# PARTIALLY RELAXED MASKS FOR LIGHTWEIGHT KNOWLEDGE TRANSFER WITHOUT FORGETTING IN CONTINUAL LEARNING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

The existing research on continual learning (CL) has focused mainly on preventing catastrophic forgetting. In the task-incremental learning setting of CL, several approaches have achieved excellent results, with almost no forgetting. The goal of this work is to endow such systems with the additional ability to transfer knowledge among tasks when the tasks are similar and have shared knowledge to achieve higher accuracy. Since the existing system HAT is one of most effective task-incremental learning algorithms, this paper extends HAT with the aim of both objectives, i.e., overcoming catastrophic forgetting and transferring knowledge among tasks without introducing additional mechanisms into the architecture of HAT. The current study finds that task similarity, which indicates knowledge sharing and transfer, can be computed via the clustering of task embeddings optimized by HAT. Thus, we propose a new approach, named "partially relaxed masks" (PRM), to exploit HAT's masks to not only keep some parameters from being modified in learning subsequent tasks as much as possible to prevent forgetting but also enable remaining parameters to be updated to facilitate knowledge transfer. Extensive experiments demonstrate that PRM performs competitively compared with the latest baselines while also requiring much less computation time.

## 1 INTRODUCTION

Continual learning has recently received substantial attention with the increasing popularity of AI-embedded systems, but these systems still struggle to maintain performance without retraining the model from scratch, which generally consumes a large amount of time. The main issue in continual learning is *catastrophic forgetting*, which refers to the phenomenon in which once a model has learned a new task or new data, its performance is likely to decline drastically on the previously learned data (Goodfellow et al., 2014); thus, many studies have proposed approaches that address this issue (Delange et al., 2021; Parisi et al., 2019). In particular, HAT (Serra et al., 2018) proposes a mechanism called hard attention that blocks the gradients of parameters, which are important for previous tasks to overcome forgetting, and it achieves learning with almost no forgetting.

However, for the realistic use of continual learning, consideration of only the forgetting issue is not sufficient. There must sometimes be similar tasks that can be exploited for other tasks and dissimilar tasks that are sensitive to forgetting issues at the same time; however, conventional approaches that have focused mainly on forgetting issues have not fully considered task similarity, which can enhance performance. Therefore, another research theme has become the transfer of knowledge into a newly coming task from previous tasks where, as a matter of course, forgetting should be restrained. These challenges have been represented as *task incremental learning* (TIL), which aims at learning a mixed sequence of similar and dissimilar tasks.

Additionally, looking ahead to the realistic use of continual learning, where AI-embedded edge devices that learn continuously but do not have substantial computational resources are most commonly utilized, several studies have focused mainly on the efficiency of learning (Borsos et al., 2020; Pellegrini et al., 2020). To advance to the next stage, continual learning methods also need to be as efficient as possible. CAT (Ke et al., 2020), for instance, is the first approach for tackling a

mixed sequence of similar and dissimilar tasks. CAT extends HAT by introducing attention mechanisms across similar tasks to enhance knowledge transfer; however, it is very inefficient and takes a much longer time for task similarity detection because it tries every previous task one by one to judge whether each task is worth being transferred to the current learning task. Although it succeeds in task similarity detection and outperforms HAT, it still faces enormous problems in terms of its efficiency and scalability.

To address these issues, the current study extends HAT so that it can enhance knowledge transfer without another mechanism, such as attention, and even with much shorter computation time. Our contributions to this challenge are as follows. First, the current study discovers that task similarity can be computed from task embeddings that are optimized by a HAT-like approach. Second, we propose a brand new approach named "partially relaxed masks" (PRM) that employs the masks that are accumulated only for dissimilar tasks so that it maintains parameters that are important for the dissimilar tasks as much as possible to prevent forgetting, while keeping the remaining parameters, which are useful for the similar tasks, free to be updated to enhance knowledge transfer. Extensive experiments demonstrate that our approach achieves equal or greater performance than state-of-the-art methods and requires much less computation time.

## 2 Related work

Continual learning approaches are categorized into three main types: 1) regularization-based methods (Kirkpatrick et al., 2017; Li & Hoiem, 2016; Zenke et al., 2017), which add another penalty so as not to change the important parameters for previous tasks; 2) replay-based methods (Chaudhry et al., 2021; 2019; Lopez-Paz & Ranzato, 2017; Riemer et al., 2019), which keep the small size of previous tasks' samples and exploit them to alleviate forgetting; and 3) parameter isolation-based (Hu, Wenpeng and Lin, Zhou and Liu, Bing and Tao, Chongyang and Tao, Zhengwei and Ma, Jinwen and Zhao, Dongyan and Yan, Rui, 2018; Rusu et al., 2016; Singh et al., 2020; von Oswald et al., 2020) methods, which create new branches for new tasks, which are defined by new parameters. EWC (Kirkpatrick et al., 2017) is one of the most popular regularization-based methods. It computes the Fisher information matrix that represents the importance of each parameter and adds a regularization term that corresponds to the matrix to prevent forgetting. A-GEM (Chaudhry et al., 2019) is a typical replay-based method, and it uses an efficient approach to select samples of previous tasks that will be learned together in a current task. A major issue with replay-based approaches is that they require an additional memory buffer for saving past samples. To address this issue, instead of real sampling, many approaches exploit a data generator inside the model, and the generated samples are used with current learning; such methods are referred to as pseudo replay-based methods. One of the latest parameter isolation-based methods is CCLL (Singh et al., 2020), which prevents forgetting with few additional parameters by introducing calibration modules that convert activation maps for previous tasks to a current task. Recently, several approaches have been combined with the meta-learning paradigm to select more effective samples or parameters (Chaudhry et al., 2021; Javed & White, 2019).

Although conventional approaches, as typified by the above, have focused mainly on catastrophic forgetting, most do not have any mechanism for transferring knowledge across similar tasks, which has become another important topic in TIL. In particular, HAT (Serra et al., 2018) achieves learning with almost no forgetting by introducing hard attention to block the updating of parameters that are important for previous tasks; however, the mechanism no longer enhances knowledge transfer, which is another issue in HAT in terms of TIL. CAT (Ke et al., 2020) is the first approach to deal with a mixed sequence of dissimilar tasks and similar tasks at the same time by extending HAT. Since HAT does not have any mechanism for knowledge transfer, CAT introduces additional attention operations into classifiers and judges similar tasks using another network separately. Although CAT achieves state-of-the-art performance in this scenario, it requires substantial computational resources for task similarity detection because it tries to build and train the reference and transfer models per the previous task and check whether this transfer actually improves its performance (i.e., whether the transfer model outperforms the reference model). Therefore, the more previous tasks there are, the longer CAT takes to learn a new task.

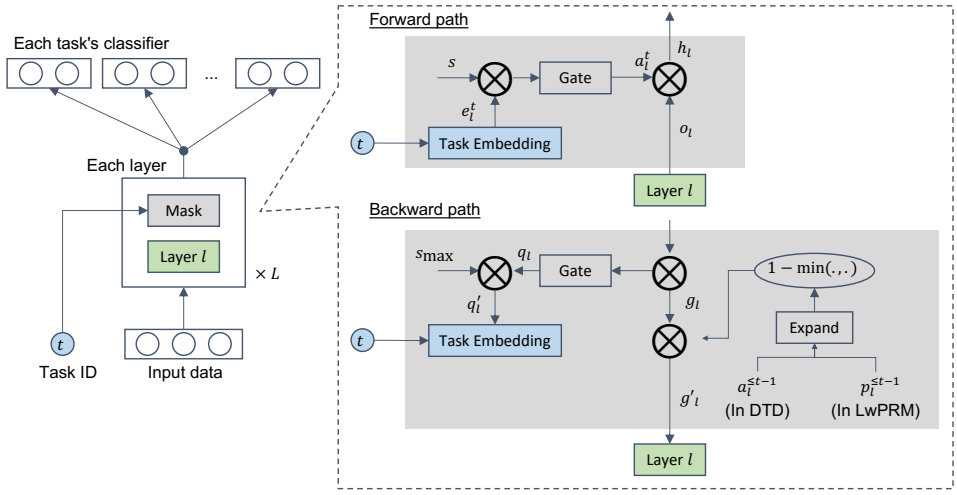


Figure 1: Structure of PRM for learning task $t$. PRM follows almost the same procedure as HAT, except for part of the backward path. In dissimilar task detection (DTD), the model uses $a_l^{\leq t-1}$ to block all the parameters. In learning with partially relaxed masks (LwPRM), it uses $p_l^{\leq t-1}$ instead, which blocks only the parameters that are important for the previous dissimilar tasks.

**Algorithm 1** Learning procedure for PRM

**Input:** $x_{1:T}, y_{1:T}$, Model $M$ with $L$ layers
**for** $t = 1 \cdots T$ **do**
*# Dissimilar task detection (DTD) phase*
state $\leftarrow$ copy($M$)
1st optimization: Freeze feature extractor and train only classifier of $M$ with $x_t, y_t$
2nd optimization: Train whole $M$ with $x_t, y_t$
Task embeddings $\left\{e_l^{1:t}\right\} \leftarrow M$
Set of dissimilar tasks $\mathcal{D}_l^t \leftarrow$ Clustering $\left(\left\{e_l^{1:t}\right\}\right)$
Load back: $M \leftarrow$ state

*# Learning with partially relaxed masks (LwPRM) phase*
Train Model with $x_t, y_t$ and $\mathcal{D}_l^t$

## 3 Proposed PRM

The structure and procedure of our proposed method are presented in Figure 1 and Algorithm 1. The procedure is composed of two phases: 1) dissimilar task detection (DTD) and 2) learning with partially relaxed masks (LwPRM). Although PRM needs twice learning per task, it does not take much computational overhead as shown in the experiments.

First, the DTD phase aims to obtain the task similarity from the task embeddings that are optimized in the same way as with HAT. Since HAT utilizes accumulated masks to block the gradients of the model's parameters, the task embeddings are more flexibly updated than the model's parameters; thus, we expect the task embeddings to provide an informative representation for task similarity. The reason that we focus on task embeddings to measure task similarity is that since they are used as the basis for masking the output of each layer, if two tasks emphasize similar parameters and try to pass them without blocking (masking), their task embeddings should be similar. Once the optimized task embeddings, which are represented by $\{e_l^{1:t}\}$, where $l$ and $t$ denote the indices of the layer and task, respectively, are obtained, the set of dissimilar tasks, $\mathcal{D}_l^t$, can be computed via a clustering on the embeddings. Second, in the LwPRM phase, the model's parameters are optimized again using $\mathcal{D}_l^t$ with the intention of not only blocking some parameters to overcome forgetting, as with HAT, but also making remaining parameters free for updating to facilitate knowledge transfer. In the following sections, the mechanism of HAT is introduced, and DTD and LwPRM are described.

### 3.1 MECHANISM OF HAT

HAT requires every layer to have a task embedding, $e_l^t$, to control the gradient of the layer's parameters. Each layer's mask, $a_l^t$, is computed from $a_l^t = \sigma(se_l^t)$, and each layer's output, $o_l$, is replaced with $h_l = o_l \otimes a_l^t$, where $\sigma$ is an activation function (e.g., sigmoid), $s$ is a positive scaling parameter, and $\otimes$ denotes elementwise multiplication. To preserve the information that is obtained in previous tasks, after learning task $t$, HAT computes an accumulated mask, $a_l^{\leq t}$, as follows:

$$a_l^{\leq t} = \max\left(a_l^t, a_l^{\leq t-1}\right), \tag{1}$$

using elementwise maximum and the all-zero vector for $a^{\leq 0}$. Then, in learning task $(t+1)$, HAT reduces the gradients of parameters based on the accumulated mask:

$$g'_{l,ij} = \left[1 - \min\left(a_{l,i}^{\leq t}, a_{l-1,j}^{\leq t}\right)\right] g_{l,ij}, \tag{2}$$

where unit indices $i$ and $j$ represent the $l$-th and $(l-1)$-th layer outputs, respectively. $g_{l,ij}$ denotes its gradient. Additionally, HAT utilizes two more tricks to stabilize learning. First, in learning, scaling parameter $s$ is linearly annealed as follows:

$$s = \frac{1}{s_{\max}} + \left(s_{\max} - \frac{1}{s_{\max}}\right)\frac{b-1}{B-1}, \tag{3}$$

where $s_{\max}$ is a hyper parameter, the value of which is a large positive number, and $b$ and $B$ denote the batch index and the total number of batches, respectively. In testing, $s_{\max}$ is used instead of $s$. Second, to alleviate the side effect on embedding gradient compensation, the formula below is used:

$$q'_{l,i} = \frac{s_{\max}\left[\cosh(se_{l,i}^t) + 1\right]}{s\left[\cosh(e_{l,i}^t) + 1\right]} q_{l,i}, \tag{4}$$

where $q_{l,i}$ represents the gradient that corresponds to $e_{l,i}^t$ and is replaced with $q'_{l,i}$.

### 3.2 DISSIMILAR TASK DETECTION (DTD)

To balance knowledge transfer and the prevention of forgetting effectively, it is important to determine which tasks are similar and can be transferred to the current task and which tasks are dissimilar and should be blocked so that they are not forgotten. To address this issue, we focus on the task embeddings that are learned through the HAT mechanism. Since the mechanism employs an accumulated mask that reduces the gradients of the model's parameters, the task embeddings can be more easily updated than the model's parameters. Therefore, the task embeddings are expected to provide an informative representation of the tasks and their relations with one another. Specifically, we adopt an unsupervised clustering method for judging task similarity.

First, the model follows the approach of HAT in learning task $t$ by reducing the gradients of the parameters according to Equation 1 and Equation 2, namely, as illustrated in Figure 1 at the bottom, $a_l^{\leq t-1}$ is used to reduce the gradients for all the previous tasks in the DTD phase. After learning task $t$, optimized $\left\{e_l^{1:t}\right\}$ are obtained. Using a clustering method, the set of previous tasks with embeddings that do not belong to the same cluster as task $t$ are found, and they are regarded as dissimilar tasks, which are represented by $\mathcal{D}_l^t$. Although any kind of clustering method can be used, we exploit X-means (Pelleg et al., 2000) because it does not require the number of clusters as input; instead, it searches for the optimal number of clusters based on the Bayesian information criterion (BIC) by applying K-means recursively.

#### 3.2.1 TWO STAGE OPTIMIZATION

Additionally, we introduce a new optimization that proceeds in two stages. The model consists of two parts: the first is the feature extractor that is to be shared across tasks, and the other is the classifier that is built for each task (so-called, a *multi-headed* model). Therefore, we hypothesize that if both the feature extractor and classifier are optimized simultaneously, the information that represents the difference across tasks and can be used as a clue for task similarity comparison may

Table 1: The statistics of each task.

| Dataset | # of Tasks | # of Classes | # of Trainings | # of Validations | # of Tests |
|---|---|---|---|---|---|
| CIFAR100-10T | 10 | 10 | 4500 | 500 | 1000 |
| EMNIST-10T | 10 | 5 (Last three: 4) | 500 | 200 | 200 |
| F-CelebA-10T | 10 | 2 | 400 | 40 | 80 |
| F-EMNIST-10T | 10 | 62 | 1240 | 310 | 310 |

be dispersed both into not only the task embeddings but also the classifier, which may degrade the performance of our approach. To ensure maximum sharing in the task embedding inside the feature extractor, which will facilitate similarity comparison, we first freeze the feature extractor and learn only the classifier (i.e., depicted as "1st optimization" in Algorithm 1); then, we optimize both the feature extractor and classifier simultaneously, from which task embeddings are obtained for the clustering (i.e., "2nd optimization").

### 3.3 Learning with partially relaxed masks (LwPRM)

Although the accumulated mask, $a_l^{\leq t}$, in HAT plays a large role in preventing forgetting, it may also restrain knowledge transfer among similar tasks because the gradients of the parameters are reduced, regardless of task similarity. Thus, following HAT, we extend it so that it can promote knowledge transfer by introducing a new mechanism named "partially relaxed masks" (PRM).

PRM employs masks per previous task like HAT; however, not all masks are used to reduce the gradients of the parameters according to the task similarity. Instead, PRM accumulates only the masks that belong to previous dissimilar tasks so that it can prevent forgetting only for dissimilar tasks while maintaining opportunities for improvement for other tasks at the same time. Namely, the accumulated mask focuses only on dissimilar tasks and is partially relaxed to keep the parameters for other tasks updatable, which can enhance knowledge transfer. Given that we know which tasks are dissimilar to the current new task by clustering, Equation 1 and Equation 2 are replaced as follows:

$$p_l^{\leq t} = \max\left(\left\{a_l^i | i \in \mathcal{D}_l^t, i \leq t\right\}\right), \tag{5}$$

$$g'_{l,ij} = \left[1 - \min\left(p_{l,i}^{\leq t}, p_{l-1,j}^{\leq t}\right)\right] g_{l,ij} \tag{6}$$

As illustrated in Figure 1, the LwPRM phase exploits $p_l^{\leq t-1}$ instead of $a_l^{\leq t-1}$ when learning task $t$.

## 4 Experiments

### 4.1 Datasets

Following the standard continual learning setting, we use the following four kinds of datasets to evaluate the performance in terms of both prevention of forgetting and knowledge transfer. The datasets are split into multiple tasks, and the statistics of each task are presented in Table 1.

**Dissimilar tasks.** CIFAR100 (Krizhevsky et al., 2009), which contains 100 classes, is split into 10 tasks, each of which has 10 classes; the dataset is named CIFAR100-10T. EMNIST (Cohen et al., 2017), which contains 47 classes, is split into 10 tasks, each of which has 5 (the last three tasks have 4) classes; the dataset is named EMNIST-10T. These datasets are expected to be sensitive to forgetting as each task has different classes and there are few relations or similarities across tasks.

**Similar tasks.** F-CelebA (Liu et al., 2015) is a dataset that contains face images of celebrities and labels that indicate whether or not they are smiling. Different celebrities correspond to different tasks. We use 10 celebrities in the experiments; the dataset is named F-CelebA-10T. F-EMNIST (Liu et al., 2015) is a dataset that contains 62 classes of character images handwritten by different users. We use the images that correspond to 10 writers; the dataset is named F-EMNIST-10T. These tasks are supposed to have shared knowledge across tasks as each task has the same set of labels and the data that are naturally similar.

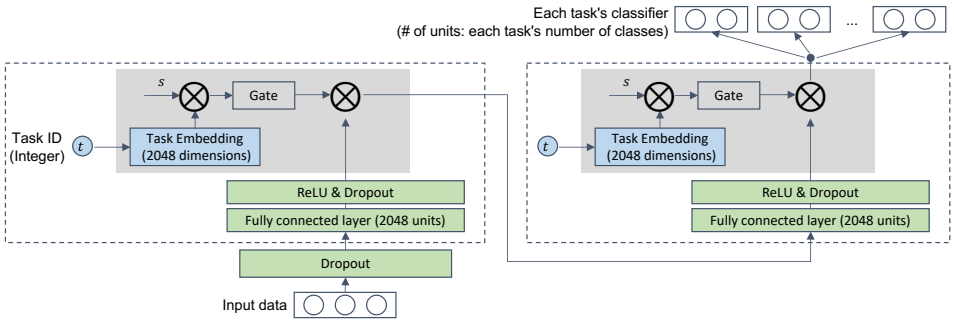


Figure 2: Proposed networks that are used in the experiments. Only the forward path is shown.

We conduct experiments with three kinds of sequences that combine at most two different tasks in random order, as presented in Table 2, Table 3, and Table 4: **only dissimilar tasks** - #1 and #2, **only similar tasks** - #3 and #4, and **mixed dissimilar and similar tasks** - #5 and #6.

### 4.2 Baselines

We compare PRM with several classic and latest continual learning methods that can work as TIL systems, namely, EWC (Kirkpatrick et al., 2017), ACL (Ebrahimi et al., 2020), PathNet (Fernando et al., 2017), SupSup (Wortsman et al., 2020), HyperNet (von Oswald et al., 2020), HAT (Serra et al., 2018) and CAT (Ke et al., 2020). Hereafter, PNT, SS, HYP, and EHAT denote PathNet, SupSup, HyperNet, and EWC in the HAT package, respectively. Since HAT focuses only on preventing forgetting and does not have any mechanism for knowledge transfer, it is expected to perform poorly on similar tasks. To the best of our knowledge, CAT is the only approach that focuses on a mixed sequence of similar and dissimilar tasks; however, CAT takes much longer to learn. Also, we prepare two reference methods: naive continual learning (NCL) and single-task learning (STL). NCL learns a new task without considering previous tasks; thus, severe forgetting is expected to occur for dissimilar tasks. STL learns all the tasks at once. Although it does not follow the continual learning scenario, it is expected to be the upper bound, only for dissimilar tasks.

### 4.3 Implementation details

The input images go through two fully connected layers passing ReLU and dropout layers, as shown in Figure 2. The networks are optimized by minimizing the last classifier's cross-entropy loss using SGD. The learning rate starts from $0.025$ and is gradually reduced until it reaches $0.001$. When there is no improvement in the validation loss for 5 epochs, the training stops. The batch size and $s_{\max}$ are set to $64$ and $400$, respectively. Other hyper parameters, such as the rate of dropout and the strength of regularization, are searched over 20 trials using Tree-structured Parzen Estimator (TPE) (Bergstra et al., 2011), which is implemented in Optuna[1] (Akiba et al., 2019). The baselines are evaluated on the original code with modifications while aligning with our setting as much as possible.

### 4.4 Metrics

**Accuracy (Acc)**: The average accuracy for all tasks after learning them, where the model is optimized with the best hyper parameters that are found in the search. **Parameter Sensitivity (PS)**: The standard deviation of "Acc" with varied hyper parameters over 20 searches. **Forward Transfer (FWT)**: The test accuracy of task $i$ just after learning task $i$ is compared to the accuracy for task $i$ by STL (Single Task Learning), which is expressed as $1/T \sum_t^T (\alpha_t^t - \tilde{\alpha}_t)$, where $\alpha_i^j$ denotes the evaluated accuracy of task $i$ after learning task $j$, $\tilde{\alpha}_i$ denotes the test accuracy for task $i$ by STL, and $T$ denotes the total number of tasks. **Backward Transfer (BWT)**: The average of improvement from each task's initial accuracy to the final accuracy, which is represented by $1/T \sum_t^T (\alpha_t^{\mathrm{T}} - \alpha_t^t)$ (Lopez-Paz & Ranzato, 2017). Negative values represent that forgetting occurs. **Computation Time (CT)**: The total computation time for learning all the tasks, which is measured in seconds.

[1] `https://optuna.org/`

Table 2: Results for sequences of only dissimilar tasks (average over the three sequences).

| | (#1) EMNIST-10T | | | | (#2) CIFAR100-10T | | | |
|---|---|---|---|---|---|---|---|---|
| | Acc | PS | CT | FWT / BWT | Acc | PS | CT | FWT / BWT |
| (STL) | 0.929 | 0.6% | 45 | - / - | 0.612 | 1.4% | 142 | - / - |
| NCL | 0.879 | 0.3% | 26 | -1.7% / -3.4% | 0.534 | 1.1% | 123 | -2.3% / -5.5% |
| ACL | 0.902 | 0.5% | 916 | -2.6% / -0.1% | 0.508 | 0.6% | 6328 | -10.3% / -0.1% |
| PNT | **0.910** | 0.3% | 43 | -2.0% / 0.0% | 0.571 | 0.5% | 394 | -4.1% / 0.0% |
| SS | 0.829 | 11.9% | 172 | -5.6% / -4.4% | 0.462 | 11.2% | 2800 | -10.8% / -4.2% |
| HYP | 0.822 | 10.9% | 1105 | -10.7% / -0.1% | 0.219 | 3.4% | 10825 | -38.8% / -0.5% |
| HAT | 0.905 | 0.3% | 101 | -2.4% / 0.0% | 0.582 | 0.8% | 912 | -3.0% / 0.0% |
| EHAT | 0.899 | 0.3% | 187 | -3.1% / 0.0% | 0.578 | 0.6% | 1011 | -3.4% / 0.0% |
| CAT | 0.907 | 0.3% | 1276 | -2.2% / 0.0% | **0.587** | 0.7% | 6018 | -2.5% / 0.0% |
| **PRM** | 0.897 | 0.4% | 80 | -2.3% / -1.0% | 0.582 | 0.9% | 635 | -2.8% / -0.1% |

Table 3: Results for sequences of only similar tasks (average over the three sequences).

| | (#3) F-EMNIST-10T | | | | (#4) F-CelebA-10T | | | |
|---|---|---|---|---|---|---|---|---|
| | Acc | PS | CT | FWT / BWT | Acc | PS | CT | FWT / BWT |
| (STL) | 0.717 | 3.2% | 126 | - / - | 0.823 | 0.6% | 15 | - / - |
| NCL | 0.654 | 10.0% | 48 | -5.3% / -0.9% | 0.820 | 1.2% | 14 | -3.7% / 3.4% |
| ACL | 0.043 | 0.9% | 1633 | -66.4% / -0.9% | 0.695 | 2.4% | 628 | -14.3% / 1.5% |
| PNT | 0.572 | 15.7% | 178 | -14.5% / 0.0% | 0.716 | 1.1% | 32 | -10.6% / 0.0% |
| SS | 0.456 | 14.6% | 1022 | -14.3% / -11.7% | 0.780 | 9.5% | 440 | -5.0% / 0.8% |
| HYP | 0.060 | 1.5% | 3313 | -65.5% / -0.2% | 0.525 | 1.5% | 1495 | -26.6% / -3.2% |
| HAT | 0.655 | 3.7% | 245 | -6.2% / 0.0% | 0.759 | 1.1% | 70 | -6.3% / 0.0% |
| EHAT | 0.655 | 3.8% | 497 | -6.1% / 0.0% | 0.769 | 0.8% | 110 | -5.3% / 0.0% |
| CAT | 0.643 | 2.1% | 2171 | -7.4% / 0.0% | 0.781 | 0.7% | 707 | -4.1% / 0.0% |
| **PRM** | **0.657** | 3.3% | 176 | -5.7% / -0.2% | **0.796** | 1.4% | 54 | -5.5% / 2.8% |

Table 4: Results for mixed sequences (average over the three sequences).

| | (#5) EMNIST-10T & F-EMNIST-10T | | | | (#6) CIFAR100-10T & F-CelebA-10T | | | |
|---|---|---|---|---|---|---|---|---|
| | Acc | PS | CT | FWT / BWT | Acc | PS | CT | FWT / BWT |
| (STL) | 0.791 | 1.4% | 113 | - / - | 0.629 | 1.1% | 254 | - / - |
| NCL | 0.728 | 5.0% | 87 | -5.6% / -0.6% | 0.540 | 1.0% | 154 | -3.1% / -5.9% |
| ACL | 0.370 | 9.3% | 5437 | -41.6% / -0.5% | 0.521 | 0.4% | 7515 | -10.9% / 0.0% |
| PNT | 0.628 | 4.7% | 281 | -16.3% / 0.0% | 0.556 | 0.7% | 425 | -7.3% / 0.0% |
| SS | 0.572 | 16.2% | 586 | -7.6% / -14.3% | 0.461 | 10.5% | 1865 | -11.1% / -5.7% |
| HYP | 0.322 | 6.0% | 6439 | -46.2% / -0.7% | 0.199 | 1.7% | 19973 | -42.2% / -0.8% |
| HAT | 0.721 | 3.6% | 357 | -7.0% / 0.0% | 0.572 | 0.5% | 891 | -5.8% / 0.0% |
| EHAT | 0.728 | 6.9% | 854 | -6.3% / 0.0% | 0.572 | 0.6% | 1801 | -5.8% / 0.0% |
| CAT | **0.737** | 7.2% | 9851 | -5.3% / 0.0% | 0.581 | 0.8% | 22810 | -4.8% / 0.0% |
| **PRM** | 0.720 | 3.4% | 310 | -6.8% / -0.3% | **0.582** | 0.6% | 642 | -4.2% / -0.6% |

### 4.5 RESULTS

The results are presented in Table 2, Table 3, and Table 4. Each row presents to the average results over the same set of three random task sequences.

**Only dissimilar tasks (#1 and #2):** NCL causes severe forgetting (-3.4% and -5.5%, as presented in the column of BWT). PRM achieves competitive accuracy compared to the baselines (especially PNT and CAT) without much forgetting. Also, PRM requires only one-tenth the computation time of CAT in #2. Among the baselines, PRM achieves almost the best score in the second shortest time.

**Only similar tasks (#3 and #4):** As the tasks are similar, NCL does not cause much forgetting, and even improves task by task, as presented in the column of BWT. In both sequences, PRM outperforms all the baselines without forgetting and even with significant backward transfer, as shown in #4. As in the case of only dissimilar tasks, which is presented in Table 2, PRM's efficiency is only behind PNT, but PNT's performance is markedly lower than PRM in these only similar tasks.

**Mixed dissimilar and similar tasks (#5 and #6):** While PRM achieves the best performance in #6, it underperforms CAT in #5. However, the performance of CAT is highly sensitive to the hyper parameters, according to the PS value of 7.2%, while PRM has 3.4% PS, which is the most stable among the baselines. Moreover, CAT requires a long computation time as shown in Figure 3, where the computation time for each task and the accumulated time for all tasks in #6 are plotted. According to these figures, CAT takes much longer than the others, and learning more tasks requires more time. Based on these observations, in practice, it is difficult to use CAT when there is a large amount of data or many tasks, and due to its unstable performance, tuning is essential. In contrast, PRM needs much less computation time, e.g., 1/35 that of CAT when learning 20 tasks, and its performance is also stable with varied hyper parameters.

**General results:** As stated above, the performance of NCL sometimes degrades along tasks when tasks are dissimilar, as shown especially in #1 and #2. While ACL and PNT perform strongly on dissimilar tasks, as presented in Table 2, they fail to learn correctly on similar tasks. Although SS can deal with both dissimilar and similar tasks, its performance on mixed sequences is very poor compared to the others, as presented in Table 4. HYP does not work well in this experimental setting. As HAT was originally proposed only for preventing forgetting, it achieves almost the best performance on dissimilar tasks; however, its performance on similar tasks and mixed sequences is not sufficient. While CAT can handle similar, dissimilar, and mixed sequences well, its long computation time cannot be ignored for practical applications and tuning. PRM performs competitively with the baselines in much less computation time, and in #3, #4, and #6, PRM outperforms all the other models. From these results, we conclude that PRM achieves state-of-the-art performance in terms of balancing accuracy and learning efficiency.

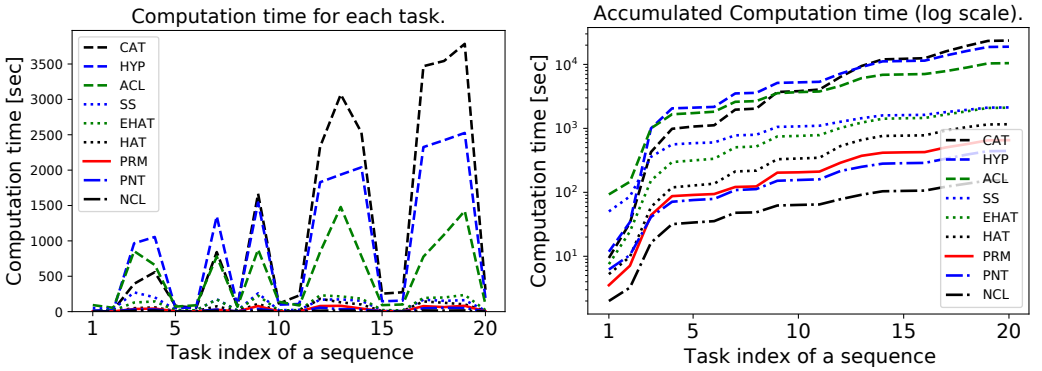


Figure 3: Computation times for one sequence of #6. The right figure is plotted in log scale. CAT, HYP and ACL take more time, while PRM requires less time.

### 4.6 Ablation Study

We check how much the DTD phase influences the total performance, as LwPRM completely depends on its behavior. The results are presented in Table 5. PRM(S) and PRM(D) indicate the cases in which all previous tasks are regarded as similar and dissimilar, respectively, instead of actual clustering. PRM(T) represents the cases in which the types of tasks are given and used as a replacement for clustering (e.g., when learning a new task of CIFAR100-10T, the model can tell which previous tasks are from CIFAR100-10T, and these tasks are regarded as similar, while other tasks are recognized as dissimilar). When the sequence of tasks consists only of tasks from the same dataset (#1 to #4), the behavior of PRM(T) is the same as that of PRM(S). Additionally, we check the effect of the two stage optimization, as shown in the column of "w/o 2SO", where PRM is learned without it (i.e., both the feature extractor and classifier are optimized simultaneously in the DTD phase).

Table 5: Results of ablation experiments (average over the three sequences).

| | PRM(S) | PRM(D) | PRM(T) | PRM | w/o 2SO |
|---|---|---|---|---|---|
| (#1) EMNIST-10T | 0.882 | **0.904** | - | 0.897 | 0.897 |
| (#2) CIFAR100-10T | 0.562 | 0.581 | - | **0.582** | 0.579 |
| (#3) F-EMNIST-10T | 0.633 | 0.654 | - | **0.657** | 0.658 |
| (#4) F-CelebA-10T | **0.825** | 0.759 | - | 0.796 | 0.775 |
| (#5) EMNIST-10T & F-EMNIST-10T | 0.713 | **0.731** | 0.720 | 0.720 | 0.721 |
| (#6) CIFAR100-10T & F-CelebA-10T | 0.568 | 0.571 | 0.567 | **0.582** | 0.586 |

Notably, PRM(T), where the types of tasks are utilized, does not always show the best performance. Instead, PRM, which employs task embeddings through clustering, can exploit not explicit but implicit relationships across tasks, thereby resulting in similar or better performance than PRM(T). Conversely, it is reasonable that PRM(D), which regards all the tasks as dissimilar, has the best performance in #1 and PRM(S) performs the best in #4. Although in #5, PRM(D) performs the best, the performance differences among other the types of PRM are small. Generally, PRM treats task embeddings via clustering to handle various types of data sequences. Moreover, it is demonstrated by comparing "PRM" and "w/o 2SO" that the two stage optimization contributes PRM's performance especially in #4, which is consistent with our hypothesis that it can facilitate the similarity comparison.

### 4.7 Limitations

PRM has several limitations. First, while PRM tries to open up masks for knowledge transfer and achieves better transfer in similar tasks as shown in Table 3, this mechanism is implicit; thus, there is room to employ another explicit mechanism, such as attention, to enhance it. PRM is a direct extension of HAT, and other techniques can be jointly exploited; however, it is still unclear which combinations perform best. Second, as presented in Table 5, the current DTD mechanism is sometimes outperformed by PRM(S) and PRM(D), which may indicate that with a more effective method for utilizing task embeddings, it will be possible to improve its performance. The effectiveness of exploiting task embeddings is proven in most cases; however, a more effective approach needs to be developed.

## 5 Conclusion

To extend HAT so that it can not only overcome catastrophic forgetting but also transfer knowledge, the current study makes two contributions. First, we discover that the task embeddings that are optimized by parameter masking approaches, such as HAT, provide an informative representation for the task similarity. Second, we propose a new approach, namely PRM, that controls which parameters should be blocked or relaxed based on the task similarity that is obtained via clustering on the task embeddings. The experimental results show that PRM performs competitively compared to the latest baselines with much less computation time, and it has the potential of better accuracy according to the ablation study. PRM is a direct extension of HAT with almost no severe side effects, and other techniques, such as attention, can be jointly exploited for improved performance.

As AI-embedded systems are being used to solve many tasks, overcoming *catastrophic forgetting* while exploiting knowledge transfer remains a large problem in continual learning. To address this issue, task similarity may provide information that indicates which tasks are worth being transferred to new tasks and which are not, and systems do not always have such information in advance of learning. Additionally, in realistic use cases, learning should require as little computation time as possible because the surrounding environment drastically changes from one task to another; otherwise, performance cannot be maintained. Our proposed approach exploits task embeddings to identify implicit relationships across tasks with short computation time and achieves both prevention of forgetting and knowledge transfer simultaneously. Hence, our proposed approach realizes the best of both worlds, namely, comparable accuracy to state-of-the-art methods and high efficiency, which will be helpful for various continual learning applications.

## REFERENCES

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A Next-generation Hyperparameter Optimization Framework. In *Proc. of SIGKDD*, 2019.

James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. 2011.

Zalán Borsos, Mojmír Mutný, and Andreas Krause. Coresets via Bilevel Optimization for Continual Learning and Streaming. In *Proc. of NeurIPS*, 2020.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient Lifelong Learning with A-GEM. In *Proc. of ICLR*, 2019.

Arslan Chaudhry, Albert Gordo, Puneet K Dokania, Philip Torr, and David Lopez-Paz. Using Hindsight to Anchor Past Knowledge in Continual Learning. In *Proc. of AAAI*, 2021.

Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. EMNIST: Extending MNIST to handwritten letters. In *Proc. of IJCNN*, 2017.

Matthias Delange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Greg Slabaugh, and Tinne Tuytelaars. A Continual Learning Survey: Defying Forgetting in Classification Tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

Sayna Ebrahimi, Franziska Meier, Roberto Calandra, Trevor Darrell, and Marcus Rohrbach. Adversarial Continual Learning. In *Proc. of ECCV*, 2020.

Chrisantha Fernando, Dylan Banarse, Charles Blundell, Yori Zwols, David Ha, Andrei A. Rusu, Alexander Pritzel, and Daan Wierstra. PathNet: Evolution Channels Gradient Descent in Super Neural Networks, 2017.

Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. An Empirical Investigation of Catastrophic Forgetting in Gradient-Based Neural Networks. In *Proc. of ICLR*, 2014.

Hu, Wenpeng and Lin, Zhou and Liu, Bing and Tao, Chongyang and Tao, Zhengwei and Ma, Jinwen and Zhao, Dongyan and Yan, Rui. Overcoming Catastrophic Forgetting for Continual Learning via Model Adaptation. In *Proc. of ICLR*, 2018.

Khurram Javed and Martha White. Meta-Learning Representations for Continual Learning. In *Proc. of NeurIPS*, 2019.

Zixuan Ke, Bing Liu, and Xingchang Huang. Continual Learning of a Mixed Sequence of Similar and Dissimilar Tasks. In *Proc. of NeurIPS*, 2020.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming Catastrophic Forgetting in Neural Networks. *Proc. of National Academy of Sciences*, 2017.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning Multiple Layers of Features from Tiny Images. Technical report, 2009.

Zhizhong Li and Derek Hoiem. Learning without Forgetting. In *Proc. of ECCV*, 2016.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep Learning Face Attributes in the Wild. In *Proc. of ICCV*, 2015.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient Episodic Memory for Continual Learning. In *Proc. of NeurIPS*, 2017.

German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual Lifelong Learning with Neural Networks: A Review. *Neural Networks*, 2019.

Dan Pelleg, Andrew W Moore, et al. X-means: Extending K-means with Efficient Estimation of the Number of Clusters. In *Proc. of ICML*, 2000.

Lorenzo Pellegrini, Gabriele Graffieti, Vincenzo Lomonaco, and Davide Maltoni. Latent Replay for Real-Time Continual Learning. In *Proc. of IROS*, 2020.

Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. Learning to Learn without Forgetting by Maximizing Transfer and Minimizing Interference. In *Proc. of ICLR*, 2019.

Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive Neural Networks, 2016.

Joan Serra, Didac Suris, Marius Miron, and Alexandros Karatzoglou. Overcoming Catastrophic Forgetting with Hard Attention to the Task. In *Proc. of ICML*, 2018.

Pravendra Singh, Vinay Kumar Verma, Pratik Mazumder, Lawrence Carin, and Piyush Rai. Calibrating CNNs for Lifelong Learning. In *Proc. of NeurIPS*, 2020.

Johannes von Oswald, Christian Henning, Jo ao Sacramento, and Benjamin F. Grewe. Continual Learning with Hypernetworks. In *Proc. of ICLR*, 2020.

Mitchell Wortsman, Vivek Ramanujan, Rosanne Liu, Aniruddha Kembhavi, Mohammad Rastegari, Jason Yosinski, and Ali Farhadi. Supermasks in Superposition. In *Proc. of NeurIPS*, 2020.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual Learning Through Synaptic Intelligence. In *Proc. of ICML*, 2017.