# MAML-CL: Edited Model-Agnostic Meta-Learning for Continual Learning

Anonymous ACL submission

## Abstract

Continual learning (CL) exhibits a learning ability to well-learn all sequentially seen tasks drawn from various domains. Yet, existing sequential training methods fail to consolidate learned knowledge from earlier tasks due to data distribution shifts, hereby leading to catastrophic forgetting. We devise an optimization-based meta learning framework for CL in accordance with MAML, where query samples are edited for generalization of learned knowledge. We conduct extensive experiments on text classification in a low resource CL setup, where we downsize training set to its 10%. The experimental results demonstrate the superiority of our method in terms of stability, fast adaptation, memory efficiency and knowledge retention across various domains.

## 1 Introduction

Existing sequential learning poses a challenge. Weights constantly vary along with the change of probability distribution, in which important information from earlier tasks can be easily erased or overwritten by information from the latest tasks. Consequently, catastrophic forgetting (McCloskey and Cohen, 1989) occurs and harms performance on preceding tasks. To address catastrophic forgetting, a continual learning (CL) method aims to guarantee the stability of handling various tasks that have been learned, while showing plasticity on the novel domain via previously acquired knowledge. The majority of CL methods tackle catastrophic forgetting in attempts to realise generalization or/and understand uncertainty using deep neural networks.

Conventional machine learning improves decision making by training on multiple data instances. Whereas, meta learning learns an optimal learning algorithm by ingesting multiple learning episodes. Meta learning has facilitated the recent work of CL mainly by Model-Agnostic Meta-Learning (MAML) (Finn et al., 2017) model. Existing CL models exploit MAML for fast adaptation to a novel domain, coupling with extra means of preventing forgetting, e.g., experience replay (Wang et al., 2020; Holla et al., 2020; Joseph and Balasubramanian, 2020; Ho et al., 2021), introducing regularisation term in loss (Acar et al., 2021).

We utilise meta learning to address CL from a different angle. We argue that the nature of continual learning can be interpreted as a form of meta learning. In a meta learning process, an inner loop algorithm models over a task, while an outer loop algorithm harnesses the optimization of the inner loop algorithm as a result of realising an outer objective. Limitation on shifting of weights in CL is imperative to alleviate catastrophic forgetting, referring to a major research problem. In this case, we adapt meta learning framework to CL. Intuitively, the outer algorithm constrains the task-specific learning algorithm by governing its optimisation process on a novel domain. The outer objective can be thereby defined as the generalization of entire learned knowledge from preceding tasks, such that the model is hardly prone to catastrophic forgetting.

Recent literature (Ho et al., 2021) has manifested that existing CL methods have the instability issue, where model performance severely depends on input sets orders. Such an issue yields a hurdle. The deficiency of existing CL models can be easily masked or neglected. Therefore, we conduct extensive experiments on Yelp, AGNews and Amazon datasets (Zhang et al., 2015) to testify the stability of MAML-CL. Additionally, further analysis on MAML-CL exhibits its outstanding performance as a CL learner.

We summarize our main contributions as:

- We fully exploit the potential of meta learning for CL. We propose a model, namely MAML-CL, to address CL problems by simply editing query information of MAML.

- MAML-CL enhances the effectiveness of MAML for knowledge retention across various domains. Under the same FOMAML framework, our model outperforms recent CL models by a large margin.

- With the same sample selection criteria, MAML-CL realizes sample efficiency and further optimizes memory footprints.

- In a low resource setup, we prove the superiority of MAML-CL in terms of stability, fast adaptation, forgetting mitigation and memory efficiency.

## 2 Related Work

Existing CL methods can be categorised into two mainstreams, i.e., memory replay-based approaches (de Masson d'Autume et al., 2019; Chaudhry et al., 2019) and regularization-based approaches (Aljundi et al., 2018; Huang et al., 2021). In general, memory replay-based methods address catastrophic forgetting by revisiting old samples. Regularization-based methods employ gradients or parameters constraints to achieve generalization, thereby retaining knowledge. Due to the complexity of deep neural networks, memory replay-based approaches are broadly deemed as a plausible means for continual learning in NLP.

Recently, meta learning has been introduced into CL models, considering its ability of fast adaptation and knowledge transfer. Existing works employ MAML to improve initial parameters of the model, such that it can fast adapt to various domains with few learning samples (Holla et al., 2020; Ho et al., 2021) or find an optimal initialization state to perform episodic experience rehearsal (Wang et al., 2020). Additionally, Joseph and Balasubramanian (2020) uses preceding task-specific priors from meta distribution to replay previous parameters and consolidate the CL model. Reptile (Nichol et al., 2018) is also leveraged in some CL models to regularize the objective of experience replay (Riemer et al., 2019) or meta updates parameters via augmented training set (Obamuyide and Vlachos, 2019).

## 3 Problem Formulation

A CL model $f$ with a learnable parameters $\theta$ over a parameters space $\Theta$ sequentially ingests a stream of labeled samples $\{(x, y)\}$ drawn from various data distributions over one pass. Concretely, it considers a sequence of $K$ tasks $\{\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_K\}$. Given a task $\mathcal{T}_k$ and a ground truth label set $L_k$, the initial parameters in $\mathcal{T}_k$, namely $\theta_k$, is a parameters set that have been finetuned in the last task $\mathcal{T}_{k-1}$, i.e., $\tilde{\theta}_{k-1}$. Ideally, we expect a CL learner $f$: (1) to update parameters from $\tilde{\theta}_{k-1}$ to $\tilde{\theta}_k$ for $\mathcal{T}_k$, such that the loss $\mathcal{L}_{\mathcal{T}_k}$ on the set of labeled instances $\{(x_k, y_k)\}$ is minimal,

$$\tilde{\theta}_k = \arg\min_{\theta_k \in \Theta} \mathcal{L}_{\mathcal{T}_k}(\theta_k), \quad \text{where} \quad \theta_k = \tilde{\theta}_{k-1} \tag{1}$$

(2) to perform well with the learned $\tilde{\theta}_k$ on all preceding tasks $\{\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_{k-1}\}$ without the need of presenting all previously seen training data.

Assuming that all tasks are equally important, the objective is thereby minimising the expected risk of $|k|$ tasks that have seen so far, with respect to $\tilde{\theta}_k$,

$$\min_{\tilde{\theta}_k} \sum_{i=1}^{k} \mathbb{E}_{\mathcal{T}_i}[\mathcal{L}_{\mathcal{T}_i}(\tilde{\theta}_k)] \tag{2}$$

CL setup allows models to preserve a certain amount of training samples from previous tasks. Whereas, optimizing memory footprint is also regarded as one major research problem in CL. Therefore, we limit the memory budget of $f$ to a constant size $B$. That is, at step $k$, we allow the learner $f$ to only store samples from $\{\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_{k-1}\}$ with the amount less than or equals to $B$.

### 3.1 Online Meta Learning

A meta learner is able to perform fast adaptation by learning an optimal initial state of an algorithm. Given a task $\mathcal{T}$, a set of initial parameters $\phi$ is over a parameters space $\Phi$. We expect $\phi$ that facilitates the model to yield a low loss after $m$ updates in $\mathcal{T}$. That is,

$$\min_{\phi} \mathbb{E}_{\mathcal{T}}[\mathcal{L}_{\mathcal{T}}(U^m_{\mathcal{T}}(\phi))] \tag{3}$$

where $U^m_{\mathcal{T}}$ is the update operation that performs $m$ times gradient-based updates on parameters $\phi$, using samples drawn from $p(\mathcal{T})$. In MAML, the objective is to learn an algorithm with optimal initial parameters $\phi^*$ such that the model efficiently solves specified problems through example problem instances. Thereby, test samples that specified problems are also required for loss computation, referring to as query samples $Q$. While, the training samples are known as support samples $S$ in meta learning. The training objective is,

$$\min_{\phi} \mathbb{E}_{\mathcal{T}}[\mathcal{L}_{\mathcal{T},Q}(U^m_{\mathcal{T},S}(\phi))] \tag{4}$$

In an online meta learning setup, each episode contains $m$ batches as the support set and each task has multiple episodes as multiple training iterations. Note that the initial parameters $\phi_k$ is $\phi^*_{k-1}$, i.e., the optimal initial state derived after learning $\mathcal{T}_{k-1}$. In an inner loop optimization process, MAML performs $m$ steps of SGD on parameters $\phi_k$.

$$\begin{aligned}\tilde{\phi}_k &= U^m_{\mathcal{T}_k,S_k}(\phi_k) \\ &= \phi_k - \alpha \nabla_{\phi_k} \mathcal{L}_{\mathcal{T}_k}(\phi_k) \\ &= \phi_k - \alpha \nabla_{\phi_k} \sum_{i=1}^{m} \mathcal{L}(f_{\phi_k}(S^i_k))\end{aligned} \tag{5}$$

where $\alpha$ denotes the step size as a hyperparameter. $\phi_k$ is finetuned by gradients of loss on the support set $S_k$ for task-specific learning on $\mathcal{T}_k$. Then, the updated $\tilde{\phi}_k$ are further optimised using the query set $Q_k$ to achieve the meta objective.

### 3.2 Catastrophic Forgetting

Performance degradation caused by catastrophic forgetting occurs in existing CL models, mainly due to the learning goal as shown in Equation 2. The optimizee for a task $\mathcal{T}_k$ in CL is generally the parameters that updated for current task $\mathcal{T}_k$, $\tilde{\theta}_k$. The optimization direction indicates such a learning process solely focus on minimising expected loss for $\mathcal{T}_k$, resulting in $\tilde{\theta}_k$ is heavily skewed towards the probability distribution $p(\mathcal{T}_k)$. It is harmful to sequential learning by neglecting the next optimal update step, leading to catastrophic forgetting.

Recently, some CL methods (Riemer et al., 2019; Obamuyide and Vlachos, 2019; Holla et al., 2020; Wang et al., 2020; Ho et al., 2021) exploit meta learning. In general, query set for existing CL methods (Holla et al., 2020; Wang et al., 2020; Ho et al., 2021) under MAML framework contains examples merely from current task, and episodically adds previously seen samples to diminish forgetting. Consequently, the meta objective mainly focuses on the expected risk of $\mathcal{T}_k$ and pays a little attention on preceding tasks, namely $\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_{k-1}$, thereby failing to achieve CL objective in Equation 2 and inducing catastrophic forgetting.

## 4 Edited MAML for Continual Learning

We propose a meta learning framework for CL, namely MAML-CL. Specifically, we utilise FO-MAML, a simple well-known parametric fast adaptation method. MAML-CL perform query information editing in accordance with prototypes-guided sample selection criteria, i.e., the choice of representative examples to achieve the generalisation for all task that have been learned. In such a way, we address CL problems.

**Query Information Editing** To retain consistency on objective loss under MAML framework with CL, query information is prominent. Hence, we expect query set $Q_k$ to contain examples that generalize tasks drawn from various probability distributions of $\mathcal{T} = \{\mathcal{T}_1, ..., \mathcal{T}_{k-1}, \mathcal{T}_k\}$. To optimize memory footprint, efficient sample selection criteria should opt for representative samples for each task. Such that, $Q_k = \sum_{i=1}^{k} Q_{k,i}$ where $Q_{k,i}$ is a set of representative examples for $\mathcal{T}_i$ while learning $\mathcal{T}_k$. The meta-objective with respect to $\phi_k$ is,

$$\begin{aligned}&\min_{\phi_k} \mathbb{E}_{\mathcal{T}}[\mathcal{L}_{\mathcal{T},Q_k}(U^m_{\mathcal{T}_k,S_k}(\phi_k))] \\ &= \min_{\phi_k} \sum_{i=1}^{k} \mathbb{E}_{\mathcal{T}_i}[\mathcal{L}_{\mathcal{T}_i,Q_{k,i}}(U^m_{\mathcal{T}_k,S_k}(\phi_k))] \\ &\approx \min_{\phi_k} \sum_{i=1}^{k} \mathbb{E}_{\mathcal{T}_i}[\mathcal{L}_{\mathcal{T}_i}(\tilde{\phi}_k)]\end{aligned} \tag{6}$$

Through simply editing query information, the expected loss of meta objective (in Equation 6 ) is consistent with that of CL objective (in Equation 2). While, the same optimizee as MAML guarantees that the optimization process performs in a meta learning manner.

**Prototypes-guided Sample Selection Schemes** The choice of query samples should provide generalized information regarding all tasks that have been seen, $\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_k$. We apply prototypical network (Snell et al., 2017) to generate prototypes and use prototypes as selection criteria. Each prototype is the mean vector of feature representations of a specified problem(e.g., a class for a classification task). This selection scheme chooses a certain amount of examples with the shortest Euclidean distance to prototypes in a ranking-based manner, and deem these examples as representatives samples that can generalize all tasks that have been learned. Note that the prototypical network is constantly optimized and corrects prototypical information in the training process.

**Algorithm 1:** Meta Training

```
Input: Initial model parameters $\boldsymbol{\theta} = \boldsymbol{\phi}_{proto} \cup \boldsymbol{\phi}_{pred}$,
       support set $S$, support set buffer size $m$,
       memory buffer $\mathcal{M}$, inner-loop learning rate $\alpha$,
       outer-loop learning rate $\beta$.
Output: Trained model parameters $\boldsymbol{\theta}$
for $i = 1, 2, ...$ do
    [Inner Loop]
    $S_i \leftarrow m$ batches from the stream
    $\mathcal{L}_{proto} \leftarrow$ MemoryModule$(\boldsymbol{\phi}_{proto}, S_i, \mathcal{M}, n)$
    $\mathcal{L}_{inner} = \mathcal{L}_{proto} + \mathcal{L}_{CE}(\boldsymbol{\theta}, S_i)$
    $\tilde{\boldsymbol{\phi}}_{pred} = \text{SGD}(\mathcal{L}_{inner}, \boldsymbol{\phi}_{proto}, \boldsymbol{\phi}_{pred}, S_i, \alpha)$
    [Read Function]
    $Q_i \leftarrow$ Sample$(\mathcal{M}, all)$
    or $Q_i \leftarrow$ RandomSample$(\mathcal{M}, m)$
    [Outer Loop]
    $J(\boldsymbol{\theta}) = \mathcal{L}_{CE}(\boldsymbol{\phi}_{proto}, \tilde{\boldsymbol{\phi}}_{pred}, Q_i)$
    $\boldsymbol{\theta} \leftarrow$ Adam$(J(\boldsymbol{\theta}), \beta)$
    if all the training data is seen then
        Stop Iteration
    end
end
```

**Algorithm 2:** MemoryModule$(\boldsymbol{\phi}_{proto}, S_i, \mathcal{M}, n)$

```
Input: Initial model parameter $\boldsymbol{\phi}_{proto}$, support set
       $S_i$, memory buffer $\mathcal{M}$, number of selected
       samples per class $n$.
Output: Prototypical network loss $\mathcal{L}_P$, updated
        memory buffer $\mathcal{M}$
[Prototypical Network]
for class $l$ in $S_i$ do
    $S_l \leftarrow$ RandomSample$(S_{i,l}, N_S)$
    $Q_l \leftarrow$ RandomSample$(S_{i,l} \backslash S_l, N_Q)$
    $\mathbf{c}_l \leftarrow \frac{1}{|S_l|} \sum_{(x_i, y_i) \in S_l} h_{\boldsymbol{\phi}_{proto}}(x_i)$
    for $(x, y)$ in $Q_l$ do
        $\mathcal{L}_P \leftarrow \mathcal{L}_P + \frac{1}{N_Q}[d(h_{\boldsymbol{\phi}_{proto}}(\mathbf{x}), \mathbf{c}_l) +$
          $\log \sum_{l'} \exp(-d(h_{\boldsymbol{\phi}_{proto}}(\mathbf{x}), \mathbf{c}_l))]$
    end
    Write or update $\boldsymbol{c}_l$ in $\mathcal{M}$
end
[Samples Selection]
for class $l$ in $\mathcal{M}$ do
    $X_l \leftarrow$ KNN $(D_{\mathcal{M}} \cup S_i, \boldsymbol{c}_l, n)$
    Updates $X_l$ in $\mathcal{M}$
end
```

**Knowledge Transfer** A CL model is expected to acquire the ability of knowledge transfer between different tasks. While, transfer learning aims to ensure the learning process of a task can benefit from acquired knowledge from another domain. Thereby, transfer learning is substantial in CL. Andrychowicz et al. (2016) state that the problem of transfer learning can be cast as one of generalisation problems in meta learning. Given inner-loop updated parameters $\tilde{\phi}_k$ that are heavily biased towards the distribution of the current domain $p(\mathcal{T}_k)$, MAML-CL interprets outer loop optimization as a generalization problem on all tasks that have been learned. Such a meta objective depicts a CL scenario and solves the same expected loss. In such a way, MAML-CL enables knowledge transfer to occur not only within the current learning domain but also between all domains that have been learned.

**Fast Adaptation** The optimizee in MAML-CL is initial parameters $\phi_k$ with optimization direction $\phi_k - \tilde{\phi}_k$, implying the concern of optimization direction for the next update step. MAML-CL finds an optimal initial state with few update steps, such that yielding a minimal expected loss on all learned tasks. In other words, it learns optimal update directions on all learned tasks. By preserving optimisation information on all tasks, MAML-CL enables fast adaptation across all learned domains, thereby catastrophic forgetting mitigation.

**Algorithm 3:** Meta Inference

```
Input: Initial model parameters $\boldsymbol{\theta} = \boldsymbol{\phi}_{proto} \cup \boldsymbol{\phi}_{pred}$,
       support set buffer size $m$, memory $\mathcal{M}$, batch
       size $b$, inner-loop learning rate $\alpha$, test set $T$.
Output: Predictions on the test set
$S \leftarrow$ Sample$(\mathcal{M}, m \cdot b)$
$Q \leftarrow T$
$\tilde{\boldsymbol{\phi}}_{pred} = \text{SGD}(\mathcal{L}, \boldsymbol{\phi}_{proto}, \boldsymbol{\phi}_{pred}, S, \alpha)$
Predict$(Q, \boldsymbol{\phi}_{proto}, \tilde{\boldsymbol{\phi}}_{pred})$
```

## 5 Model

We incorporate the devised MAML framework with a prototypes-guided samples selection strategy (Ho et al., 2021) to address text classification. We employ First-order MAML (FOMAML) (Finn et al., 2017) so as to reduce computational complexity of MAML.

The proposed CL model $f_\theta$ consists of a representation learning network (RLN), $h_{\boldsymbol{\phi}_{proto}}$ with parameters $\boldsymbol{\phi}_{proto}$ , and a prediction network (PN) $g_{\boldsymbol{\phi}_{pred}}$ with parameters $\boldsymbol{\phi}_{pred}$. In particular, RLN trains a model $h_{\boldsymbol{\phi}_{proto}}(\cdot) : \mathcal{X} \to \boldsymbol{c} \in \mathbb{R}^{D \times N}$ where $\boldsymbol{c}$ denotes a prototype with a $D$-dimensional representation and $N$ is the number of classes. While, $g_{\boldsymbol{\phi}_{pred}}(\cdot)$ learns a mapping : $\boldsymbol{c} \to \mathcal{Y} \in \mathbb{R}^N$. We add a single-hidden-layer feed-forward neural network on top of a encoder to formulate a prototypical network and use a single linear layer as the prediction learning network.

| Method | Order 1 | Order 2 | Order 3 | Order 4 | Order 5 | Order 6 |
|---|---|---|---|---|---|---|
| MAML | $34.58 \pm 6.98$ | $38.80 \pm 11.19$ | $32.36 \pm 16.81$ | $29.32 \pm 14.23$ | $22.55 \pm 12.93$ | $27.96 \pm 3.88$ |
| Replay | $42.52 \pm 3.87$ | $29.81 \pm 0.19$ | $29.64 \pm 0.43$ | $46.50 \pm 5.31$ | $40.43 \pm 2.29$ | $38.71 \pm 2.26$ |
| AGEM | $38.36 \pm 3.12$ | $29.84 \pm 0.14$ | $29.86 \pm 0.18$ | $40.35 \pm 1.15$ | $37.34 \pm 1.39$ | $37.82 \pm 1.16$ |
| OML-ER | $47.80 \pm 3.41$ | $25.43 \pm 2.71$ | $31.41 \pm 4.19$ | $40.76 \pm 9.27$ | $\mathbf{48.52 \pm 5.58}$ | $39.20 \pm 12.16$ |
| OML-ER$_{\text{limit}}$ | $36.78 \pm 8.61$ | $31.42 \pm 4.78$ | $37.19 \pm 11.79$ | $34.54 \pm 4.38$ | $39.19 \pm 4.72$ | $42.25 \pm 10.43$ |
| PMR | $26.89 \pm 3.51$ | $26.44 \pm 1.68$ | $21.57 \pm 3.58$ | $26.33 \pm 1.44$ | $19.84 \pm 5.53$ | $25.20 \pm 3.61$ |
| MAML-CL$_{\text{all}}$ | $\mathbf{56.25 \pm 2.03}$ | $\mathbf{53.59 \pm 2.79}$ | $\mathbf{49.41 \pm 4.59}$ | $\mathbf{54.23 \pm 3.15}$ | $47.93 \pm 3.70$ | $\mathbf{45.56 \pm 3.71}$ |
| MAML-CL$_{\text{random}}$ | $52.09 \pm 0.23$ | $47.06 \pm 5.74$ | $41.44 \pm 2.27$ | $49.04 \pm 8.55$ | $42.98 \pm 9.20$ | $39.09 \pm 1.85$ |

Table 1: Performance Using Different Training Set Permutation in Terms of Accuracy.

| Memory Size | Method | Overall Accuracy |
|---|---|---|
| **All** Seen Data | MAML | $30.93 \pm 5.64$ |
| | Replay | $37.94 \pm 6.87$ |
| | AGEM | $35.60 \pm 4.57$ |
| | OML-ER | $38.85 \pm 9.09$ |
| **45** Samples | OML-ER$_{\text{limit}}$ | $36.90 \pm 3.73$ |
| | PMR | $24.38 \pm 2.95$ |
| | MAML-CL$_{\text{all}}$ | $\mathbf{51.16 \pm 4.15}$ |
| | MAML-CL$_{\text{random}}$ | $45.28 \pm 4.94$ |

Table 2: Overall Performance in Terms of Accuracy.

### 5.1 Meta Training and Inference

The inner optimization performs task-specific funetuning only on parameters $\phi_{pred}$ in PN, where inner loop loss $\mathcal{L}_{inner}$ contains cross entropy loss $\mathcal{L}_{CE}$ and prototypical network loss $\mathcal{L}_{proto}$. The inner loop learning process defines decision making boundaries for current learning task $\mathcal{T}_k$. The outer loop regularizes RLN and PN over all learned tasks, $\mathcal{T}_1, ..., \mathcal{T}_{k-1}, \mathcal{T}_k$, by meta learning model parameters $\theta$ using stored samples from $D_{\mathcal{M}}$ as query samples. Note that $\boldsymbol{\theta} = \boldsymbol{\phi}_{proto} \cup \boldsymbol{\phi}_{pred}$. The meta training and inference process are shown in Algorithm 1 and Algorithm 3 respectively.

### 5.2 Query Samples Selection

To optimize memory footprint and achieve sample efficiency, we expect the stored data set $D_{\mathcal{M}}$, which serves as the query set $Q_k$, consists of representative samples set for all tasks that have been learned. Akin to Prototypes-Guided Memory Replay Network (Ho et al., 2021), each prototype selects representative instances for a corresponding class by similarity from current support set $S_i$, where $i$ denotes episode index. We use Euclidean distance $d(\cdot)$ to measure similarity between samples and prototypes. Following the memory constraint in the CL setup, we limit the number of stored samples for each class (i.e., 5 in this paper). Note that prototypes are dynamically updated in each learning iteration. We propose two read functions: (1) read *all* from memory; (2) read *randomly* from memory.

## 6 Experiments

### 6.1 Datasets

Following prior work on class-incremental learning, we leverage the benchmark datasets introduced by de Masson d'Autume et al. (2019), where each dataset contains 115,000 training samples and 7,600 test samples. Each dataset is seen as a separated learning task. We use three datasets from two different domains, i.e., AGNews (news classification; 4 classes), Yelp (sentiment analysis; 5 classes) and Amazon (sentiment analysis; 5 classes). Hereby, we can observe CL models performance between tasks from the same or different domains.

### 6.2 Setup

Considering the real-world scenario, we use a low resource CL setup where we reduce the size of the training set to its **10%** , i.e., 11, 500 per task and 34, 500 in total. We further limit our memory budget to a constant size $B = \mathbf{45}$, i.e., selecting 5 samples per class.

The encoder for all models is a pretrained ALBERT-Base-v2 (Lan et al., 2020) from Hugging Face Transformers, where the input sequence length is pruned to 200. The setup for models using MAML framework is as follows. The inner loop optimizer is SGD with learning rate, $\alpha = 3e^{-3}$. The outer loop optimizer is Adam with learning rate, $\beta = 3e^{-5}$. The baselines without a prototypical network utilise a random sampler with batch size, $b = 25$. The models with a prototypical network use a sampler that randomly selects 5 training

| **Method** | **Yelp** | **AGNews** | **Amazon** | **Average Accuracy** |
|---|---|---|---|---|
| OML-ER$_{\text{limit}}$ | 39.26 (-12.27) | 13.04 (-20.44) | 40.74 (-11.43) | 31.02 (-14.71) |
| PMR | 41.61 (-13.02) | 0.19 (-66.62) | 38.87 (-18.08) | 26.89 (-32.57) |
| MAML-CL$_{\text{all}}$ | **44.73** (**-8.99**) | **79.54** (**+5.28**) | **44.48** (**-10.47**) | **56.25** (**-4.73**) |
| MAML-CL$_{\text{random}}$ | 41.29 (-9.92) | 74.32 (-0.77) | 40.65 (-10.46) | 52.09 (-7.05) |
| **Method** | **Amazon** | **Yelp** | **AGNews** | **Average Accuracy** |
| OML-ER$_{\text{limit}}$ | 10.41 (-9.63) | 11.81 (-6.31) | **84.17** (**-2.72**) | 35.46 (-6.22) |
| PMR | 0.0 (-9.75) | 0.0 (-21.57) | 64.72 (-23.07) | 21.57 (-18.13) |
| MAML-CL$_{\text{all}}$ | **38.61** (**+16.37**) | **44.22** (**+11.68**) | 65.39 (-19.88) | **49.41** (**+2.73**) |
| MAML-CL$_{\text{random}}$ | 34.37 (+6.60) | 38.93 (-4.69) | 51.01 (-32.08) | 41.44 (-10.05) |
| **Method** | **AGNews** | **Yelp** | **Amazon** | **Average Accuracy** |
| OML-ER$_{\text{limit}}$ | 47.04 (-31.03) | 37.29 (-12.98) | 36.14 (-18.03) | 40.16 (-20.67) |
| PMR | 0.0 (-65.11) | 28.15 (-27.21) | 31.38 (-26.08) | 19.84 (-39.47) |
| MAML-CL$_{\text{all}}$ | 67.99 (**+13.71**) | **38.18** (**-11.78**) | **37.62** (**-12.42**) | **47.93** (**-3.49**) |
| MAML-CL$_{\text{random}}$ | **74.11** (+9.70) | 26.85 (-15.49) | 27.96 (-14.5) | 42.98 (-6.75) |

Table 3: Per Task and Overall Performance Using Training Set Order 1, Order 3 and Order 5. Note that the values in brackets represent the accuracy difference, where "+" indicates an increase in accuracy after downsizing training set and vice versa.

samples from each class for each epoch without replacement, where $b = 20$ for AGNews and $b = 25$ for Yelp and Amazon.

### 6.3 Baselines

We use the following CL models as baselines:

- **MAML** (Finn et al., 2017) refers to FOMAML model without extra means of forgetting mitigation in our evaluations.

- **Replay** performs one gradient update on randomly selected samples from memory. Replay model utilises the sparse experience replay strategy with 1% replay rate.

- **A-GEM** (Chaudhry et al., 2019) imposes one gradient constraint to restrict current task gradient projection regions. A-GEM randomly reads samples and decides the direction of optimization constraints.

- **OML-ER** (Holla et al., 2020) is a recent CL model, which uses FOMAML framework with episodic experience replay. OML-ER writes all seen samples into memory and randomly chooses samples for episodic replay. Note that OML-ER$_{\text{limit}}$ refers to OML-ER with limited memory budgets.

- **PMR** (Ho et al., 2021) employs FOMAML framework with the prototypes-guided samples selection scheme for episodic experience replay. It outperforms OML-ER given limited memory budgets.

### 6.4 Results

We evaluate model performance in terms of test set accuracy. Specifically, we test model performance for each task after completing the learning of the last task. Note that the test set has the same permutation of tasks as the training set. Each result of the different methods is the average accuracy of the 3 best results in 5 runs. Table 1 presents the evaluation results of all baselines and the proposed models in all 6 different training set orders. The permutations are detailed in Appendix A.2. In particular, each result indicates average accuracy and standard deviations of the 3 best runs. Table 2 shows the overall performance and standard deviations across all training set orders.

As shown in Table 1 and Table 2, the proposed model, MAML-CL$_{\text{all}}$ yields the highest average accuracy in almost all orders. Its overall performance surpasses the strong baseline, namely OML-ER, by more than 12%, using only 45 samples occupied in memory. Its standard deviations are all less than 5%, which indicates strong stability of performance in both random seeds and all training set permutations. The other proposed method, MAML-CL$_{\text{random}}$ also exhibits good results across all various permutations of tasks, with the second-highest average accuracy. But its standard deviations vary from 0.23% to 9.20 %, implying rel-

| Method | Memory Size | Yelp | AGNews | Amazon | Average Accuracy |
|---|---|---|---|---|---|
| OML-ER | 27 | **49.81** | 19.38 | **49.06** | 39.4 |
| MAML-CL$_{\text{all}}$ | | 44.04 | **67.58** | 43.75 | **51.79** |
| MAML-CL$_{\text{random}}$ | | 36.16 | 63.57 | 35.10 | 44.94 |
| OML-ER | 45 | **47.96** | 13.05 | **49.31** | 36.78 |
| MAML-CL$_{\text{all}}$ | | 44.73 | **79.54** | 44.48 | **56.25** |
| MAML-CL$_{\text{random}}$ | | 41.29 | 74.32 | 40.65 | 52.09 |
| OML-ER | 63 | 52.67 | 24.63 | 50.25 | 42.51 |
| MAML-CL$_{\text{all}}$ | | **55.05** | **86.21** | **52.21** | **64.59** |
| MAML-CL$_{\text{random}}$ | | 40.89 | 69.33 | 39.25 | 49.83 |
| OML-ER | All Seen Data | 46.83 | 49.24 | 47.32 | 47.80 |
| MAML-CL$_{\text{all}}$ [1] | | – | – | – | – |
| MAML-CL$_{\text{random}}$ | | **55.05** | **86.21** | **52.21** | **64.59** |

Table 4: Per Task and Overall Performance Using Various Memory Limitations

atively weak stability. Remarkably, the two proposed models even outperform baselines that have unlimited memory budgets. In addition, we surprisingly noted that PMR underperforms the non-CL model, MAML. It implies insufficient training iterations leads to an immature prototypical network, which severely impacts the performance of PMR. Such a deficiency do not cause the inadequate performance of our models. Arguably, MAML-CL lessens the burden of the prototypical network, by editing query information instead of episodic memory replay. It manifests the benefit of MAML-CL framework in terms of forgetting mitigation.

Note that per-task performance across all models is detailed in Appendix A.3. It shows the outstanding performance of MAML-CL models on each task.

### 6.5 Analysis

**Fast Adaptation** Table 3 shows per task and overall performance in Order 1, Order 3 and Order 5 respectively [2]. To evaluate in terms of fast adaptation, we compare model performance of ingesting full and downsized training sets. The accuracy differences are given in brackets. PMR shows its vulnerability of model performance with insufficient training instances. The accuracy is even down to 0 for some preceding tasks. OML-ER$_{\text{limit}}$ also exhibits a large decline in performance. Even though its performance in the latest tasks are competitive, OML-ER$_{\text{limit}}$ still underperforms in all earlier tasks compared to MAML-CL$_{\text{all}}$, suggesting its insufficient ability to ease forgetting. MAML-CL$_{\text{all}}$ yields the best performance on almost all tasks by a relatively small degradation. Intriguingly, MAML-CL$_{\text{all}}$ can even improve the accuracy of some preceding tasks given a smaller set of training data. Especially, MAML-CL$_{\text{all}}$ rises accuracy of preceding tasks to more than 10% in Order 3 and Order 5, indicating its impressive ability of knowledge retention. Similarly, MAML-CL$_{\text{random}}$ also poses a small degradation of performance or improvements in previously seen tasks. It testifies the superior of the proposed framework, MAML-CL in terms of fast adaptation and forgetting mitigation. The reason behind might be immature prototypical network solves the over-fitting problem (i.e., overfitting towards training samples) , thereby more generalised samples are selected as query samples.

**Memory Efficiency** We choose OML-ER for comparison, given its similar FOMAML framework with MAML-CL and competitive performance. We present the experimental results of using the training set permutation in Order 1, shown in Table 4. With the same random read function and memory size, MAML-CL models are superior to OML-ER, especially in the average accuracy and accuracy of the second task. Notably, the second task, AGNews belongs to news classification task, which is a different domain compared to sentiment analysis for Yelp (i.e., the first task) and Amazon (i.e., the latest task). We argue that the MAML-CL can strike an optimized balance of its performance between two different domains, rather than skewing towards the latest one. Considering its outstanding ability of knowledge retention despite various memory size, we argue that MAML-CL models achieve memory efficiency.

[2] For training set permutations, we found similar learning behaviours using Order1 and Order 4, Order 2 and Order 3, and Order 5 and Order 6.

| Method | Yelp | AGNews | Amazon | Average Accuracy |
|---|---|---|---|---|
| PMR | 41.61 | 0.19 | 38.87 | 26.89 |
| MAML-CL$_{\mathrm{all}}$ | **44.73** | **79.54** | **44.48** | **56.25** |
| OML-ER | 46.83 | 49.24 | 47.32 | 47.80 |
| MAML-CL$_{\mathrm{random}}$ | **55.05** | **86.21** | **52.21** | **64.59** |

Table 5: Per Task and Overall Performance Using Various Forgetting Mitigation Strategies

| Sample Selection Method | | Yelp | AGNews | Amazon | Average Accuracy |
|---|---|---|---|---|---|
| **Random** | | 20.98 | 0.0 | 21.72 | 14.23 |
| **Prototypes-Guided** | Diversity$^\dagger$ | **41.29** | **74.32** | **40.65** | **52.09** |
| | Uncertainty | 28.8 | 63.43 | 28.77 | 40.33 |

Table 6: Per Task and Overall Performance Using Various Sample Selection Methods. $^\dagger$ The proposed method.

**Effect of Query Information Editing** MAML-CL$_{\mathrm{all}}$ has the same prototypes-guided sample selection scheme and the same read mechanisms (i.e., read all samples from memory) as PMR. We compare these two models to analyze the effect of query information editing and memory replay. Table 5 illustrates that MAML-CL$_{\mathrm{all}}$ surpasses PMR in all tasks. In particular, MAML-CL$_{\mathrm{all}}$ outperforms PMR in terms of average accuracy by nearly 30%. As for the ability of knowledge retention from various domains, the performance of MAML-CL$_{\mathrm{all}}$ on AGNews exceeds that of PMR by more than 75%. Additionally, we replace the prototypes-guided selection strategy with OML-ER's selection criteria in MAML-CL$_{\mathrm{random}}$ to maintain the consistency of read and write mechanisms between these two methods. Table 5 displays that MAML-CL$_{\mathrm{random}}$ still manage to outperform OML-ER in all tasks and overall performance, showing a strong sequential learning ability for various tasks. It is obvious that MAML-CL successfully beats the most widely-used episodic memory replay method in CL. The proposed MAML-CL framework exhibits its superiority of alleviating catastrophic forgetting by simply editing query information given low training resources.

**Effect of Prototypes-Guided Sample Selection** Under the same MAML-CL framework with the same random read mechanism and the same memory size limitation (i.e., $B = 45$), we consider two main samples selection strategies [3], i.e., random selection and prototypes-guided selection. As for the prototypes-guided sample selection, we further deliberate two popular paradigms in active learning, namely the diversity-based method and the uncertainty-based method (Wang et al., 2020). We consider selecting representative samples of all classes as a diversity-based method. Opting for samples that are far away from prototypes is an uncertainty-based method. Table 6 displays that prototypes-guided selection methods clearly outperform random selection, especially diversity-based criteria. Random selection is considered as a simple but efficient sample selection strategy in memory replay (de Masson d'Autume et al., 2019; Wang et al., 2020). But, we find that random selection is incompetent for query information editing of MAML-CL, when training resources are limited. Arguably, not enough training iterations for the random selection strategy leads to inadequate generalization information, thereby prone to forgetting. Furthermore, the uncertainty-based method is inferior to the diversity-based method. It proves the verity that using representative samples as query information is competent to solve CL problems.

## 7 Conclusion

We introduce a meta learning framework to address CL problems, namely MAML-CL. It is designed to enhance MAML framework in CL, by editing query information. In particular, MAML-CL edits query information coupling with the prototypes-guided sample selection scheme to achieve generalization. Given limited training resources, MAML-CL shows its robustness in terms of stability, fast adaptation, forgetting mitigation and memory efficiency. A future research direction can be exploring and redesigning other meta learning frameworks that are conducive to CL.

[3] Note that sample selection strategy refers to the selection criteria of storing examples into memory in this paper.

## References

Durmus Alp Emre Acar, Ruizhao Zhu, and Venkatesh Saligrama. 2021. Memory efficient online meta learning. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 32–42. PMLR.

Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. 2018. Memory aware synapses: Learning what (not) to forget. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part III*, volume 11207 of *Lecture Notes in Computer Science*, pages 144–161. Springer.

Marcin Andrychowicz, Misha Denil, Sergio Gomez Colmenarejo, Matthew W. Hoffman, David Pfau, Tom Schaul, and Nando de Freitas. 2016. Learning to learn by gradient descent by gradient descent. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 3981–3989.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. 2019. Efficient lifelong learning with A-GEM. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Cyprien de Masson d'Autume, Sebastian Ruder, Lingpeng Kong, and Dani Yogatama. 2019. Episodic memory in lifelong language learning. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 13122–13131.

Chelsea Finn, P. Abbeel, and S. Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*.

Stella Ho, Ming Liu, Lan Du, Longxiang Gao, and Yong Xiang. 2021. Prototypes-guided memory replay for continual learning. *ArXiv*, abs/2108.12641.

Nithin Holla, Pushkar Mishra, Helen Yannakoudakis, and Ekaterina Shutova. 2020. Meta-learning with sparse experience replay for lifelong language learning. *CoRR*, abs/2009.04891.

Yufan Huang, Yanzhe Zhang, Jiaao Chen, Xuezhi Wang, and Diyi Yang. 2021. Continual learning for text classification with information disentanglement based regularization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 2736–2746. Association for Computational Linguistics.

K. J. Joseph and Vineeth Nallure Balasubramanian. 2020. Meta-consolidation for continual learning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

M. McCloskey and N. J. Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation*, 24:109–165.

Alex Nichol, Joshua Achiam, and John Schulman. 2018. On first-order meta-learning algorithms. *CoRR*, abs/1803.02999.

Abiola Obamuyide and Andreas Vlachos. 2019. Meta-learning improves lifelong relation extraction. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 224–229, Florence, Italy. Association for Computational Linguistics.

Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, , and Gerald Tesauro. 2019. Learning to learn without forgetting by maximizing transfer and minimizing interference. In *International Conference on Learning Representations*.

Jake Snell, Kevin Swersky, and Richard S. Zemel. 2017. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 4077–4087.

Zirui Wang, Sanket Vaibhav Mehta, Barnabas Poczos, and Jaime Carbonell. 2020. Efficient meta lifelong-learning with limited memory. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 535–548, Online. Association for Computational Linguistics.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 649–657.

# A Appendix

## A.1 Comparison of Continual Learning, MAML and MAML-CL

As shown in Table 7, we illustrate the differences and similarities of CL, MAML and MAML-CL. In such a way, we explain how MAML-CL augments knowledge transfer and fast adaptation across various domains so as to prevent catastrophic forgetting, detailed in Section 4.

## A.2 Training Set Orders for Evaluations

For model stability evaluation, we use three datasets, i.e., Yelp, AGNews and Amazon and form a total of 6 different training set orders as follows:

1. Yelp → AGNews → Amazon
2. Yelp → Amazon → AGNews
3. Amazon → Yelp → AGNews
4. Amazon → AGNews → Yelp
5. AGNews → Yelp → Amazon
6. AGNews → Amazon → Yelp

## A.3 Extra Evaluations Using Downsized Training set

We conduct extra evaluations on the proposed models using the downsize training set.

**Per Task Performance** Figure 1 demonstrates that the proposed methods outstanding ability to retain knowledge across various domains. In particular, both MAML-CL$_{\text{all}}$ and MAML-CL$_{\text{random}}$ only store 45 samples in memory and significantly outperform the strong baseline, OML-ER, where OML-ER writes all training data into memory. It suggests that MAML-CL models achieve samples efficiency. Clearly, MAML-CL$_{\text{all}}$ and MAML-CL$_{\text{random}}$ exhibit impressive performance in terms of stability, forgetting mitigation and memory efficiency across various domains.

**Memory Insight** To further analyze the effect of Prototypes-Guided Sample Selection, we visualize unigram distribution change inside memory in three different learning iterations, i.e., Episode 50, Episode 150 and Episode 250. In Figure 2, the y-axis presents the counts of each unigram. While the x-axis presents the unigram index. It shows that the saved samples provide a good diversity of information. Consequently, we testify that the prototypical-guided selection strategy enables samples efficiency, thereby optimizing memory footprints. Note that we perform this evaluation using training set permutation follows Order 1.

## A.4 Evaluations Using Full Training set

We further conduct evaluations on the proposed models given the full training set.

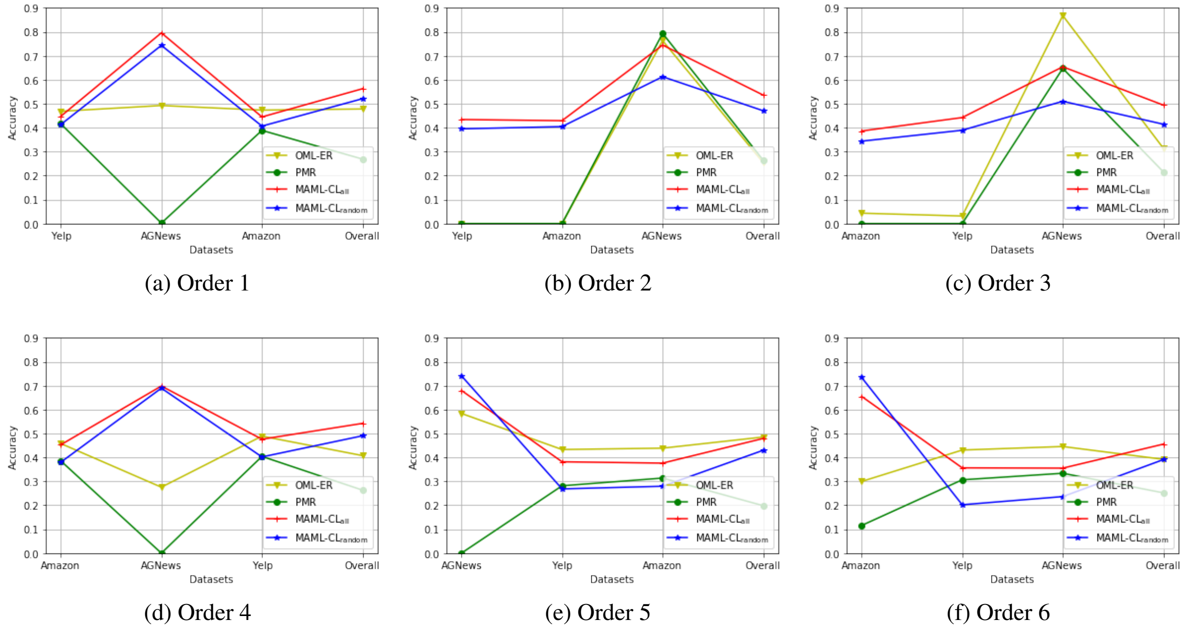
**Overall Performance** Table 8 shows performance using different training set permutations given the full training set. The proposed model, MAML-CL$_{\text{all}}$ yields the highest average accuracy in Order 1, Order 2, Order 4 and overall performance. Its standard deviation is relatively small, compared to two strong baselines, OML-ER$_{\text{limit}}$ and PMR. While, the other proposed method, MAML-CL$_{\text{random}}$ exhibits strong stability across all various permutations of tasks, with the second-highest average accuracy and the smallest standard deviations among all methods. Notably, the two proposed methods surpass OML-ER$_{\text{limit}}$ and PMR, which also use FOMAML framework, by approximately $1 \sim 4\%$ in accuracy and $3 \sim 7\%$ in standard deviations.

**Per Task Performance** As shown in Figure 3, the proposed methods demonstrate a stable performance on each task, in comparison of OML-ER$_{\text{limit}}$ and PMR. Especially, MAML-CL$_{\text{all}}$ obtains a higher than 50% accuracy for each task in Order 1, Order 4, Order 5 and Order 6. It manifests that the proposed method provides stability for solving catastrophic forgetting in CL.

**Impact of Memory Size Limitations** We evaluate the proposed models using various memory size limitations as shown in Table 9. We spot a phenomenon that the variation of performance is not obvious between MAML-CL$_{\text{all}}$ and MAML-CL$_{\text{random}}$ regardless of memory limitation size. When the memory constraint reaches 45 samples and above (i.e., $B >= 45$), performance are not improved or improved by a small margin. Note that MAML-CL$_{\text{all}}$ has a restriction on the size of the saved sample set, given the size of the query set in each iteration should not be large. Hence, we only conduct this evaluation with the memory size of 27, 45, and 63 samples.

| Paradigm | Continual Learning | MAML | MAML-CL |
|---|---|---|---|
| Initial Parameters | $\theta_k$ | $\phi_k$ | $\phi_k$ |
| Finetuned Parameters | $\tilde{\theta}_k$ | $\tilde{\phi}_k$ | $\tilde{\phi}_k$ |
| Expected Loss | $\sum_{i=1}^{k} \mathbb{E}_{\mathcal{T}_i}[\mathcal{L}_{\mathcal{T}_i}(\tilde{\theta}_k)]$ | $\mathbb{E}_{\mathcal{T}}[\mathcal{L}_{\mathcal{T}}(\tilde{\phi}_k)]$ | $\sum_{i=1}^{k} \mathbb{E}_{\mathcal{T}_i}[\mathcal{L}_{\mathcal{T}_i}(\tilde{\phi}_k)]$ |
| Optimizee | $\tilde{\theta}_k$ | $\phi_k$ | $\phi_k$ |
| Optimization Direction | $\tilde{\theta}_k - \theta_k$ | $\phi_k - \tilde{\phi}_k$ | $\phi_k - \tilde{\phi}_k$ |
| Transferred Parameters | $\tilde{\theta}_k$ | $\phi_k$ | $\phi_k$ |

Table 7: Comparison of Continual Learning, MAML and MAML-CL.



(a) Order 1

(b) Order 2

(c) Order 3

(d) Order 4

(e) Order 5

(f) Order 6

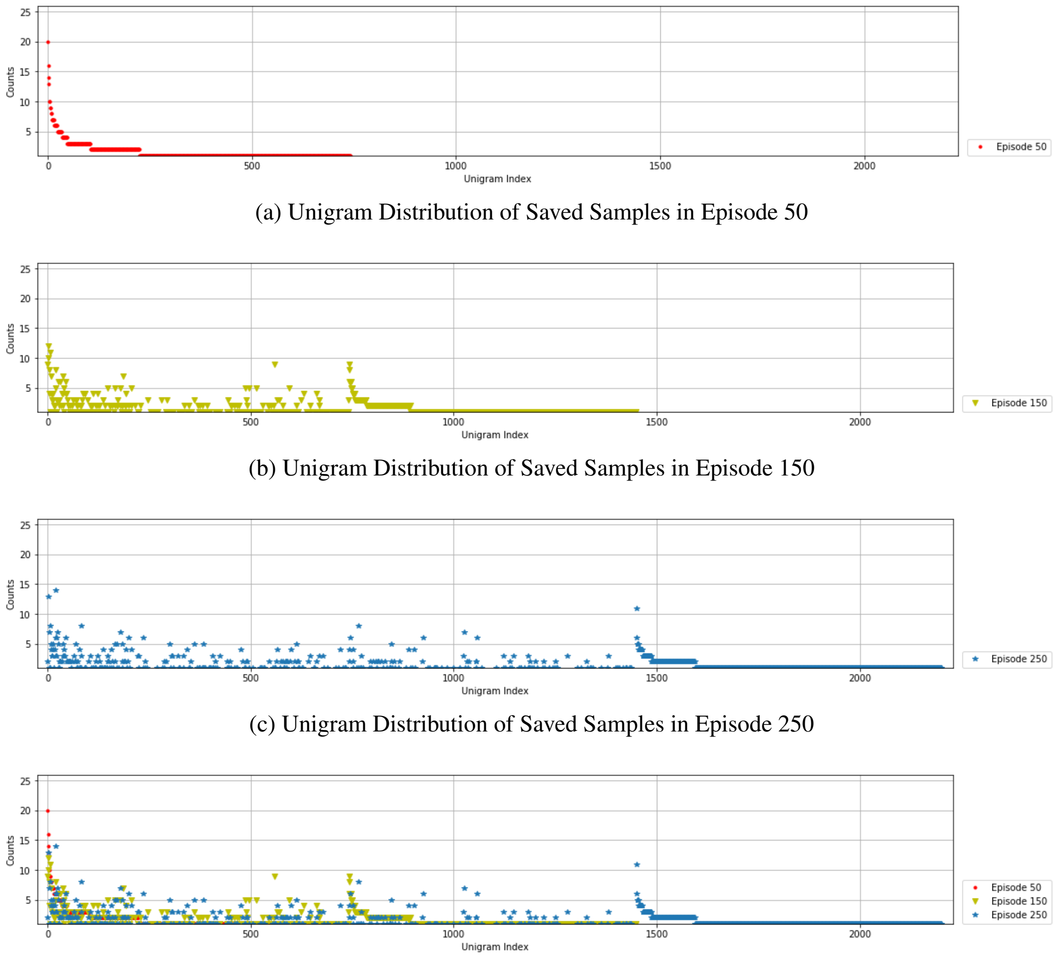
Figure 1: Per Task and Overall Performance Using Various Training Set Permutations (Downsized Training set).

| Method | Order 1 | Order 2 | Order 3 | Order 4 | Order 5 | Order 6 | Overall |
|---|---|---|---|---|---|---|---|
| AGEM† | 37.62 | 30.20 | 30.55 | 41.94 | 39.59 | 39.75 | 36.61 ± 5.02 |
| Replay† | 43.76 | 30.07 | 30.45 | 41.91 | 42.12 | 44.42 | 38.79 ± 6.68 |
| OML-ER$^{\dagger}_{\text{limit}}$ | 45.73 | 46.44 | 41.68 | 47.49 | **60.83** | 62.11 | 50.71 ± 8.57 |
| PMR† | 59.46 | 35.38 | 39.70 | 56.56 | 59.31 | **62.19** | 52.09 ± 11.50 |
| MAML-CL$_{\text{all}}$ | **60.98** | **54.88** | 46.68 | **61.43** | 51.42 | 53.30 | **54.78 ± 5.69** |
| MAML-CL$_{\text{random}}$ | 59.14 | 53.70 | **51.49** | 58.29 | 49.73 | 46.81 | 53.19 ± 4.84 |

Table 8: Performance Using Different Training Set Permutation in Terms of Accuracy. Note that the same memory limitation apply to all methods shown above. † Results obtained from (Ho et al., 2021).

| Method | Memory Size | Yelp | AGNews | Amazon | Average Accuracy |
|---|---|---|---|---|---|
| MAML-CL$_{\text{all}}$ | 27 | **46.90** | 61.46 | **48.37** | 52.25 |
| MAML-CL$_{\text{random}}$ | | 44.35 | **72.62** | 43.68 | **53.55** |
| MAML-CL$_{\text{all}}$ | 45 | **53.72** | 74.26 | **54.95** | **60.98** |
| MAML-CL$_{\text{random}}$ | | 51.21 | **75.09** | 51.11 | 59.14 |
| MAML-CL$_{\text{all}}$ | 63 | **54.38** | 71.41 | **54.46** | **60.08** |
| MAML-CL$_{\text{random}}$ | | 51.67 | **75.38** | 51.54 | 59.53 |

Table 9: Per Task and Overall Performance Using Various Memory Limitations



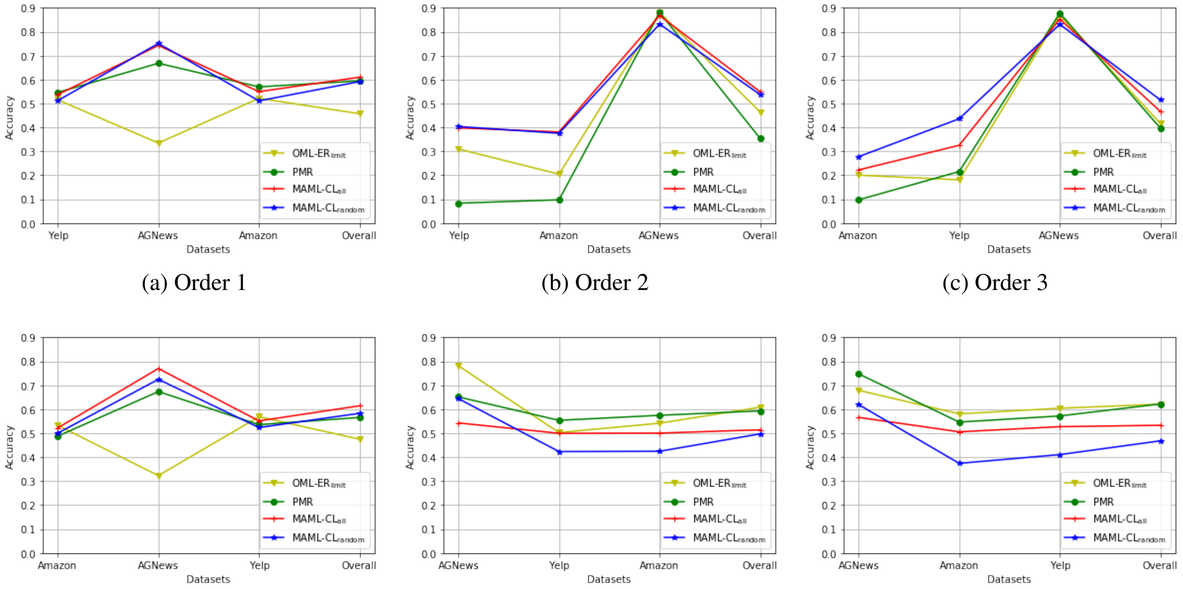
(a) Unigram Distribution of Saved Samples in Episode 50

(b) Unigram Distribution of Saved Samples in Episode 150

(c) Unigram Distribution of Saved Samples in Episode 250

(d) Comparison of Unigram Distributions in Episode 50, Episode 150 and Episode 250

Figure 2: Visualization of Unigram Distribution Shift in Memory. Note that y-axis is in the range $[\mathbf{1}, \mathbf{26})$. The data points on the x-axis indicate the count of the corresponding unigram is 1.



(a) Order 1

(b) Order 2

(c) Order 3

(d) Order 4

(e) Order 5

(f) Order 6

Figure 3: Per Task and Overall Performance Using Various Training Set Permutations (Full Training set).