Pick-a-back: Selective Device-to-Device Knowledge Transfer in Federated Continual Learning

JinYi Yoon[1] and HyungJune Lee[2]*

[1] Virginia Tech, Blacksburg, VA, USA
jinyiyoon@vt.edu
[2] Ewha Womans University, Seoul, South Korea
hyungjune.lee@ewha.ac.kr

Abstract. With the explosion of edge intelligence, leveraging federated indirect knowledge has become crucial for boosting the tasks of individual learners. However, the conventional approach to knowledge reuse often leads to catastrophic forgetting issues. In this paper, we revisit the concept of continual learning in the context of edge intelligence and address the knowledge transfer problem to enhance federated continual learning. Since each learner processes private heterogeneous data, we propose *Pick-a-back*, a device-to-device knowledge federation framework by selectively reusing the external knowledge with similar behavioral patterns. By borrowing indirect experiences, an edge device can initiate learning from useful knowledge and thus achieve faster yet more generalized knowledge acquisition. Using continual tasks consisting of various datasets on lightweight architectures, we have validated that *Pick-a-back* provides a significant inference improvement of up to 8.0% via selective knowledge federation. Our codes are available at `https://github.com/jinyi-yoon/Pick-a-back.git`.

Keywords: Federated continual learning, edge intelligence, device-to-device knowledge federation, selective knowledge transfer

1 Introduction

As artificial intelligence (AI) permeates nearly every field, data has emerged as a pivotal source driving remarkable advancements. Most of the data originates from the end users or systems, stimulating a need for distributed learning on low-end edge devices to address privacy and latency concerns [2, 31]. However, individual low-end edge devices often lack the computational capability to generate advanced knowledge [6]. In the context of human learning paradigms, where any individuals become educated and exchange helpful knowledge with one another, the emergence of edge intelligence and its widespread availability allow external knowledge to serve as a valuable resource in supporting individual learners. By facilitating some sharable insights [28, 59] and harnessing collective intelligence [14], it has widely revolutionized learning paradigms across various fields, even where numerous data have non-identical forms and heterogeneous distributions.

* Corresponding author

While leveraging external knowledge to enhance individual tasks, conventional learners often encounter the challenge of catastrophic forgetting, wherein they adapt only to the assigned tasks, and consequently lose previously acquired knowledge in some degree over time. In contrast, to more effectively retain and reinforce pre-trained knowledge together with new knowledge, the learners would rather adopt a continual learning approach for ensuring both robustness and adaptability. To enhance the efficacy of individual *continual* learning, it is not imperative for learners to confine themselves to their own knowledge. The prevalence of edge intelligence may facilitate knowledge *federation*, enabling individual learners to leverage a broader or deeper understanding of data, which can be possibly heterogeneous. While recent works have devised federated continual learning frameworks, they typically take a centralized approach for assistance [3, 18, 50], rather than direct device-to-device knowledge transfer in a distributed manner.

As for edge AI, individual learners might be confronted with a series of diverse heterogeneous tasks. At the edge side accompanied with its own private data, each learner attempts to form its own distinct knowledge. Then, a question to raise is that *how can an individual edge learner effectively harness externally available prior knowledge?* It has been reported that there exists unique positive and negative knowledge at each model [43, 57]. Maybe the answer would be straightforward yet meaningful – the learners would *selectively* reuse or *customize* some knowledge that aligns well with their specific behavioral patterns. Rather than starting from scratch, learners can benefit from the indirectly attained knowledge formed by others who have already seen similar patterns. In that the decision patterns can not be exactly the same, *indirect knowledge transfer* (with follow-up training) may offer a rich source of otherwise untrainable and unattainable knowledge from private data. This can lead to performance improvement via selective acquisition of some common yet distinct knowledge.

In this work, we revisit the knowledge reuse problem. We consider three challenges in edge intelligence with knowledge transfer: 1) edge devices have limited computational and storage resources; 2) learners are not willing to disclose their raw data; and 3) edge devices should take into account the reduction of communication costs. To overcome these concerns, our approach aims to enhance the performance of lightweight learning models, while maintaining data privacy and avoiding additional communication overhead.

We propose a selective and customized knowledge federation framework from one device to another, based on heterogeneous continual learning, called ***Pick-a-back***, which ***pick***s ***a*** relevant ***back***bone knowledge from other devices. Beyond the achievable knowledge of each individual learner, we strategically search for some effective knowledge sources selectively and then customize the commonly shared knowledge to closely align with the local task. By leveraging decision similarity based on mutated data to assess the efficacy of the backbone among the neighboring models spread in the form of physically separated edge intelligence, we selectively fetch and customize constructive knowledge. This approach not only enables decentralized knowledge exchange among the learners at devices, but also facilitates a more efficient and scalable learning paradigm.

This work contributes to advancing the field of deep learning in edge intelligence by providing selective knowledge transfer and fostering collaboration among local learners. It can thereby enhance the efficiency of the individual continual learning process. The main contributions can be summarized as follows:

1) We address the problem of device-to-device knowledge federation in the continual learning. We demonstrate the potential of leveraging indirect knowledge extracted by the neighboring edge learners to obtain otherwise untrainable and unattainable insights. This work provides a way to maintain a model with heterogeneous data collected across a network of edge devices.
2) As the diverse tasks imply their own unique knowledge, we propose selective and customized knowledge transfer by capturing the decision pattern similarity relevant to a target task running on a device. *Pick-a-back* adopts each relevant knowledge for a specific task, achieving high fidelity through the acquisition of some common yet distinct knowledge from unseen examples, while preserving data privacy and communication cost at the edge.
3) Extensive experimental results on various datasets across different architectures have shown that *Pick-a-back* provides a significant inference improvement up to 8.0% via selective knowledge federation.

2 Related Work

Typical continual learning refers to training on a series of multiple tasks [22] or incremental classes [13]. The primary objective is to address the catastrophic forgetting problem, where previously learned knowledge is decaying over time as adapting to new tasks, using memorization [40,52] or regularization [9,23].

Knowledge-reusable continual learning. More closely related to our problem, there has been a growing interest in reusing existing knowledge for its subsequent tasks [30,58]. Unlike conventional approaches, these methods have focused on concatenating a new model to its previous model, allowing input data to propagate through the previous model [46]. Rather than reusing the whole model, researchers have proposed to activate only the essential parts of the model [19,29]. In *Piggyback* [29], they leverage a common model and mask out unnecessary neurons to extract relevant knowledge for a target task. It effectively reuses existing knowledge with only a single sharable model, but it is less able to accept incremental tasks. *CPG* [19] combines neuron expansion with neuron pruning and masking techniques to address the challenges of forgetting, model compactness, and achieving higher fidelity in knowledge reuse. We adopt these knowledge reusable continual learning approaches to retain and extract valuable insights from previous tasks, while aiming to find useful knowledge (for reuse) based on model similarity. Although some works have applied some similarity metrics such as cosine similarity to federated learning scenarios [20], it is limited in applicability when dealing with clients having a different number of outputs (classes). In contrast, our approach provides flexibility in output heterogeneity, ensuring broader applicability in real-world scenarios, where most clients may have different shapes of outputs. Moreover, our decision pattern-based similarity is novel in federated continual learning, as the knowledge itself is not exactly the same even with the similar decision patterns. This enables private learners to be exposed to similar yet unseen broader knowledge, leading to improved performance and efficiency in subsequent learning tasks.

Federated continual learning. With widespread accessibility of edge AI, learners do not necessarily be confined to their own private knowledge, as a standalone learner is permitted to only access its own local information and thus suffer from restricted experiences to some extent. Although recent works tackle the problem of federated continual learning [50, 61], they focus on class-incremental learning, whereas we tackle task-incremental learning. In task-incremental learning, transfer learning turns out to be very effective, where a model is pretrained based on the source domain and then finetuned for the target domain. This selective device-to-device knowledge transfer can be a significant impetus to enhance model fidelity, also in continual learning. It has been commonly addressed in the field of federated continual learning but only in a centralized manner [18, 50]. It has been also observed that there is both positive and negative knowledge related to a specific target task [43, 57], highlighting the potential harm of static knowledge selection. To address the two-sided nature of knowledge reuse, *FedWeIT* [57] aggregates the parameters from clients as global knowledge, and then the clients mask the federated parameters adaptive to the specific task. However, previous works including *FedWeIT* rely on a central server, causing communication bottlenecks and privacy limitations. Furthermore, ImageNet-pretrained models have widely been utilized in transfer learning. However, it is not feasible for edge-side federated continual learning, as pretrained models are specifically designed for well-known large-scale architectures. The architectures in continual learning are dynamically changing, and edge AI cannot be tailored to large-scale models. In *Pick-a-back*, we rather explore decentralized, communication-efficient, and privacy-preserving knowledge transfer in lightweight on-device models.

3 Selective Knowledge Transfer

We consider the problem of device-to-device knowledge federation to enhance individual continual learning in a distributed manner. We propose *Pick-a-back*, a simple yet effective knowledge federation approach by discovering a *relevant* neighboring device who has acquired the knowledge of *relevant* tasks and then selectively infusing it into the model via continual learning. By leveraging a federated learning framework, which allows collaborative learning across devices, we aim to address the challenge of distributed continual learning, while also facilitating the reuse of previously acquired knowledge at each different device. The integration of these two fields of study enables efficient knowledge selection, sharing, and expansion among devices (with heterogeneous data), yielding continuous learning improvement and adaptation across tasks or learning scenarios.

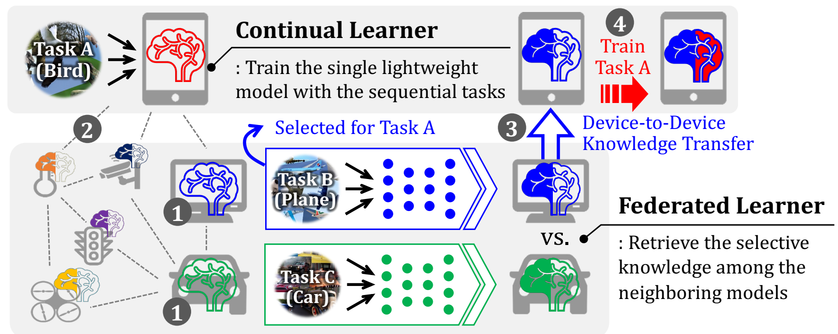


Fig. 1: Overall architecture of *Pick-a-back*: (1) Each device trains an individual continual learning model on a series of tasks. (2) Given these available neighboring knowledge, a device selects a relevant neighboring model that can improve learning performance by acquiring indirect knowledge from the federated but physically separated devices. (3) On top of the retrieved knowledge, (4) the continual learner stacks new local tasks and resumes learning and federation. By fetching the generalized knowledge with relevant decision patterns, *Pick-a-back* offers a cumulative yet selective knowledge federation with heterogeneous data at a distinct device

3.1 Problem Formulation

Given L devices, each learner $l \in \{1, 2, \cdots, L\}$ at device d_l processes its own sequential tasks (i.e., training data) of $T^{(l)} = \{T_1^{(l)}, T_2^{(l)}, \cdots\}$. A scratch model $M_i^{(l)}$ is constructed by training only with private data $T_i^{(l)} = \{(X_i^{(l)}, Y_i^{(l)})\}$, where $T_i^{(l)}$ for task i and device d_l is heterogeneous across other tasks and devices. We train the model $\hat{M}_i^{(l)}$ with target data $T_i^{(l)}$ on top of its previous version $\hat{M}_{i-1}^{(l)}$ with some learning parameters $\theta_i^{(l)}$, i.e., $\hat{M}_i^{(l)} = \hat{M}_{i-1}^{(l)}(T_i^{(l)}, \theta_i^{(l)})$ in a continual manner. Here, at every task i, the goal is to find a good reference model that can maximize the overall fidelity of model $\hat{M}_i^{(l)}$.

3.2 Architecture

Our learning architecture incorporates two distinct modes (Fig. 1): i) continual learner for local training; and ii) federated learner for knowledge transfer. Initially, multiple learners train independently on a specific private task. Subsequently, one learner at a device transitions to a federated learning mode to explore effective indirect knowledge from a neighboring relevant model. Once a neighboring model is selected, we use the selected model as a backbone, the main target task is then executed in a continual manner. By incorporating the retrieved knowledge together with its existing own knowledge, the learner can construct cumulative knowledge in a continual manner without forgetting. This process of continual learning and federated learning is repeated for each task.

Continual learner. In our approach to continual learning, we adopt task-incremental learning, where a series of tasks is processed sequentially. Each task has a distinct learning objective, and a local model is trained independently on each task with freezing or isolating the models for previous knowledge. By employing task-oriented learning, we aim to mitigate the catastrophic forgetting problem, where previously learned knowledge vanishes as training on new tasks or incorporating new data over time. Our approach can be combined with any continual learner that has gradually been constructing knowledge. It is fundamentally different from conventional continual learning that simply memorizes multiple models or data, which do not leverage the given knowledge. We suppose that a continual learner concatenates a new model on top of the previously acquired knowledge, and thus it can reduce memory usage and also enhance the model fidelity. This sustainability of the continual model is particularly valuable in the context of edge AI and aligns with the goals of the recent learning paradigm.

We pursue reusing the pre-configured knowledge by using the model. We mainly use the latest continual learning effort, *CPG* [19], which expands the model when necessary, prunes it for compactness, and reuses the network by masking out specific ineffective neurons. Inspired by the masking strategy of *Piggyback* [29], it samples some necessary neurons for the target task to allow the extraction of common knowledge. *CPG* only trains the learnable binary mask whether to activate each neuron for the target task. As it ensures the retention of previous knowledge via freezing, the results remain exactly the same even when training new tasks. However, without loss of generality, this masking strategy can be applied to any continual learner that holds multiple knowledge.

Federated learner. Due to the privacy and communication overhead, distributed learners are unwilling to share private raw data. We thus adopt a model-sharing approach rather than data-sharing. Here, each client is regarded as an independent learner. When each learner at a distinct device is working on a distinct task but lacks knowledge, there might be some other knowledge that has a similar, broader, or deeper experience beyond the learner based on its own local data. We assume that clients are knowledgeable with their own currently available neighboring models via existing neighbor selection methods [38, 53] or graph construction approaches [33, 62]. These neighboring diverse knowledge sources have been trained on different datasets, capturing various perspectives and feature extraction capabilities. By exposing the learner to external knowledge, the model is capable of leveraging enhanced feature representations, enabling it to capture a broader set of patterns and information that may be beneficial for its specific learning objective. The learner can also develop a more comprehensive understanding of the data, generalize better to unseen examples by mitigating overfitting, and thus improve the fidelity of the model on a variety of tasks and data. By leveraging a diverse ensemble of knowledge sources, the learner can benefit from its complementary expertise, leading to accuracy improvement.

Each learner at a device seeks relevant knowledge that can contribute to further improving its performance based on local continual learning. To acquire necessary constructive knowledge, we identify effective indirect experiences based on decision pattern similarities. Once we have obtained knowledge from other learners, we accumulate ongoing new local tasks on top of the model in a continual manner, ensuring that the retrieved knowledge becomes infused with local knowledge. Our main focus is on the current target task, and thus, if we want to retain the knowledge from our previous tasks, we have to either store the existing knowledge separately or concatenate it with the fetched model. If the currently constructed knowledge turns out to be sufficiently effective, there may be no further need to incorporate further knowledge from other learners.

3.3 Selective Knowledge Federation

Each learner is responsible for distinct tasks, and tasks may exhibit unique strengths, weaknesses, or biases. Consequently, each task at a local learner plays its own unique different role and forms distinct knowledge. Some external models may have learned other representations that have captured specialized knowledge relevant to specific tasks. Therefore, selective knowledge transfer can be a very effective way for enhancing the performance of each student model at an edge device. Our approach is simple and straightforward – if models have similar or synergetic capabilities in classifying different samples, they are likely to have potential sources of common but more generalized knowledge. Since it has the possibility to attain broader intuition but also inexperienced knowledge, this indirect knowledge characterized by similar decision patterns may greatly accelerate and improve the learning for individual learners.

As the similarity of model do not straightly incur similar knowledge, we focus on the decision pattern similarity rather than the model itself. We borrow an idea to calculate the model similarity from *ModelDiff* [27], which calculates the similarity of decision boundaries by comparing the reaction to the input data. Initially, we sample input data x_i from the requester and adversarial data x'_i from a neighbor. To address privacy concerns, an alternative approach may sample data only from the requester, or the participants may opt to exchange mutated data, thus avoiding the sharing of private data. Then, we inject a pair-wise set of input data, $\{(x_1, x'_1), (x_2, x'_2), \cdots, (x_n, x'_n)\}$, into each model f and g and then computes the decision distance vector (DDV), $DDV_f = \{d_1, d_2, \cdots, d_n\}$, which is a distance vector for the output pair, where $d_i = distance(f(x_i), f(x'_i)), i \in \{1, 2, \ldots, n\}$. Finally, it compares the cosine similarity between the distance vectors of models, $sim(DDV_f, DDV_g)$. Since it focuses the decision similarity instead of directly comparing the outputs, it is applicable to any heterogeneous task with a different number of output classes. To search a pair-wise set that can approximate the decision boundary most effectively, the input data is mutated, and similarities are iteratively computed, similar to genetic algorithms.

By incorporating it in *Pick-a-back*, when we search for a similar model for task $T_i^{(l)}$ of learner l among $\hat{M}_{i-1}^k$ of all neighboring learner k. We sample an input pair-wise data $X_i^{(l+k)}$ and $X_i'^{(l+k)}$ from $X_i^{(l)} \cup X_{i-1}^{(k)}$ (or just from $X_i^{(l)}$). By injecting $X_i^{(l+k)}$ and $X_i'^{(l+k)}$ into the learner model $M_i^{(l)}$ and the neighboring models $\hat{M}_{i-1}^{(k)}$, we produce the distance vector of outputs $D_{M_i^{(l)}}$ and $D_{\hat{M}_{i-1}^{(k)}}$. Then,

Algorithm 1 Federated Continual Learning with *Pick-a-back*

Input: L: # of devices, (for each device d_l) $M_0^{(l)}$: random initial model, $T^{(d_l)}$: a sequence of tasks, $\theta_i^{(l)}$: learning parameters
for task $i = 1 : length(T^{(l)})$ **do**
  Device d_l trains task i such that $M_i^{(l)} = M_{i-1}^{(l)}(T_i^{(l)}, \theta_i^{(l)})$; ▷ Continual Learner
  for device $k = 1 : L$ **do** ▷ Federated Learner
    Compute the similarity, $sim(M_i^{(l)}, \hat{M}_{i-1}^{(k)})$;
  end for
  Select and retrieve knowledge $\hat{M}_{i-1}^{(l)} \leftarrow \hat{M}_{i-1}^{(k)}$ *such that* $\text{argmax}_k\, sim(M_i^{(l)}, \hat{M}_{i-1}^{(k)})$;
  Train task i on the fetched model, $\hat{M}_i^{(l)} = \hat{M}_{i-1}^{(l)}(T_i^{(l)}, \theta_i^{(l)})$; ▷ Continual Learner
end for

we select the most similar model as follows:

$$\underset{k}{\text{argmax}}\, sim(DDV_{M_i^{(l)}}, DDV_{\hat{M}_{i-1}^{(k)}}) \tag{1}$$

We fetch the indirect knowledge that has already experienced the target decision pattern. After selecting the exterior knowledge, we continue to learn T_i^l on $\hat{M}_{i-1}^{(k)}$.

Training process. Each learner l trains i-th task $T_i^{(l)}$ and constructs a local model $M_i^{(l)}$ with continual learning. Then, the learner l checks if there exists any helpful exterior knowledge by comparing $M_i^{(l)}$ with $M_{i-1}^{(k)}$ for all neighboring edge learners k. After selecting relevant knowledge for knowledge transfer, device d_l processes the i-th task $T_i^{(l)}$ over $\hat{M}_{i-1}^{(k)}$ and constructs $\hat{M}_i^{(l)} = \hat{M}_{i-1}^{(k)}(T_i^{(l)}, \theta_i^{(l)})$.

The overall procedure is described in Algorithm 1.

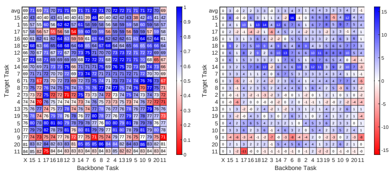
4 Experiments

We have validated *Pick-a-back*[1] extensively on four different classification task groups with 13 public data using four different well-known architectures. Each group is considered as one experimental environment where knowledge can be shared, and one superclass group is a single task.

Datasets. We used four different groups of CIFAR-100 [25], Classified-ImageNet [8], Fine-Grained [10, 24, 37, 48, 55], and Digits [7, 11, 35, 36, 39, 45]. The organization of the dataset and the details are provided in Appendix Sec. A1.

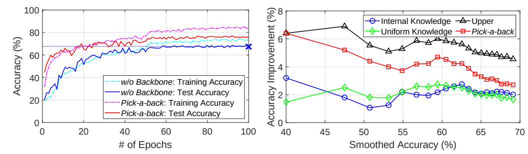
Architectures. We have presented the results using both lightweight models and large-scale models. For the lightweight models, we used LeNet-5 [26] and MobileNetV1 [17], where ImageNet-pretrained models are not publicly available. On the other hand, VGG16-BN [51] and EfficientNet-B0 [54] that have publicly available ImageNet-pretrained models were implemented using *PyTorch* [42]. Details of model configurations and learning parameters are in Appendix Sec. A2. We used CIFAR-100 on LeNet-5 as default, unless otherwise noted.

[1] Our codes are available at `https://github.com/jinyi-yoon/Pick-a-back.git`.



(a) Absolute Acc. **(b)** Improved Acc.

Fig. 2: Heatmap of the accuracy for various backbone and target tasks sorted by the accuracy of *w/o Backbone*. The first row represents the averaged values for all tasks, while the first column shows the results of *w/o Backbone*. Absolute accuracy is scaled per each row, and accuracy improvement is scaled across all results



(a) Training dynamics **(b)** Accuracy improvement

Fig. 3: Training dynamics for the most effective case of the target task 14 using the backbone task of 4 and accuracy improvement with respect to the smoothed accuracy of *w/o Backbone*. The smoothed accuracy refers to the averaged accuracy for a subset of tasks with lower accuracy. For example, the second x-value represents the average of two tasks with the lowest performance, while the last value is the average across all tasks

Baselines. We compare *Pick-a-back* with diverse learning approaches of federated learning, continual learning, and continual federated learning.

First, we let our *Pick-a-back* run on top of a certain continual learning scheme: 1) *ProgressiveNet* [47]: continual training only with expansion by concatenating scratch models for sequential tasks; 2) *PackNet* [30]: continual training only with pruning by allocating partial neurons exclusively to each task; 3) *CPG* [19]: continual training with expansion if necessary and pruning for the model compactness; and 4) *WSN* [21]: continual learning with reusing the masked weights of the prior subnetworks.

We further compare across existing federated and continual federated learning approaches: 1) *FedAVG* [32]: one of the pioneering works for model regularization in federated learning; 2) *FedWeIT* [57]: a federated continual learning paradigm by introducing sparse task-specific parameters along with global federated parameters; and 3) *FCCL* [18]: a federated continual learning approach to tackle the catastrophic forgetting problem within a heterogeneous setting.

We also compare *Pick-a-back* with different knowledge selection strategies: 1) *w/o Backbone*: intact continual learning without using any selective knowledge transfer, such as *Progressive Networks*, *PackNet*, or *CPG*; 2) *Uniform Knowledge*: applying the same task to its backbone model and averaging all of the results for each task; 3) *Internal Knowledge*: training the target task iteratively as a backbone model; 4) *Random Knowledge*: randomly selecting a backbone model; and 5) *Upper*: selecting an ideal backbone with a certain task that shows the highest accuracy improvement, being served as an unachievable upper bound.

Measures. We have investigated 1) **Avg**: the averaged accuracy of all tasks and 2) **Top**: the accuracy of the most effective task that has gained the best accuracy improvement via selective knowledge. To compare the feasibility of models for edge AI, we have shown 1) Model size: the actual storage overhead of a model; 2) Expansion: the expansion degree where the scratch model has the value of 1;

Table 1: Performance comparison. The bold text highlights the best performance per client, excluding *Upper* as an ideal upper-bound due to its infeasible brute-force manner

Client ID	15	1	17	16	18	12	3	14	7	6	8	2	4	13	19	5	10	9	20	11	Avg
w/o Backbone	40.0	55.0	57.2	60.4	51.8	66.0	67.4	67.6	69.2	70.8	72.6	**73.0**	**74.2**	75.2	75.8	76.4	77.0	77.4	81.2	84.0	**69.1**
FedAVG	24.4	31.6	34.6	38.0	32.0	39.6	31.8	32.0	49.0	37.8	29.8	34.0	26.4	30.8	35.8	45.4	27.2	23.4	35.6	39.0	**33.9**
ProgressiveNet	44.6	58.8	54.4	63.0	64.4	68.0	71.2	68.2	69.2	70.6	73.8	72.4	70.4	73.0	74.0	78.8	77.2	73.6	**84.6**	83.8	**69.7**
PackNet	38.0	55.0	56.4	59.4	63.8	69.0	68.2	72.2	71.2	72.8	74.4	72.0	70.2	74.8	**79.6**	78.0	78.6	76.8	82.6	80.2	**69.7**
WSN	34.2	53.8	49.4	53.4	59.8	60.2	63.6	63.0	66.8	65.0	64.6	65.4	66.6	62.6	61.2	67.6	66.6	58.6	74.0	79.8	**61.8**
FedWeIT	40.0	52.5	55.0	56.0	56.2	60.5	56.7	60.5	67.0	65.2	60.2	65.0	66.0	64.2	65.0	65.7	66.7	51.0	68.7	77.8	**61.0**
FCCL	37.0	54.4	52.4	60.6	62.2	68.2	67.2	64.4	68.2	70.6	72.8	72.4	68.4	**77.0**	73.8	75.0	78.6	68.8	79.2	78.2	**67.5**
Uniform	41.5	58.6	57.6	62.1	65.6	70.7	69.7	71.6	71.2	72.5	74.7	**73.0**	73.7	76.4	77.1	78.7	79.4	75.2	83.5	83.0	**70.8**
Internal	43.2	55.4	56.8	62.2	**67.8**	67.0	69.0	71.4	**73.8**	**74.8**	76.8	71.8	73.4	76.2	79.2	77.6	**81.0**	**79.2**	82.0	83.8	**71.1**
Random	39.8	**59.0**	56.6	59.2	67.6	**72.6**	68.2	73.0	71.4	71.4	75.6	72.8	74.4	76.4	78.0	78.2	77.8	74.6	84.2	83.8	**70.7**
Upper	46.4	62.4	60.0	64.2	67.8	74.8	72.2	75.6	73.8	75.6	77.0	75.0	76.0	79.4	80.0	80.6	81.6	79.2	86.2	85.4	**73.7**
Pick-a-back	**46.4**	**59.0**	**60.0**	**63.2**	64.4	**72.6**	**71.8**	**75.6**	72.4	72.2	**77.0**	**73.0**	72.8	76.2	76.0	**80.2**	78.2	75.8	84.2	**85.4**	71.8

3) Sparsity: the released ratio of a model, where higher sparsity indicates more space for next tasks; 4) Training time: the execution time to run for one epoch; and 5) Used data: the number of data samples used for training.

4.1 Overall Performance and Comparisons

Motivations for selective knowledge federation. We first examined the impact of different knowledge on various target tasks. As in Fig. 2a, a significant portion of the results and the average accuracy achieved through backbone usage are represented by a blue color, indicating that the continual learner benefited from the knowledge federation. However, there exist not only positive backbones but also negative backbones, suggesting a need for selective knowledge to assist a continual learner specific to its target task. Taking a closer look at the accuracy improvement relative to *w/o Backbone* in Fig. 2b, a well-constructed model such as task 9 or 11 is more likely disturbed by external knowledge. On the other hand, a target task with initially poor performance tends to gain a greater improvement in accuracy. It implies the importance of selectively incorporating constructive knowledge to enhance the learning process for a specific target task.

Ablation study on selective knowledge federation. Inspired by the Fig. 2, we verified the effectiveness, compared to diverse learning approaches, including *FedAVG*, *ProgressiveNet*, *PackNet*, *WSN*, *FedWeIT*, and *FCCL*, along with various backbone selection strategies of *Uniform*, *Internal*, *Random*, and *Upper*, as in Table 1 and in Table A5. We consistently demonstrate superior fidelity on average. By employing our selection approach, we observed a significant accuracy improvement of up to 8.0%, with only two negative cases out of 20 tasks. Since *FedAVG* simply averages the model parameters without considering the continual tasks, it suffers from losing the knowledge of trained tasks. Notably, although *Pick-a-back* with *CPG* shows the best performance, *Pick-a-back* with other continual learning approaches of *ProgressiveNet* and *PackNet* also outperforms *w/o Backbone*. Even *Uniform*, *Internal*, and *Random* selection strategies exhibit enhanced performance, suggesting the efficacy of inter-device knowledge transfer based on *Pick-a-back* primitive. We also compared with other recent (federated) continual learning approaches (*e.g.*, *WSN*, *FedWeIT*, and *FCCL*), showing that *Pick-a-back* outperforms these methods. This is due to the effectiveness of our knowledge reuse approach along with our backbone selection strategy.

Benefit of backbone knowledge. Taking a closer look at how *Pick-a-back* benefits from backbone knowledge, in Fig. 3a, it is evident that the test accuracy of *Pick-a-back* consistently surpasses the training accuracy of *w/o Backbone*, starting from the beginning until the saturation. It indicates that indirect knowledge assists individual learners to be exposed to inexperienced knowledge and adapt to unseen data. Even within *Pick-a-back* itself, the test accuracy outperforms the training accuracy in the early stages of training, suggesting that the transferred knowledge has relevance to the target knowledge and drives toward high performance even earlier. Furthermore, *w/o Backbone* achieves an accuracy of 67.4%, which can be attained only within approximately 20 epochs using *Pick-a-back*, whereas it takes over 50 epochs without knowledge transfer. This demonstrates that *Pick-a-back* not only accelerates the learning, but also enhances the overall performance. Please refer to the empirical convergence analysis in Appendix Fig. A1.

Possibility of similar but generalized knowledge. To justify the task-specific selection, we compared to uniform knowledge, where the same backbone is applied to all tasks. As in Fig. 3b, the results showed that *Upper* exhibits better improvement in tasks with poor initial performance. However, the uniform knowledge ignores task-specific improvements, leading to similar improvements across all target tasks. Interestingly, we also observed that only using internal knowledge, where the target task is trained twice, allowed further improvement. It means that even with a single task iteratively training by accumulating knowledge on top of an identical model turns out to induce better knowledge. It also implies that a backbone task with a similar, ideally the same in personal knowledge case, the decision boundary can further improve the knowledge. While the same task aims for identical knowledge, using similar but distinct knowledge enables generalization and better improvement beyond individual tasks. As expected, *Pick-a-back* outperformed the one only with internal knowledge, highlighting the advantage of leveraging similar indirect experiences. This means that we provide a more opportunity to acquire superior knowledge by mitigating the overfitting problem and extending to the unseen data beyond local knowledge.

4.2 Scalability Analysis

Lightweight models for edge AI. We evaluated *Pick-a-back* in various architectures from the lightweight models of LeNet-5, MobileNet-V1, and EfficientNet-B0[2]

[2] For EfficientNet-B0, we enlarge the images to 224×224 pixels for evaluation, but a model size is measured with the original size of 32×32 for a fair comparison.

Table 2: Accuracy, model size, and averaged training time per one epoch with respect to different architectures on CIFAR-100. We averaged the training time over 10 epochs

		Accuracy		Model Size	Training Time
		Avg	Top		
VGG16-BN	**w/o Backbone**	78.9	68.2	164 MB	4.10 s
	Pick-a-back	81.0 (+2.1)	76.6 (+8.4)		
EfficientNet-B0	**w/o Backbone**	77.8	44.6	20 MB	9.29 s
	Pick-a-back	81.8 (+4.0)	54.4 (+9.8)		
MobileNet-V1	**w/o Backbone**	65.4	59.8	16 MB	4.95 s
	Pick-a-back	66.7 (+1.3)	65.4 (+5.6)		
LeNet-5	**w/o Backbone**	69.1	67.6	**0.3 MB**	**2.18 s**
	Pick-a-back	71.8 (+2.7)	75.6 (+8.0)		

Table 3: Various datasets with LeNet-5

	Digits		CIFAR-100		Classified-ImageNet		Fine-Grained	
	Avg	Top	Avg	Top	Avg	Top	Avg	Top
Upper	95.8	89.4	73.7	75.6	61.4	71.8	33.9	48.5
w/o Backbone	93.8	80.7	69.1	67.6	56.1	59.2	25.0	21.0
Pick-a-back	**95.1 (+1.3)**	**86.4 (+5.7)**	**71.8 (+2.7)**	**75.6 (+8.0)**	**58.1 (+2.0)**	**67.6 (+8.4)**	**28.3 (+3.3)**	**27.5 (+6.5)**

to the large-scale models of VGG16-BN in Table 2. It indicates that all models benefit from selectively transferred knowledge, leading to accuracy improvement of up to 8.4%, 9.8%, 5.6%, and 8.0%, respectively. However, in resource-constrained environments, the limitations of memory and processing power make it challenging to deploy a large-scale model at the edge. To make learning practical in the real-world, lightweight models can be flexibly deployed across multiple edge devices, reducing dependence on remote servers. The lightweight LeNet-5 has a much smaller size of 0.18 times compared to VGG16-BN, enabling faster training and inference time. Although the lightweight models may not match closely to the large-scale models, *Pick-a-back* can still achieve improvement, comparable to the large models, while greatly reducing the storage requirement and run-time.

Scalability across diverse datasets. We assessed the scalability of *Pick-a-back* across various datasets in Table 3. We stably achieve improvements across different datasets, even with varying degrees of class complexity. In Digits, where most tasks except for the *Roman* task are generally easy to classify, the average accuracy improvements may appear modest, but *Pick-a-back* approach achieved a significant accuracy improvement of up to 5.72% in the relatively challenging *Roman* task. Similarly, in the classified-ImageNet dataset, where there exist enough neighboring tasks to provide additional knowledge, *Pick-a-back* shows substantial improvements even for the underperforming tasks. However, when considering fine-grained datasets with a large number of classes per task (ranging 102∼250 classes), classification becomes inherently more difficult with simple models, resulting in relatively lower accuracy. Nonetheless, by applying *Pick-*

Table 4: Performance on Fine-Grained datasets using EfficientNet-B0

Client ID	1	2	3	4	5	Avg
Pretrained	28.9	**90.5**	42.4	**49.9**	37.1	**49.8**
w/o Backbone	8.3	66.3	7.4	28.2	12.7	**24.6**
Pick-a-back	**49.5** (**+41.2**)	77.4 (+11.1)	**44.0** (**+36.6**)	46.4 (+18.2)	**64.6** (**+51.9**)	<u>**56.4**</u> (<u>**+31.8**</u>)

Table 5: Heterogeneous data sampling with $\alpha = 0.5$

	Hetero		Balanced		w/o Sampling	
	Avg	**Top**	**Avg**	**Top**	**Avg**	**Top**
Upper	59.3	60.2	69.6	84.4	73.7	75.6
w/o Backbone	52.6	40.2	64.1	71.8	69.1	67.6
Pick-a-back	56.1 (+3.5)	58.6 (<u>**+18.4**</u>)	67.3 (+3.2)	80.0 (+8.2)	71.8 (+2.7)	75.6 (+8.0)

a-back, it is possible to advance the knowledge and achieve performance gains. While there is a noticeable performance gap of 0.7%, 1.9%, and 3.3%, respectively, between *Pick-a-back* and *Upper* in other datasets, the gap in the fine-grained dataset is quite large with 5.6% on average and up to 21.0% for the task of *flowers*. This implies the potential of *Pick-a-back* to drive further improvements in complex tasks that require more sophisticated classification. We further explore the performance with the different number of clients (i.e., classes) and the different learning seeds in Appendix Table A9 and in Appendix Table A10, respectively.

Impact on the most complicated task. We investigate the most complicated tasks of CUB [55], Stanford Cars [24], Flower [37], WikiArt [48], and Sketch [10], with each featuring a diverse range of class numbers of 200, 196, 102, 195, and 250 and running at each separate client. As shown in Table 4, with the adoption of *Pick-a-back*, we observe remarkable improvements in overall accuracy, averaging at 31.8%, and peaking at an impressive 51.9%. Although these results might appear surprising, they align with the findings previously established by [19] and [29], where scratch accuracy on ImageNet-pretrained models of ResNet50 and VGG16-BN, achieved 58.83% and 43.84%, respectively. Incorporating *Pick-a-back* not only surpasses the average accuracy of ImageNet-pretrained models, but does so through a judicious infusion of tailored knowledge. It is worth noting that ImageNet-pretrained models often necessitate extensive prior training data. However, in many instances, the efficacy diminishes due to the lack of distinct knowledge selection, showing some critical drawbacks in clients #1, #3, and #5.

Superiority in heterogeneous data scenarios. In real-world scenarios, there are many unique challenges such as personalization [5] or heterogeneity [4]. Here, we explored the effectiveness of *Pick-a-back* under data heterogeneity. Leveraging the heterogeneity settings from federated learning researches [16, 60], we applied α of 0.5 based on the Dirichlet distribution $Dir(\alpha)$. As in Table 5, we observed a degradation in the accuracy of scratch models as the data heterogeneity increases. It implies that the accuracy is achievable, and the external knowledge can be particularly beneficial for weaker classes in the presence of heterogeneous data. While the average accuracy improvement remained consistent, we achieved significant enhancements in heterogeneous data settings. It is essential to address unbalanced data distributions, as certain classes may receive less training attention. By leveraging *Pick-a-back*, we are able to achieve substantial performance gains, with improvements of up to 18.4%.

Table 6: Performance and data used for training in ImageNet-pretrained VGG16-BN

	w/o Backbone → Pick-a-back		**Size of Training Data**
	Avg	**Top**	
w/ Pretrained	84.9 → 83.8	82.8 → 84.4	92.8 GB
w/o Pretrained	78.9 → 81.0	68.2 → 76.6	**0.3 GB**

4.3 Discussions and Limitations

ImageNet-pretrained models. It is important to note that a lightweight model such as LeNet-5 may not outperform a large-scale model even with *Pick-a-back*. Also, VGG16-BN models initialized with ImageNet-pretrained weights demonstrate strong performance generally for visual tasks as in Table 6 and in Appendix Table A6. However, their effectiveness heavily relies on the availability of large-scale labeled datasets that need to be uploaded to a central server. Additionally, the pretrained models are specifically designed for image classification, and their performance may vary when applied to specific or specialized tasks. Furthermore, the availability of pretrained models is limited only to well-known large-scale architectures, making their deployment usually infeasible to edge learning. Without pretrained model, selective knowledge federation through *Pick-a-back* achieves similar performance on average with a small gap of 3.9% compared to the original pretrained model (*w/o Backbone*), but only with 0.3% of training data.

We further provided the orthogonality of continual learner in *Pick-a-back* and catastrophic forgetting prevention across different continual learning approaches in Appendix Table A11. Lastly, we need image sharing among clients and model sharing from a selected neighboring client, which may incur transmission overhead and privacy issues. We thus further assess the communication and computation time and how we can defend against attacks in Appendix Table A12 and Fig. A2.

5 Conclusion

We propose *Pick-a-back* that picks a task-specific backbone knowledge and fosters the collective intelligence in federated continual learning. As the individual learner pursues each distinct goal with private heterogeneous data, we selectively transfer the relevant external knowledge to the target task between neighboring learners.

As AI expands towards edges for scalability, low-end devices tend to be trained toward personalized and specialized but somewhat partially limited intelligence. Here, in case of a scarcity of data or with limited performance, it becomes essential to seek assistance from external sources. The question – *how can a device make a local model more effectively trained with exterior knowledge?* – can be answered: *Pick-a-back* offers an effective way to selective knowledge transfer from one device to another towards distributed edge intelligence. By harnessing the power of neighboring devices, individual learners can gain access to valuable insights and expertise via collective intelligence, augmenting their own capabilities.

Acknowledgments

We thank the Virginia Tech Presidential Postdoctoral Fellowship Program for supporting JinYi Yoon's postdoctoral associate position. This work was supported by the Institute of Information & communications Technology Planning and Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2022-00155966, Artificial Intelligence Convergence Innovation Human Resources Development (Ewha Womans University) and No. RS-2021-II212068, Artificial Intelligence Innovation Hub).

References

1. Arani, E., et al.: Learning fast, learning slow: A general continual learning method based on complementary learning system. ICLR (2022)
2. Cao, L.: Decentralized ai: Edge intelligence and smart blockchain, metaverse, web3, and desci. IEEE Intelligent Systems **37**(3), 6–19 (2022)
3. Casado, F.E., Lema, D., Criado, M.F., Iglesias, R., Regueiro, C.V., Barro, S.: Concept drift detection and adaptation for federated and continual learning. Multimedia Tools and Applications pp. 1–23 (2022)
4. Chen, D., Gao, D., Kuang, W., Li, Y., Ding, B.: pfl-bench: A comprehensive benchmark for personalized federated learning. Advances in Neural Information Processing Systems **35**, 9344–9360 (2022)
5. Chen, D., Gao, D., Xie, Y., Pan, X., Li, Z., Li, Y., Ding, B., Zhou, J.: Fs-real: Towards real-world cross-device federated learning. In: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 3829–3841 (2023)
6. Chen, Y., Inaltekin, H., Gorlatova, M.: Adaptslam: Edge-assisted adaptive slam with resource constraints via uncertainty minimization. arXiv preprint arXiv:2301.04620 (2023)
7. Colianni, S.: Mnist as .jpg (2017), https://www.kaggle.com/scolianni/mnistasjpg
8. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 248–255 (2009). https://doi.org/10.1109/CVPR.2009.5206848
9. Dhar, P., Singh, R.V., Peng, K.C., Wu, Z., Chellappa, R.: Learning without memorizing. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5138–5146 (2019)
10. Eitz, M., Hays, J., Alexa, M.: How do humans sketch objects? ACM Transactions on graphics (TOG) **31**(4), 1–10 (2012)
11. El-Sawy, A., El-Bakry, H., Loey, M.: Cnn for handwritten arabic digits recognition based on lenet-5. In: Proceedings of the International Conference on Advanced Intelligent Systems and Informatics 2016 2. pp. 566–575. Springer (2017)
12. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016), http://www.deeplearningbook.org
13. Gu, Y., Yang, X., Wei, K., Deng, C.: Not just selection, but exploration: Online class-incremental continual learning via dual view consistency. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7442–7451 (2022)

14. Ha, D., Tang, Y.: Collective intelligence for deep learning: A survey of recent developments. Collective Intelligence **1**(1), 26339137221114874 (2022)
15. Haim, N., Vardi, G., Yehudai, G., Shamir, O., Irani, M.: Reconstructing training data from trained neural networks. Advances in Neural Information Processing Systems **35**, 22911–22924 (2022)
16. He, C., Li, S., So, J., Zeng, X., Zhang, M., Wang, H., Wang, X., Vepakomma, P., Singh, A., Qiu, H., et al.: Fedml: A research library and benchmark for federated machine learning. arXiv preprint arXiv:2007.13518 (2020)
17. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)
18. Huang, W., Ye, M., Du, B.: Learn from others and be yourself in heterogeneous federated learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10143–10153 (2022)
19. Hung, C.Y., Tu, C.H., Wu, C.E., Chen, C.H., Chan, Y.M., Chen, C.S.: Compacting, picking and growing for unforgetting continual learning. Advances in Neural Information Processing Systems **32** (2019)
20. Jeong, W., Hwang, S.J.: Factorized-fl: Personalized federated learning with parameter factorization & similarity matching. Advances in Neural Information Processing Systems **35**, 35684–35695 (2022)
21. Kang, H., Mina, R.J.L., Madjid, S.R.H., Yoon, J., Hasegawa-Johnson, M., Hwang, S.J., Yoo, C.D.: Forget-free continual learning with winning subnetworks. In: International Conference on Machine Learning. pp. 10734–10750. PMLR (2022)
22. Kim, G., Esmaeilpour, S., Xiao, C., Liu, B.: Continual learning based on ood detection and task masking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3856–3866 (2022)
23. Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A.A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al.: Overcoming catastrophic forgetting in neural networks. Proceedings of the national academy of sciences **114**(13), 3521–3526 (2017)
24. Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3d object representations for fine-grained categorization. In: Proceedings of the IEEE international conference on computer vision workshops. pp. 554–561 (2013)
25. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
26. LeCun, Y., et al.: Lenet-5, convolutional neural networks. URL: http://yann. lecun. com/exdb/lenet **20**(5), 14 (2015)
27. Li, Y., Zhang, Z., Liu, B., Yang, Z., Liu, Y.: Modeldiff: testing-based dnn similarity comparison for model reuse detection. In: Proceedings of the 30th ACM SIGSOFT International Symposium on Software Testing and Analysis. pp. 139–151 (2021)
28. Liu, F., Wu, X., Ge, S., Fan, W., Zou, Y.: Federated learning for vision-and-language grounding problems. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 11572–11579 (2020)
29. Mallya, A., Davis, D., Lazebnik, S.: Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 67–82 (2018)
30. Mallya, A., Lazebnik, S.: Packnet: Adding multiple tasks to a single network by iterative pruning. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 7765–7773 (2018)
31. Mao, B., Liu, J., Wu, Y., Kato, N.: Security and privacy on 6g network edge: A survey. IEEE Communications Surveys & Tutorials (2023)

32. McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Artificial intelligence and statistics. pp. 1273–1282. PMLR (2017)
33. Meng, C., Tang, M., Setayesh, M., Wong, V.W.: Gnn-based neighbor selection and resource allocation for decentralized federated learning. In: GLOBECOM 2023-2023 IEEE Global Communications Conference. pp. 1223–1228. IEEE (2023)
34. Miller, G.A.: Wordnet: a lexical database for english. Communications of the ACM **38**(11), 39–41 (1995)
35. Nazarpour, K., Chen, M.: Handwritten Chinese Numbers (1 2017). `https://doi.org/10.17634/137930-3`, `https://data.ncl.ac.uk/articles/dataset/Handwritten_Chinese_Numbers/10280831`
36. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning (2011)
37. Nilsback, M.E., Zisserman, A.: Automated flower classification over a large number of classes. In: 2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing. pp. 722–729. IEEE (2008)
38. Nishio, T., Yonetani, R.: Client selection for federated learning with heterogeneous resources in mobile edge. In: ICC 2019-2019 IEEE international conference on communications (ICC). pp. 1–7. IEEE (2019)
39. Noyan, M.A.: Hand written roman numerals dataset (Sep 2021). `https://doi.org/10.5281/zenodo.5385144`, `https://doi.org/10.5281/zenodo.5385144`
40. Ostapenko, O., Puscas, M., Klein, T., Jahnichen, P., Nabi, M.: Learning to remember: A synaptic plasticity driven framework for continual learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11321–11329 (2019)
41. Parks Associates: At CES 2024, Parks Associates Announces New Research Showing Average Number of Connected Devices per US Internet Household Reached 17 in 2023. `https://www.parksassociates.com/blogs/press-releases/at-ces-2024-parks-associates-announces-new-research-showing-average-number-of-connected-devices-per-us-internet-household-reached-17-in-2023` (2024), accessed: 2024-05-15
42. PyTorch: Retrieved from pytorch model zoo (2017), `https://pytorch.org/vision/stable/models.html`
43. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021)
44. Roy, A.G., Siddiqui, S., Pölsterl, S., Navab, N., Wachinger, C.: Braintorrent: A peer-to-peer environment for decentralized federated learning. arXiv preprint arXiv:1905.06731 (2019)
45. Roy, P., Ghosh, S., Bhattacharya, S., Pal, U.: Effects of degradations on deep neural network architectures. arXiv preprint arXiv:1807.10108 (2018)
46. Rusu, A.A., Rabinowitz, N.C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., Hadsell, R.: Progressive neural networks. arXiv preprint arXiv:1606.04671 (2016)
47. Rusu, A.A., Rabinowitz, N.C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., Hadsell, R.: Progressive neural networks. arXiv preprint arXiv:1606.04671 (2016)
48. Saleh, B., Elgammal, A.: Large-scale classification of fine-art paintings: Learning the right metric on the right feature. arXiv preprint arXiv:1505.00855 (2015)

49. Shayan, M., Fung, C., Yoon, C.J., Beschastnikh, I.: Biscotti: A blockchain system for private and secure federated learning. IEEE Transactions on Parallel and Distributed Systems **32**(7), 1513–1525 (2020)
50. Shenaj, D., Toldo, M., Rigon, A., Zanuttigh, P.: Asynchronous federated continual learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5054–5062 (2023)
51. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
52. Stoychev, S., Churamani, N., Gunes, H.: Latent generative replay for resource-efficient continual learning of facial expressions. In: 2023 IEEE 17th International Conference on Automatic Face and Gesture Recognition (FG). pp. 1–8. IEEE (2023)
53. Sun, W., Yang, Z., Zhang, X., Liu, Y.: Energy-efficient neighbor discovery in mobile ad hoc and wireless sensor networks: A survey. IEEE Communications Surveys & Tutorials **16**(3), 1448–1459 (2014)
54. Tan, M., Le, Q.: Efficientnet: Rethinking model scaling for convolutional neural networks. In: International conference on machine learning. pp. 6105–6114. PMLR (2019)
55. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The caltech-ucsd birds-200-2011 dataset (2011)
56. Wei, K., Li, J., Ding, M., Ma, C., Yang, H.H., Farokhi, F., Jin, S., Quek, T.Q., Poor, H.V.: Federated learning with differential privacy: Algorithms and performance analysis. IEEE transactions on information forensics and security **15**, 3454–3469 (2020)
57. Yoon, J., Jeong, W., Lee, G., Yang, E., Hwang, S.J.: Federated continual learning with weighted inter-client transfer. In: International Conference on Machine Learning. pp. 12073–12086. PMLR (2021)
58. Yoon, J., Yang, E., Lee, J., Hwang, S.J.: Lifelong learning with dynamically expandable networks. arXiv preprint arXiv:1708.01547 (2017)
59. Yu, Q., Liu, Y., Wang, Y., Xu, K., Liu, J.: Multimodal federated learning via contrastive representation ensemble. arXiv preprint arXiv:2302.08888 (2023)
60. Yurochkin, M., Agarwal, M., Ghosh, S., Greenewald, K., Hoang, N., Khazaeni, Y.: Bayesian nonparametric federated learning of neural networks. In: International conference on machine learning. pp. 7252–7261. PMLR (2019)
61. Zhang, J., Chen, C., Zhuang, W., Lyu, L.: Target: Federated class-continual learning via exemplar-free distillation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4782–4793 (2023)
62. Zhang, T., Mai, C., Chang, Y., Chen, C., Shu, L., Zheng, Z.: Fedego: privacy-preserving personalized federated graph learning with ego-graphs. ACM Transactions on Knowledge Discovery from Data **18**(2), 1–27 (2023)
63. Zhu, L., Liu, Z., Han, S.: Deep leakage from gradients. Advances in neural information processing systems **32** (2019)

Appendix

We offer detailed data descriptions in Sec. A1 and configuration details in Sec. A2. Additionally, we provide some detailed results in Sec. A3 and additional evaluations to assess the robustness of *Pick-a-back* in Sec. A4.

A1 Dataset Specification

Datasets. We used four groups of dataset: 1) CIFAR-100: using the original *CIFAR-100* [25] consisting of 20 superclasses with 5 classes each; 2) Classified-ImageNet: extracting 310 classes from *ImageNet* [8] and grouping into 31 superclasses based on WordNet [34]; 3) Fine-Grained: including *CUB* [55], *Stanford Cars* [24], *Flowers* [37], *WikiArt* [48], and *Sketches* [10]; and 4) Digits: using six distinct datasets of Arabia (MNIST) from *MNIST as .jpg* [7], Arabia (Synthetic) from *Synthetic Digits* [45], Arabia (SVHN) from *Street View House Numbers* [36], Arabic from *Arabic Handwritten Characters Dataset* [11], Chinese from *Handwritten Chinese Numbers* [35], and Roman from *Hand Written Roman Numerals Dataset* [39]. The details of the dataset are in Table A1.

Dataset hierarchy. We categorized the classes into superclasses using publicly available sources. In case of *CIFAR-100* [25], the dataset originally consists of 100 classes that are further classified into 20 superclasses. For *Classified-ImageNet*, we manually grouped 310 classes from the original 1,000 classes in the *ImageNet* dataset [8], based on the WordNet [34] hierarchical structure. WordNet is a lexical database that establishes semantic relationships between words, primarily used for text analysis and artificial intelligence applications. The detailed hierarchy for *CIFAR-100* is provided in Table A2, while the hierarchy for *Classified-ImageNet* is presented in Table A3.

Image preprocessing. In order to train the model, we applied image resizing to ensure consistent input sizes across tasks. The original image sizes and the corresponding resized image sizes are presented in Table A4.

How to partition into training and test dataset. For the *CIFAR-100* dataset [25], we utilized the officially provided training and test sets. In the case of the *Classified-ImageNet* dataset [8], as the labels were not available for the test set, we employed the training data for training purposes and the validation data for testing. For the *Fine-Grained* dataset, we utilized the respective datasets provided by *PiggyBack* [29]. Regarding the *Digits* dataset, we split 80% of the training set for training and the remaining 20% for testing for *Arabia* (*MNIST*). *Arabic* (*SVHN*) utilized the training set for training and the validation set for testing. *Arabia* (*SVHN*) and *Arabic* datasets used their respective training and testing sets. Regarding *Chinese* and *Roman* datasets, we divided the data into a training set and a test set, with a ratio of 80% for training and 20% for testing. Furthermore, to increase the task diversity, we extracted digits ranging from one to nine for *Arabic*, *Chinese*, and *Roman* datasets.

Table A1: Summary of datasets

Group	**Dataset**	**# of Clients (= # of Tasks)**	**# of Classes per Task**	**Image Size**	**# of Training Data**	**# of Test Data**
CIFAR-100 [25]		20	5	32×32	50,000	10,000
Classified-ImageNet [8]		31	10	224×224	422,849	16,500
Fine-Grained	**CUB** [55]	5	200	224 × 224	5,994	5,794
	Stanford Cars [24]		196		8,144	8,041
	Flower [37]		102		2,040	6,149
	WikiArt [48]		195		42,129	10,628
	Sketch [10]		250		16,000	4,000
Digits	**Arabia (MNIST)** [7]	6	10 (0∼9)	32 × 32	33,601	8,399
	Arabia (Synthetic) [45]				10,000	2,000
	Arabia (SVHN) [36]				73,257	26,032
	Arabic [11]		9 (1∼9)		54,000	9,000
	Chinese [35]				7,200	1,800
	Roman [39]				1,888	472

Table A2: Dataset hierarchy of *CIFAR-100*

Task ID	**Superclass**	**Subclass**
1	Aquatic mammals	Beacer, Dolphin, Otter, Seal, Whale
2	Fish	Aquarium fish, Flatfish, Ray, Shark, Trout
3	Flowers	Orchid, Poppy, Rose, Sunflower, Tulip
4	Food containers	Bottle, Bowl, Can, Cup, Plate
5	Fruit and vegetables	Apple, Mushroom, Orange, Pear, Sweet pepper
6	Household electrical devices	Clock, Keyboard, Lamp, Telephone, Television
7	Household furniture	Bed, Chair, Couch, Table, Wardrobe
8	Insects	Bee, Beetle, Butterfly, Caterpillar, Cockroach
9	Large carnivores	Bear, Leopard, Lion, Tiger, Wolf
10	Large man-made outdoor things	bridge, Castle, House, Road, Skyscraper
11	Large natural outdoor scenes	Cloud, Forest, Mountain, Plain, Sea
12	Large omnivores and herbivores	Camel, Cattle, Chimpanzee, Elephant, Kangaroo
13	Medium mammals	Fox, Porcupine, Possum, Raccoon, Skunk
14	Non-insect invertebrates	Crab, Lobster, Snail, Spider, worm
15	People	Baby, Boy, Girl, Man, Woman
16	Reptiles	Crocodile, Dinosaur, Lizard, Snake, Turtle
17	Small mammals	Hamster, Mouse, Rabbit, Shrew, Squirrel
18	Trees	Maple tree, Oak tree, Palm tree, Pine tree, Willow tree
19	Vehicles 1	Bicycle, Bus, Motorcycle, Pickup truck, Train
20	Vehicles 2	Lawn mower, Rocket, Streetcar, Tank, Tractor

A2 Model Configurations and Learning Parameters

Model configurations. We implemented benchmark architectures including LeNet-5 [26], MobileNet-V1 [17], VGG16-BN [51], and EfficientNet-B0 [54]. Pretrained models were utilized by loading the publicly available models from the *PyTorch* [42].

Table A3: Dataset hierarchy of *Classified-ImageNet*

Task ID	**Superclass**	**Subclass**
1	Bird	Bald eagle, Black grouse, Black stork, Crane, Junco, Lorikeet, Red-breasted merganser, Ruddy turnstone, Sulphur-crested cockatoo, Vulture
2	Clothing	Bearskin, Bonnet, Bow tie, Cowboy hat, Feather boa, Knee pad, Mask, Miniskirt, Mitten, Sarong
3	Container/Others	Beer glass, Carton, Cassette, Envelope, Milk can, Packet, Plastic bag, Purse, Sleeping bag, Soap dispenser
4	Container/Vessel	Beer bottle, Bucket, Caldron, Coffee mug, Ladle, Mortar, Pitcher, Tub, Vase, Water jug
5	Covering	Bottlecap, Cloak, Clog, Cowboy boot, Doormat, Loafer, Manhole cover, Nipple, Prayer rug, Running shoe
6	Craft	Airliner, Airship, Balloon, Canoe, Container ship, Fireboat, Gondola, Pirate, Schooner, Submarine
7	Device/Machine	Abacus, Cash machine, Desktop computer, Hand-held computer, Harvester, Laptop, Notebook, Slot, Thresher, Vending machine
8	Device/Musical instrument	Banjo, Bassoon, Cello, Cornet, Electric guitar, Grand piano, Harp, Maraca, Ocarina, Panpipe
9	Device/Others	Combination lock, Electric fan, Gas pump, Loudspeaker, Padlock, Pier, Pinwheel, Safety pin, Switch, Typewrite keyboard
10	Dog	Beagle, Border collie, Borzoi, Collie, Irish wolfhound, Komondor, Mexican hairless, Shetland sheepdog, Staffordshire bullterrier, Toy poodle
11	Equipment	Carpenter's kit, Cellular telephone, Computer keyboard, Drilling platform, Golf ball, Horizontal bar, iPod, Oscilloscope, Photocopier, Rugby ball
12	Fish	Barracouta, Eel, Electric ray, Great white shark, Hammerhead, Lionfish, Rock beauty, Stingray, Sturgeon, Tiger shark
13	Fruit	Banana, Corn, Custard, Apple, Granny smith, Hip, Jackfruit, Orange, Pineapple, Pomegranate, Strawberry
14	Furnishing	Bookcase, Chiffonier, China cabinet, Crib, Desk, Dining table, File, Park bench, Pool table, Table lamp
15	Geological Formation	Alp, Cliff, Coral reef, Geyser, Lakeside, Promontory, Sandbar, Seashore, Valley, Volcano
16	Implement/Others	Crockpot, Crutch, Drumstick, Fountain pen, Frying pan, Paintbrush, Pole, Quill, Rubber erase, Spindle
17	Implement/Tool	Chain saw, Cleaver, Corkscrew, Hammer, Hatchet, Lawn mower, Letter opener, Plane, Plow, Power drill
18	Instrument	Binoculars, Digital clock, Digital watch, Hourglass, Loupe, Magnetic compass, Parking meter, Scale, Stethoscope, Sunglass
19	Invertebrate/Insects	Bee, Cabbage butterfly, Cicada, Dragonfly, Ground beetle, Ladybug, Leafhopper, Lycaenid, Monarch, Tiger beetle
20	Invertebrate/Others	Black widow, Brain coral, Conch, Flatworm, Harvestman, Hermit crab, Isopod, Rock crab, Sea slug, Starfish
21	Placental/Carnivore/Canine	Arctic fox, Coyote, Dhole, Dingo, Grey fox, Hyena, Red fox, Red wolf, Timber wolf, White wolf
22	Placental/Carnivore/Others	Black-footed ferret, Brown bear, Cheetah, Ice bear, Meerkat, Mink, Persian cat, Siamese cat, Tabby, Tiger
23	Placental/Others	Arabian camel, Armadillo, Beaver, Dugong, Fox squirrel, Guinea pig, Hare, Indian elephant, Killer whale, Three-toed sloth
24	Placental/Primate	Chimpanzee, Colobus, Gibbon, Howler monkey, Indri, Macaque, Marmoset, Orangutan, Patas, Squirrel monkey
25	Placental/Ungulate	Hartebeest, Hippopotamus, Hog, Ibex, Impala, Llama, Ox, Ram, Sorrel, Water buffalo
26	Protective Covering	Birdhouse, Breastplate, Chain mail, Dome, Gasmask, Mosquito net, Scabbard, Shield, Thatch, Thimble
27	Reptile	American alligator, Boa constrictor, Box turtle, Garter snake, Green lizard, Green mamba, Green snake, King snake, Mud turtle, Thunder snake
28	Structure	Barbershop, Boathouse, Bookshop, Cliff dwelling, Megalith, Monastery, Mosque, Pedestal, Tobacco shop, Triumphal arch
29	Vegetable	Acorn squash, Artichoke, Bell pepper, Broccoli, Butternut squash, Cardoon, Cauliflower, Head cabbage, Mushroom, Spaghetti squash
30	Wheeled vehicle/Motor	Beach wagon, Cab, Convertible, Garbage truck, Go-kart, Golfcart, Limousine, Moving van, Pickup, Snowplow
31	Wheeled vehicle/Others	Electric locomotive, Forklift, Freight car, Jinrikisha, Motor scooter, Oxcart, Recreational vehicle, Shopping cart, Snowmobile, Tricycle

Table A4: Image preprocessing for resizing ('-' indicates that the sizes of images are not identical)

Dataset	**Original Image**	**Resized Image**
CIFAR-100	32×32	
Classified-ImageNet	-	224×224
CUB	-	
Stanford Cars	-	
Flower	-	224×224
WikiArt	-	
Sketch	-	
Arabia (MNIST)	28×28	
Arabia (Synthetic)	-	
Arabia (SVHN)	32×32	32×32
Arabic	28×28	
Chinese	64×64	
Roman	-	

Table A5: Performance on CIFAR-100 using LeNet-5 using our backbone selection in other (federated) continual approaches. *w/o Backbone* results are different across methods

Client ID	**15**	**1**	**17**	**16**	**18**	**12**	**3**	**14**	**7**	**6**	**8**	**2**	**4**	**13**	**19**	**5**	**10**	**9**	**20**	**11**	**Avg**
w/o Backbone	26.2	46.0	48.6	53.8	55.2	54.2	63.8	57.0	62.2	64.6	64.0	67.8	68.6	64.0	63.4	64.4	61.8	51.8	72.8	76.0	**59.3**
WSN	34.2	53.8	49.4	53.4	59.8	60.2	63.6	63.0	66.8	65.0	64.6	65.4	66.6	62.6	61.2	67.6	66.6	58.6	74.0	79.8	**61.8**
w/o Backbone	42.8	50.5	51.7	56.0	58.3	58.8	59.3	62.5	64.7	64.2	63.3	65.7	67.8	64.3	59.2	69.8	63.0	56.2	57.7	76.3	**60.6**
FedWeIT	40.0	52.5	55.0	56.0	56.2	60.5	56.7	60.5	67.0	65.2	60.2	65.0	66.0	64.2	65.0	65.7	66.7	51.0	68.7	77.8	**61.0**
w/o Backbone	37.4	51.2	49.2	58.6	56.6	62.0	64.0	61.4	64.2	63.0	64.4	66.4	71.8	65.2	66.2	70.6	73.0	71.2	72.2	73.0	**63.1**
FCCL	37.0	54.4	52.4	60.6	62.2	68.2	67.2	64.4	68.2	70.6	72.8	72.4	68.4	**77.0**	73.8	75.0	**78.6**	68.8	79.2	78.2	**67.5**
w/o Backbone	40.0	55.0	57.2	60.4	51.8	66.0	67.4	67.6	69.2	70.8	72.6	**73.0**	**74.2**	75.2	75.8	76.4	77.0	**77.4**	81.2	84.0	**69.1**
Pick-a-back	**46.4**	**59.0**	**60.0**	**63.2**	**64.4**	**72.6**	**71.8**	**75.6**	**72.4**	**72.2**	**77.0**	**73.0**	72.8	76.2	**76.0**	**80.2**	78.2	75.8	**84.2**	**85.4**	71.8

Learning parameters. In our experiments on all datasets, we employed the Stochastic Gradient Descent (SGD) optimizer with a momentum of 0.9 to train a model. The training process consisted of 100 epochs. We initialized the learning rate to 0.01 and decreased it by a factor of 0.1 at epochs 50 and 80 to facilitate convergence. Additionally, as we trained the masks on top of the backbone model, we utilized the Adam optimizer for 20 epochs with a learning rate of 0.001. To ensure model compactness during continual learning, we imposed a constraint on the expansion size, limiting it to the number of tasks. For example, when training two sequential tasks, the model's expansion is restricted to 2 times to the size of the scratch model.

A3 Detailed Results

Ablation study on selective knowledge federation (detailed results of Table 1. Our approach is orthogonal to the choice of federated/continual learning algorithms. To verify the effectiveness of our knowledge selection approach, we demonstrated accuracy improvement by integrating our backbone knowledge

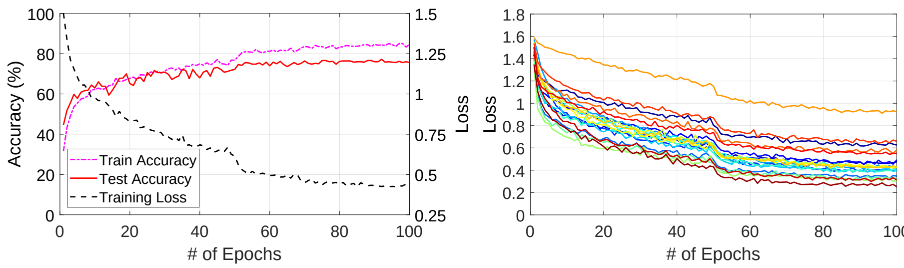
Table A6: Performance in ImageNet-pretrained models using CIFAR-100

		w/o Backbone → Pick-a-back		**Training Data**
		Avg Acc.	**Top Acc.**	
VGG16-BN	**w/ Pretrained**	84.9 → 83.8	82.8 → 84.4	92.5 + 0.3 GB
	w/o Pretrained	78.9 → 81.0	68.2 → 76.6	0.3 GB
EfficientNet-B0	**w/ Pretrained**	89.8 → 87.1	75.8 → 76.6	92.5 + 0.3 GB
	w/o Pretrained	77.8 → 81.8	44.6 → 54.4	0.3 GB

selection into existing (federated) continual learning approaches, as shown in Table A5. Different from *ProgressiveNet* [47], *PackNet* [30], and *CPG* [19], which were implemented on our code-base with the same continual learning architecture, we have implemented *WSN*[A1], *FedWeIT*[A2], and *FCCL*[A3] on their respective code-bases. Although all approaches use the same LeNet-5 architecture, their detailed learning configurations are different, resulting in varied *w/o Backbone* accuracy. Our key metric of interest is in the efficacy of *Pick-a-back* in performance improvement by selecting beneficial backbone knowledge from others. With our backbone selection approach, *WSN*, *FedWeIT*, and *FCCL* show accuracy improvements of 2.5%, 0.4%, and 4.4%, respectively on average, and 6.0%, 11.0%, and 11.8%, respectively at maximum, compared to their *w/o Backbone*, implying the versatility of *Pick-a-back*.

ImageNet-pretrained models (detailed results of Table 6. We validated our approach using EfficientNet-B0 in Table A6. Due to the small size of images in CIFAR-100 with 32×32 pixels, it is not feasible to train the EfficientNet-B0 model directly on original images. Therefore, we enlarged the images to 224×224 pixels and evaluated the performance accordingly. It should be noted that the results for VGG16-BN are using the original size of the images. As expected, we observed an increase in performance from 77.8% to 81.8% on average and from 44.6% to 54.4% in the most improved task. Similar to VGG16-BN, initializing EfficientNet-B0 with the ImageNet-pretrained model did not yield significant benefits in terms of average performance, but we still observed improvements in the effective cases. Furthermore, when we initialize some models with powerful datasets, the amount of data required is excessively expensive, approximately 300 times in our case. Even though some pretrained models with ImageNet are publicly available, they mostly consist of large-scale models that are not suitable for edge scenarios. Moreover, ImageNet-pretrained models may adapt well to general image tasks, but the specific tasks require different and task-specific knowledge. It highlights the importance of *selective* knowledge, as demonstrated by the effectiveness of our *Pick-a-back*.

[A1] https://github.com/ihaeyong/WSN
[A2] https://github.com/wyjeong/FedWeIT
[A3] https://github.com/WenkeHuang/FCCL



(a) Task 14 **(b)** All tasks

Fig. A1: Training dynamics with respect to training epochs

Table A7: Grouping 2 superclasses into 1 task with 10 classes

Task ID	Grouped Superclasses
1	Aquatic mammals, Fish
2	Flowers, Trees
3	Food containers, Fruit and vegetables
4	Household electrical devices, Household furniture
5	Insects, Non-insect invertebrates
6	Large carnivores, Large omnivores and herbivores
7	Large man-made outdoor things, Large natural outdoor scenes
8	Medium mammals, People
9	Reptiles, Small mammals
10	Vehicles 1, Vehicles 2

Table A8: Grouping 4 superclasses into one task consisting of 20 classes

Task ID	Grouped Superclasses
1	Aquatic mammals, Fish, Large carnivores, Large omnivores and herbivores
2	Flowers, Trees, Large man-made outdoor things, Large natural outdoor scenes
3	Food containers, Fruit and vegetables, Insects, Non-insect invertebrates
4	Household electrical devices, Household furniture, Vehicles 1, Vehicles 2
5	Medium mammals, People, Reptiles, Small mammals

A4 Additional Results

We conducted additional experiments to show the convergence of our framework and validate the stability of our *Pick-a-back* approach by varying dataset hierarchies and the learning seeds. Furthermore, we included another privacy leakage problem from model sharing.

Convergence analysis. *Pick-a-back* provides beneficial knowledge selection for the subsequent task, and thus, *Pick-a-back* is orthogonal to learning approaches. Please refer to the convergence proof in each continual learning [1]. To show the convergence of *Pick-a-back* with *CPG* [19], we empirically showed the training loss for the most effective case, task 14 in Fig. A1a and across all tasks in Fig. A1b.

Scalability of task-class combinations. We revisited the CIFAR-100 dataset by grouping 20 clients (i.e., superclasses) into 10 and 5 groups (the detailed class

Table A9: Performance with different number of clients by grouping superclass of CIFAR-100 using LeNet5

	5 Clients		**10 Clients**		**20 Clients**	
	Avg	**Top**	**Avg**	**Top**	**Avg**	**Top**
Upper	64.7	55.1	65.7	68.1	73.7	75.6
w/o Backbone	54.6	49.1	61.6	62.4	69.1	67.6
Pick-a-back	58.1 (+3.5)	53.5 (+4.4)	63.6 (+2.0)	67.4 (+5.0)	71.8 (+2.7)	75.6 (**+8.0**)

Table A10: Performance of various architectures and datasets using different seeds

Dataset	**Architecture**		**Accuracy**	
			Avg	**Top**
CIFAR-100	**VGG16-BN**	**w/o Backbone**	79.1	61.6
		Pick-a-back	80.6 (+1.5)	67.8 (+6.2)
	LeNet-5	**w/o Backbone**	68.5	70.4
		Pick-a-back	71.0 (+2.5)	78.2 (+7.8)
		w/o Backbone	69.2	58.5
		Pick-a-back	71.5 (+2.3)	65.4 (+7.4)
		w/o Backbone	68.9	68.0
		Pick-a-back	71.5 (+2.6)	74.8 (+6.8)
Digits	**LeNet-5**	**w/o Backbone**	93.4	79.7
		Pick-a-back	95.0 (+1.6)	86.4 (+6.7)
Fine-Grained	**LeNet-5**	**w/o Backbone**	24.8	20.5
		Pick-a-back	30.8 (+6.0)	42.3 (+21.8)

combination is Table A7 and Table A8, respectively). As depicted in Table A9, it is evident that as the number of classes per task increases, the overall accuracy naturally decreases. However, with the utilization of *Pick-a-back*, we observed an average task accuracy improvement of 2.7%, 2.0%, and 3.5% across 20 tasks. In the most effective cases, we achieved inference accuracy improvement of 8.0%, 5.0%, and 4.4%, which are quite reasonable compared to the *Upper* cases.

Robustness against learning seeds. In Table A10, we present the evaluation results of different datasets and architectures using various learning seeds, as shown in Tables 2 and 3. As demonstrated in Sec. 4, the selective knowledge reuse facilitated by *Pick-a-back* leads to improved inference accuracy. Notably, for the Fine-Grained dataset, the most effective case achieved an accuracy of 27.5% with a notable improvement of 6.5%, as demonstrated in Table 3), and there is potential for further improvement up to 48.5% in the *Upper*. When applying different learning seeds in Table A10, we observed accuracy improvements of up to 21.8%, highlighting the significant benefits of leveraging effective knowledge. Although there are variations regarding learning seeds, *Pick-a-back* consistently delivers accuracy improvements both on average and in maximum performance.

Synergy between continual learning and selective device-to-device learning. As in Table A11, we evaluated the performance to verify the compatibility

Table A11: Orthogonality of continual learner on CIFAR-100 using LeNet-5 when testing after the 2^{nd} task. Higher sparsity represents more neurons to be released

		1st Task→		2nd Task		Expansion (Sparsity)
		Avg	**Top**	**Avg**	**Top**	
CPG	**w/o Backbone**	-		69.1	67.6	1.0 (0.73)
	Pick-a-back	67.9	73.2	**71.8**	75.6	2.0 (0.86)
	Upper		73.2	73.7	75.6	
Progressive Networks	**w/o Backbone**	-		69.7	38.4	1.0 (0.00)
	Pick-a-back	69.7	70.0	**69.9**	44.6	2.0 (0.00)
	Upper		70.4	72.4	45.6	
PackNet	**w/o Backbone**	-		69.7	72.2	1.0 (0.75)
	Pick-a-back	68.6	69.0	**69.5**	78.0	1.0 (0.71)
	Upper		69.0	72.7	78.0	

with different continual learning approaches. The results indicate that the accuracy of the backbone task remains stable across all of the approaches. Noticeably, *Pick-a-back* further improves the performance of both *CPG* and *Progressive Networks*, while showing a slight degradation in *PackNet*. This can be attributed to the specific strategy employed by *PackNet*, which does not expand the model and thus limits the capacity for accumulating new task knowledge in its backbone network. However, we observed substantial enhancements in specific tasks, with accuracy improvements of up to 8.0%, 6.2%, and 5.8%, respectively across all of the approaches. These findings highlight the versatility of *Pick-a-back*, as it can be effectively applied in conjunction with any continuous learning approach that leverages the reuse of backbone knowledge.

Edge-effective knowledge selection. The *Adversarial Set with Genetic Algorithm (GA)*, introduced in *ModelDiff* [27], selects effective pairwise inputs by mutating input data, primarily employed in *Pick-a-back*. As it needs image sharing among all of the clients, which may incur transmission overhead and privacy issues, we further assess *Pick-a-back* in various ways, using different input sets: 1) *Noise*: by comprising totally random noise data, unrelated to target tasks, without requiring any raw data from clients; 2) *Local*: by sampling the validation data from the requester; and 3) *Mutated Local*: by adding noise to *Local* to address privacy concerns arising from direct raw data delivery. We also compared the results without the genetic algorithm (GA) to reduce the P2P cost. As shown in Table A12a, while *Adv. w/ GA* yields the most compelling outcomes, *Pick-a-back* with *Local w/ GA* also demonstrates comparable results. For communication efficiency and privacy problems, the option of revealing solely the requester's data, adding some variations on top of raw data, or using even random data emerges as viable strategies. Encouragingly, all approaches achieve performance enhancements, compared to 69.1% of results without backbone.

Table A12: Performance comparison on CIFAR-100 using LeNet-5 with respect to diverse input data to measure the model similarity

(a) Averaged accuracy and accuracy improvement

	Avg	**Top**
w/o Backbone	69.1	51.8
Noise	70.2 (+1.1)	65.2 (+13.4)
Local	70.9 (+1.8)	63.6 (+11.8)
Mutated Local	71.2 (+2.1)	65.0 (+13.2)
Local w/ GA	71.1 (+2.0)	63.2 (+11.4)
Mutated Local w/ GA	71.2 (+2.1)	65.0 (+13.2)
Mutated Adv. w/ GA	70.8 (+1.7)	64.0 (+12.2)
Adv. w/ GA	<u>**71.8 (+2.7)**</u>	64.4 (+12.6)

(b) Accuracy per each task

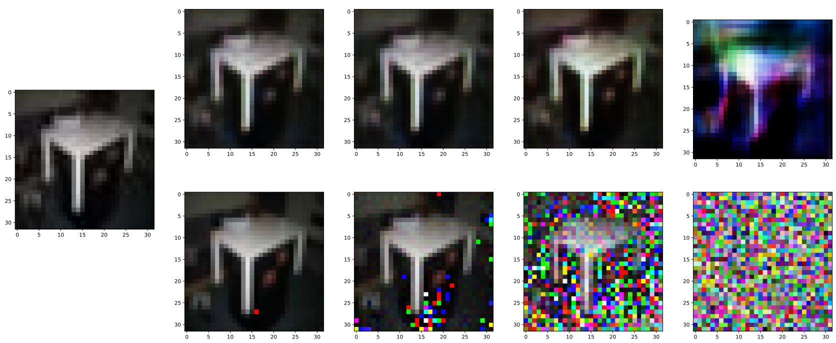
Client (Task) ID	**15**	**1**	**17**	**16**	**18**	**12**	**3**	**14**	**7**	**6**	**8**	**2**	**4**	**13**	**19**	**5**	**10**	**9**	**20**	**11**	**Avg**
w/o Backbone	40.0	55.0	57.2	60.4	51.8	66.0	67.4	67.6	69.2	70.8	72.6	73.0	74.2	75.2	75.8	76.4	77.0	77.4	81.2	84.0	**69.1**
Noise	**46.4**	55.6	59.2	**63.6**	65.2	69.8	71.0	69.8	71.0	**74.0**	71.6	72.6	71.6	77.2	74.4	76.4	76.4	71.4	82.2	**85.4**	**70.2**
Local	42.4	58.8	58.8	62.4	63.6	70.2	**72.0**	73.4	71.0	67.0	71.6	73.6	74.4	77.2	76.0	79.0	**80.8**	**78.8**	**84.2**	82.0	**70.9**
Mutated Local	39.8	58.4	59.2	61.4	65.0	**72.6**	71.2	69.8	71.4	73.0	**77.0**	74.6	**76.0**	**78.2**	**77.4**	**80.2**	78.2	75.8	83.6	82.0	**71.2**
Noise w/ GA	41.8	58.2	56.2	**63.6**	**66.8**	70.2	70.6	73.4	71.4	72.6	74.2	71.6	74.6	76.2	76.8	76.4	81.2	74.2	82.8	82.4	**70.8**
Local w/ GA	39.2	**59.0**	54.8	62.4	63.2	**72.6**	71.8	75.2	71.4	72.2	75.4	**75.0**	74.2	76.2	76.0	77.4	**80.8**	**78.8**	**84.2**	83.0	**71.1**
Mutated Local w/ GA	39.6	58.2	59.2	62.4	65.0	71.6	71.6	69.8	69.4	73.4	77.0	74.6	76.0	76.0	77.4	78.2	80.8	74.6	86.2	82.0	**71.2**
Mutated Adv. w/ GA	43.0	58.2	58.8	63.2	64.0	69.4	71.0	65.6	71.6	73.0	74.2	74.6	74.6	76.6	76.8	79.2	78.8	75.8	82.2	84.6	**70.8**
Adv. w/ GA	**46.4**	**59.0**	**60.0**	63.2	64.4	**72.6**	**71.8**	**75.6**	**72.4**	72.2	**77.0**	73.0	72.8	76.2	76.0	**80.2**	78.2	75.8	**84.2**	**85.4**	<u>71.8</u>

Furthermore, there can also be privacy leakage through the sharing of model parameters.

More closely looking into the detailed accuracy of each task, as presented in Table A12b, and examining the number of negative cases in comparison to *w/o Backbone*, we observe 6, 4, 3, 3, 3, 3, 2, and 2 negative cases for *Noise*, *Local*, *Mutated Local*, *Noise w/ GA*, *Local w/ GA*, *Mutated Local w/ GA*, *Mutated Adv. w/ GA*, and *Adv. w/ GA*, respectively. This indicates that *Noise* poses a higher risk of adverse effects when using backbone models. Although *Adv. w/ GA* achieves the best accuracy with the fewest negative cases, *Mutated Local*, which transmits mutated local data without periodic exchange, also demonstrates comparable results with only one additional negative case. It suggests that employing cost-effective and privacy-preserving similarity computation could be a promising solution for edge-side learning.

Defense against privacy leakage problem from model transfer. As a federated continual learning paradigm, each device retrieves the model from the selected backbone model, and thereby, there can be another privacy leakage problem by sharing the model parameters [56, 63]. In Fig. A2, we explore how a common yet effective defense technique to apply noise into the model parameters, also introduced in *DLG* [63], can survive against data reconstruction attacks

	Original	w/o Noise	$\sigma = 10^{-3}$	$\sigma = 10^{-2}$	$\sigma = 10^{-1}$



Acc.	w/o Noise	$\sigma = 10^{-3}$	$\sigma = 10^{-2}$	$\sigma = 10^{-1}$
Avg:	71.8 (+2.7)	70.4 (+1.3)	69.9 (+0.8)	69.2 (+0.1)
Top (Task 14):	75.6 (+8.0)	73.2 (+5.6)	72.0 (+4.4)	70.2 (+2.6)

Fig. A2: Recovered images (using *DLG* [63] and *Autoencoder* [56]) and performance on a noise-additive model with Gaussian noise of $\mu = 0$ and variance σ

from revealed model parameters. We showed the accuracy with varying levels of noise and their corresponding image recovery capability using two baselines: 1) *DLG* [63]: to recover the image from gradients generated by noise-added models[A4]; and 2) *Autoencoder* [12]: to extract (unachievable) decoders from trained autoencoder[A5] and recover the image from outputs generated by noise-added encoders. As expected, increasing the noise level makes image recovery more difficult, similar to random noise initialization, resulting in a corresponding reduction in performance gain. However, even with large noise, *Pick-a-back* still shows the improvement. It implies that the noise-based defense mechanism significantly enhances privacy, while showing the effectiveness of our knowledge transfer. However, it should be noted that our approach only shares model parameters, not exposing any input data-oriented information such as gradients (in *DLG*) or output features (in *Autoencoder*). This means that even if attackers know the model parameters, it is hard to reconstruct the original input data. While recent techniques attempt to recover training data only from model parameters [15], the recovered inputs are very noisy, and they cannot specify the image to recover. Also, it may show more noisy data if we further apply the noise-additive models. This highlights that our model parameter-only sharing approach ensures a high privacy level, as verified by the difficulty in reconstructing meaningful input data from the shared parameters alone.

Scalability of neighboring devices and the complexity for real-world settings. We primarily investigate the communication-/computation-wise com-

[A4] https://github.com/mit-han-lab/dlg

[A5] Intro to Autoencoders, https://www.tensorflow.org/tutorials/generative/autoencoder, TensorFlow plexity to find the neighboring device with the most effective backbone. Here, there is a trade-off in similarity computation, where it is more likely to find better backbone knowledge with more neighbors, while incurring higher transmission and computation costs at the same time. This cost is especially critical in device-to-device (D2D) settings, as peer-to-peer (P2P) [44, 49] is with logical connection (usually in wired networks), whereas D2D implies physical connection (typically wireless or geographically proximate). Thereby, we measured time complexity on the device only with CPU using Wi-Fi 6, and it consumes only 8.7025 sec for similarity computation and 0.0020 sec for transmission per client for input data sharing on an average of 10 trials. It means that, in Table 4, we can gain improvement of up to 51.9% within less than a minute. In practice, considering that the average number of connected devices per US Internet household is 17 [41], it took $(8.7025\ sec + 0.0020\ sec) \times 17 \approx 2.5\ min$, emphasizing the cost-effectiveness of *Pick-a-back* in D2D knowledge transfer.