	0.9311	0.8755

表 1. 论文中的实验结果

Model	Method	Dataset	
		Movielens-1M	Movielens-100K
DIN	base	0.9198	0.8845
	finetune	0.9286	0.8917
	DUET	0.9306	0.9004
SASRec	base	0.9375	0.9079
	finetune	0.9393	0.9070
	DUET	0.9389	0.9067
GRU4Rec	base	0.9378	0.9060
	finetune	0.9388	0.9066
	DUET	0.9389	0.9040

表 2. 本人复现结果

位或者千分位上有差距，基于 DUET 框架的各个模型的性能在大多数情况下要优于基础模型 (base)，由此可见本次复现工作成功。

4.4 改进

原始 DUET 存在的问题：DUET 将用户交互序列中不同位置的物品视为同等重要，没有对序列中物品的重要性进行区分。用户行为通常具有时间依赖性，越临近的行为可能越能反映用户的当前兴趣。原始模型未考虑这种动态特性，可能导致模型对用户兴趣的预测失准。

在用户行为序列中，近期的交互物品对用户当前状态的影响通常大于早期的行为。例如，用户可能刚搜索了某种商品，这类行为更能反映他们的即时需求。基于这种特性，本人考虑引入位置感知权重，通过引入位置感知权重，可以动态调节序列中每个物品对用户状态的贡献程度，从而让模型更聚焦于近期的关键行为。

位置感知权重可形式化为：

$$W_i^p = f(pos_i, SeqLen) \tag{9}$$

其中 pos_i 表示物品 i 在交互记录中位置，$SeqLen$ 表示交互记录的长度，$f(\cdot)$ 表示位置感知权重的计算方式，W_i^p 表示物品 i 在交互记录中位置感知权重。

由于不同的任务和数据对位置信息的依赖可能有很大差异，本文采用多种位置感知权重计算方式，分别如下：线性权重，权重随着位置线性增加，适合简单位置建模：

$$W_i^p = \alpha \frac{pos_i}{SeqLen} \tag{10}$$

平滑权重，各位置权重变化平缓，不会出现突变，适合强调序列整体趋势：

$$W_i^p = \frac{1}{1 + \alpha(SeqLen - pos_i)} \tag{11}$$

指数衰减权重，靠近序列尾部的权重更高，随着位置的前移，权重呈指数级快速衰减，适合需要短期记忆的任务：

$$W_i^p = \alpha * e^{\frac{pos_i - SeqLen}{\beta}} \tag{12}$$

高斯权重，围绕中心位置 μ 呈现高斯分布，中心位置的权重最大，远离中心的权重逐渐减小，对序列中的某一部分（例如某个时间段的行为）给予重点关注:

$$W_i^p = e^{-\frac{(pos_i - \mu)^2}{2\sigma^2}} \tag{13}$$

在 DUET 框架的基础上部署上述四种位置感知权重后，实验结果如表 3，其中 linear 表示线性权重，smooth 表示平滑权重，exp 表示指数权重，gaussian 表示高斯权重。

Model	**Method**	**Movielens-1M**	**Movielens-100K**
DIN	DUET	0.930643	0.900383
	DUET-PA-linear	0.931477	**0.901150**
	DUET-PA-smooth	0.930596	0.900569
	DUET-PA-exp	0.931287	0.900464
	DUET-PA-gaussian	**0.931809**	0.900714
SASRec	DUET	0.938873	**0.906741**
	DUET-PA-linear	0.938953	0.904589
	DUET-PA-smooth	0.938715	0.902327
	DUET-PA-exp	0.939617	0.903983
	DUET-PA-gaussian	**0.940085**	0.902489
GRU4Rec	DUET	0.938858	**0.904021**
	DUET-PA-linear	0.939187	0.902140
	DUET-PA-smooth	0.938413	0.900960
	DUET-PA-exp	0.938900	0.901106
	DUET-PA-gaussian	**0.939328**	0.901361

表 3. 改进后实验结果

通过分析表格可得，引入位置感知权重后 DUET 的效果得到了普遍提升，其中引入高斯权重后 DUET 的提升最为显著，这说明数据集中用户行为存在明显的时序特性或局部模式，对推荐效果起到关键作用。

5 总结与展望

论文中提出的 DUET 方法，用于通过从云端生成自适应设备模型参数来实现高效的设备模型泛化，而无需在设备端进行训练。DUET 能够有效地学习从实时样本到设备模型参数的映射函数，从而实现低延迟和更好的设备特定个性化。在 Movielens-1M、Movielens-100K 上的大量实验表明，DUET 在准确性和实时性能方面相比微调方法有显著优势，这验证了其在实际应用中的潜在价值。

DUET 框架作为一种端云协同的推荐系统设计方案，已经展示出一定的创新性和优势。然而，其发展空间依然很大，未来可以从以下几个方向改进或扩展：1、冷启动问题，当用户实时数据不足时，生成的参数质量难以保证；2、隐私问题，用户的实时数据需要上传云端进行参数生成，这个过程可能导致隐私的泄漏；3、参数生成效率，DUET 的参数生成网络虽然灵活，但仍然可能存在计算和存储的瓶颈，尤其是在设备端资源有限的情况下。

参考文献

[1] Yuval Alaluf, Omer Tov, Ron Mokady, Rinon Gal, and Amit Bermano. Hyperstyle: Stylegan inversion with hypernetworks for real image editing. In *Proceedings of the IEEE/CVF conference on computer Vision and pattern recognition*, pages 18511–18521, 2022.

[2] Oscar Chang, Lampros Flokas, and Hod Lipson. Principled weight initialization for hypernetworks. *arXiv preprint arXiv:2312.08399*, 2023.

[3] Zhengyu Chen and Donglin Wang. Multi-initialization meta-learning with domain adaptation. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1390–1394. IEEE, 2021.

[4] Zhengyu Chen, Teng Xiao, and Kun Kuang. Ba-gnn: On learning bias-aware graph neural network. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pages 3012–3024. IEEE, 2022.

[5] Tan M Dinh, Anh Tuan Tran, Rang Nguyen, and Binh-Son Hua. Hyperinverter: Improving stylegan inversion via hypernetwork. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11389–11398, 2022.

[6] David Ha, Andrew Dai, and Quoc V Le. Hypernetworks. *arXiv preprint arXiv:1609.09106*, 2016.

[7] Kai Han, Yunhe Wang, Qi Tian, Jianyuan Guo, Chunjing Xu, and Chang Xu. Ghostnet: More features from cheap operations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1580–1589, 2020.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[9] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1314–1324, 2019.

[10] Forrest N Iandola. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

[11] Mengze Li, Ming Kong, Kun Kuang, Qiang Zhu, and Fei Wu. Multi-task attribute-fusion model for fine-grained image recognition. In *Optoelectronic Imaging and Multimedia Technology VII*, volume 11550, pages 114–123. SPIE, 2020.

[12] Zheqi Lv, Feng Wang, Shengyu Zhang, Kun Kuang, Hongxia Yang, and Fei Wu. Personalizing intervened network for long-tailed sequential user behavior modeling. *arXiv preprint arXiv:2208.09130*, 2022.

[13] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)*, pages 116–131, 2018.

[14] Fang-Yu Qin, Zhe-Qi Lv, Dan-Ni Wang, Bo Hu, and Chao Wu. Health status prediction for the elderly based on machine learning. *Archives of gerontology and geriatrics*, 90:104121, 2020.

[15] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[16] Aviv Shamsian, Aviv Navon, Ethan Fetaya, and Gal Chechik. Personalized federated learning using hypernetworks. In *International Conference on Machine Learning*, pages 9489–9502. PMLR, 2021.

[17] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[18] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[19] Johannes Von Oswald, Christian Henning, Benjamin F Grewe, and João Sacramento. Continual learning with hypernetworks. *arXiv preprint arXiv:1906.00695*, 2019.

[20] Zhou Xian, Shamit Lal, Hsiao-Yu Tung, Emmanouil Antonios Platanios, and Katerina Fragkiadaki. Hyperdynamics: Meta-learning object and agent dynamics with hypernetworks. *arXiv preprint arXiv:2103.09439*, 2021.

[21] Junkun Yuan, Xu Ma, Defang Chen, Kun Kuang, Fei Wu, and Lanfen Lin. Domain-specific bias filtering for single labeled domain generalization. *International Journal of Computer Vision*, 131(2):552–571, 2023.

[22] Chris Zhang, Mengye Ren, and Raquel Urtasun. Graph hypernetworks for neural architecture search. *arXiv preprint arXiv:1810.05749*, 2018.

[23] Wenqiao Zhang, Siliang Tang, Yanpeng Cao, Shiliang Pu, Fei Wu, and Yueting Zhuang. Frame augmented alternating attention network for video question answering. *IEEE Transactions on Multimedia*, 22(4):1032–1041, 2019.

[24] Wenqiao Zhang, Lei Zhu, James Hallinan, Shengyu Zhang, Andrew Makmur, Qingpeng Cai, and Beng Chin Ooi. Boostmis: Boosting medical image semi-supervised learning with adaptive pseudo labeling and informative active annotation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 20666–20676, 2022.

[25] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6848–6856, 2018.