F-MALLOC: Feed-forward Memory Allocation for Continual Learning in Neural Machine Translation

Anonymous ACL submission

Abstract

In the evolving landscape of Neural Machine Translation (NMT), the pretrain-then-finetune paradigm has yielded impressive results. However, the persistent challenge of Catastrophic Forgetting (CF) remains a hurdle. While previous work has introduced Continual Learning (CL) methods to address CF, these approaches grapple with the delicate balance between avoiding forgetting and maintaining system extensibility. To address this, we propose a CL method, named **F-MALLOC** (**F**eed-forward **M**emory **ALLOC**ation). F-MALLOC is inspired by recent insights highlighting that feed-forward layers emulate neural memories and encapsulate crucial translation knowledge. It decomposes feed-forward layers into discrete memory cells and allocates these memories to different tasks. By learning to allocate and safeguard these memories, our method effectively alleviates CF while ensuring robust extendability. Besides, we propose a comprehensive assessment protocol for multi-stage CL of NMT systems. Experiments conducted following this new protocol showcase the superior performance of F-MALLOC, evidenced by higher BLEU scores and almost zero forgetting.[1]

1 Introduction

In the pursuit of achieving state-of-the-art results in Neural Machine Translation (NMT), the reliance on large-scale parallel corpora has been pivotal (Bahdanau et al., 2015; Vaswani et al., 2017). However, practical application scenarios often present challenges, especially when translation is necessitated for specific domains with limited data resources (Chu and Wang, 2018; Saunders, 2022). Typically, the prevalent paradigm involves the initial pretraining of models on expansive general domain corpus, followed by finetuning for the target domain (Freitag and Al-Onaizan, 2016; Chu and Dabre, 2019).

Despite the efficacy of this pretrain-then-finetune paradigm, it has been demonstrated that fine-tuning on the target domain can result in significant performance degradation in the general domain, a phenomenon known as Catastrophic Forgetting (CF) (French, 1993). In response to this challenge, various Continual Learning (CL) approaches have emerged to address CF in NMT systems. Existing efforts primarily rely on regularization-based techniques to constrain the divergence of model parameters from their previous values (Khayrallah et al., 2018; Saunders et al., 2019; Cao et al., 2021). While these methods are mathematically elegant, they still face challenges related to forgetting. Alternatively, some approaches take an architecture-based framework, isolating parameters specific to different tasks to prevent forgetting (Gu et al., 2021; Liang et al., 2021; Huang et al., 2023). However, they require prior information on task numbers to allocate parameters and rely on external storage of model or mask matrices, limiting its extendibility and applicability.

In summary, the demand for a CL method for NMT systems that is both extendable and effective in preventing forgetting is pressing. To this end, we introduce a new CL method termed **F-MALLOC** (**F**eed-forward **M**emory **ALLOC**ation), which is inspired by recent insights that feed-forward layers emulate neural memories and encapsulate crucial translation knowledge (Geva et al., 2021; Huang et al., 2023). Therefore, we facilitate new knowledge learning and mitigate CF by allocating and protecting these memories. F-MALLOC first leverages a structural pruning method to trim the feed-forward layers of a pretrained NMT model, preserving memories that encapsulate crucial general domain knowledge. Subsequently, F-MALLOC proceeds to learn a set of non-exclusive task masks, automatically allocating the 'writable' memory ca-

[1]The code and data for this work will be made publicly available after the completion of the review process.

pacity to upcoming tasks. The memories allocated in this manner are then designated as 'read-only.' F-MALLOC strategically blocks gradient flows through these 'read-only' memories, effectively mitigating the risk of forgetting.

Meanwhile, conventional CL evaluation protocols in the NMT area typically focus on a single stage of training, lacking a holistic perspective over multiple stages. Therefore, we introduce a comprehensive evaluation protocol for multi-stage CL in the NMT scenario. Our protocol incorporates metrics assessing forgetting mitigation and adaptation to novel tasks. To enhance robustness, we conduct tests with random task sequences, reducing biases from specific orders. This protocol provides a nuanced understanding of F-MALLOC and competing methods' performance over time in NMT.

Experiments conducted following the proposed protocol highlight the superior performance of F-MALLOC with high robustness. Additional analysis of F-MALLOC's memory allocation strategy reveals its effective utilization of task information, such as inherent difficulty or inter-task similarities, resulting in enhanced performance.

In summary, the contributions of this paper are as follows:

- We propose F-MALLOC, a multi-stage CL method that prevents forgetting and promotes new knowledge acquisition through feed-forward memory allocation. It requires no prior task information and minimal storage overhead.
- Through a tailored evaluation protocol for multi-stage CL in NMT systems, we enhance the understanding of system performance on both stability and plasticity.
- Further analysis of F-MALLOC's adaptive memory allocation strategy demonstrates its effectiveness in leveraging task difficulty and inter-task similarities to optimize capacity usage and encourage knowledge transfer.

2 Background

2.1 Feed-forward Layers Emulate Memory Networks

Feed-forward Layer. The prevalent architecture in NMT is the encoder-decoder Transformers(Vaswani et al., 2017), which is made of intertwined multi-head attention (MHA) and point-wise feed-forward layers. Our specific focus lies in the feed-forward layer, formally defined as:

$$\mathrm{FF}(x) = W^{(2)} \cdot \sigma(W^{(1)} \cdot x) \quad (1)$$

where $W^{(1)}, W^{(2)}$ represent learnable parameters (bias term omitted for simplification), and σ typically denotes the activation function, commonly ReLU.

Feed-forward layer as neural memory of knowledge. Recent research has explored the interpretability of feed-forward structures, noting a significant resemblance between the feed-forward layer and neural memory (Sukhbaatar et al., 2015). Treating parameter matrices $W^{(1)}$ and $W^{(2)}$ as keys and values respectively, the feed-forward layer can be seen as an unnormalized key-value memory (Sukhbaatar et al., 2019). Studies have delved into this similarity, with Geva et al. (2021) revealing that in feed-forward layers, each key correlates with textual patterns in training examples, while each value induces a distribution over the output vocabulary. In the context of Neural Machine Translation, Huang et al. (2023) demonstrates that feed-forward layers encapsulate crucial translation knowledge and can facilitate knowledge transfer between models.

3 Methods

3.1 Overview

Building upon prior research that characterizes feed-forward layers as neural memory repositories of knowledge, we posit a hypothesis that effective allocation and protection of these memories within feed-forward layers can facilitate both the acquisition of new knowledge and the prevention of forgetting. Our proposed method, F-MALLOC, is devised on the premise of this hypothesis.

F-MALLOC is specifically tailored to the feed-forward structure, with all other parameters held constant throughout the process. To preserve critical general domain knowledge while allowing flexibility for future task learning, we initiate the process with a structured pruning method (3.2). This method aids in eliminating unimportant memories, making them 'writable'. Subsequently, we introduce learnable task masks to manage these free memories (3.3). These task masks, acquired through learning, play a vital role in memory allocation for new tasks, designating them as 'read-only' to prevent alterations. For an overview of our method, please refer to Fig.1.

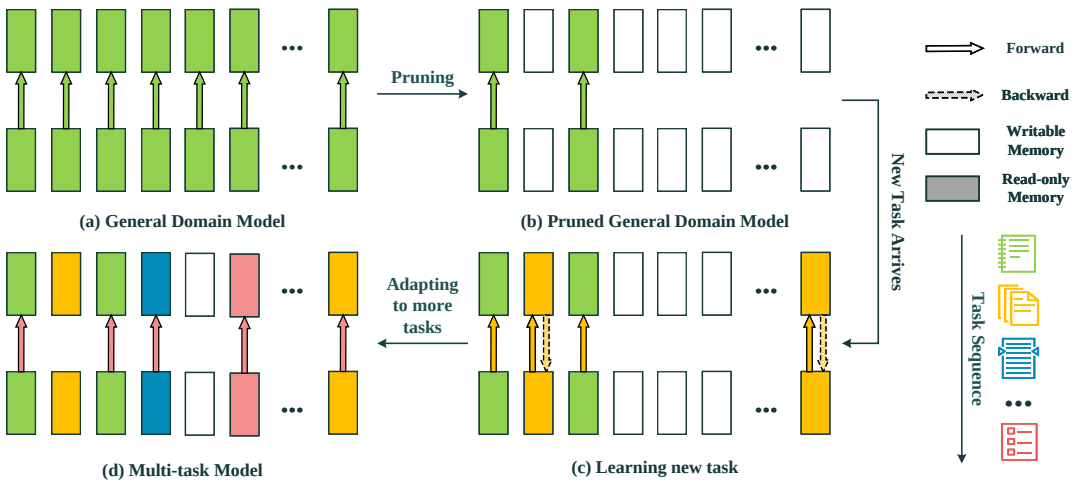

Figure 1: Illustration of F-MALLOC. For simplification, we depict a decomposed feed-forward layer. (a) **The Original General Domain Model:** Highlighting the general domain task in green. (b) **Pruned General Domain Model:** Post-pruning, pruned memories are 'writable' (depicted in white), while others are designated as 'read-only.' (c) **Learning a New Task:** The model learns to allocate some memories to the new task and mark them 'read-only' (depicted in yellow). 'read-only' memories remain available for future tasks' forward propagation. However, backward propagation through them is prohibited. (d) **Multi-task Model:** After learning all tasks, each task occupies a share of memory capacity. The forward pass of the last task is shown.

3.2 Preserving General Domain Knowledge

Pruning has demonstrated effectiveness in retaining essential parameters while eliminating unnecessary ones in neural networks. In this context, we adopt a structured pruning method, which is designed to preserve general domain knowledge. The pruning process calculates an importance score for each memory cell in the feed-forward layer, retaining only the most crucial ones.

Importance-based memory pruning. The pruning problem can be seen as finding an optimal mask under a sparsity constraint. To formalize this, we decompose the feed-forward layer into N key-value pairs, which we call a memory cell[2]. Subsequently, a mask is introduced to control the activation of them:

$$\mathrm{FF}(x, m) = \sum_{i=1}^{N} m_i \odot W^{(2)}_{:,i} \cdot \sigma(W^{(1)}_{i,:} \cdot x) \qquad (2)$$

Here, N denotes the hidden dimension of the feed-forward layer, $m \in \{0,1\}^N$ represents the mask vector and $\odot$ denotes the Hadamard product.

A common approach to select unnecessary memory is to estimate the importance of different memories with gradient (Michel et al., 2019) or Fisher information (Liu et al., 2021). The precise calculation of the importance score typically demands the use of the same data and loss functions employed during model training, which is often impractical in CL scenarios where obtaining the training data of a pretrained model may be unfeasible.

To address this challenge, we propose an alternative approach employing Jensen–Shannon (JS) divergence. The method draws inspiration from the stochastic dropout mechanism (Hinton et al., 2012), which introduces randomness by eliminating a portion of units in each layer during training, mitigating co-adaptation and overfitting. In our approach, dropout is applied to the feed-forward layer, generating unique memory activations and distinct outputs during each forward pass. By comparing these outputs and computing the gradient of the divergence, we derive a novel importance score for memories.

Specifically, we perform two forward passes of the input data x through the network, generating two distributions of model predictions, denoted as $\mathcal{P}_1(y|x)$ and $\mathcal{P}_2(y|x)$. We then calculate the JS divergence between these predictions:

$$\mathcal{L}_{\mathrm{JS}}(x) = \frac{1}{2}\big(\mathrm{KL}(\mathcal{P}_1(y|x), \mathcal{P}_2(y|x)) + \mathrm{KL}(\mathcal{P}_2(y|x), \mathcal{P}_1(y|x))\big) \qquad (3)$$

where $\mathrm{KL}(\cdot,\cdot)$ denotes the Kullback–Leibler (KL) divergence. In practice, we adopt an external dataset $\mathcal{D}$ and estimate the average gradient of

[2]We use memory and memory cell interchangeably.

JS divergence, serving as an empirical importance score:

$$I_k = \mathbb{E}_{x \in \mathcal{D}} \left| \frac{\partial \mathcal{L}_{\text{JS}}(x)}{\partial m_k} \right| \tag{4}$$

Following the derivation of the importance score, a binary mask is generated through a binarization function utilizing the s quantile of the importance score, denoted as $q_s(I)$, as the threshold:

$$m_k^G = \begin{cases} 1, & \text{if } I_k \geq q_s(I) \\ 0, & \text{if } I_k < q_s(I) \end{cases} \tag{5}$$

where s is the desired sparsity. Substituting this mask m^G into Formula 2 accomplishes the pruning.

3.3 Learning New Domain Continually

After the structure pruning stage, wherein specific feed-forward memories are pruned and marked 'writable' for future learning, we introduce a task mask mechanism to manage memory. Throughout the forward pass, these task masks govern the activation of feed-forward memory, conditioning the model for specific tasks. In the backward pass, the task masks are employed to suppress gradient updates to the 'read-only' memories, effectively preventing CF. Fig.2 provides an overview of this procedure.

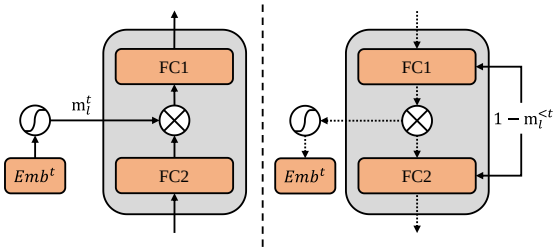

Figure 2: Illustration of new domain learning: forward (the left) and backward (the right) propagation. Here, we show the inner structure of the feed-forward layer.

Learning task mask to allocate 'writable' memory. To adapt to a new task t, a task mask m_l^t is learned. This task mask serves to conditionally activate the memories in the l-th feed-forward layer. We adopt the task-based hard attention mechanism proposed by Serrà et al. (2018) to train the mask. For each task t, a learnable task embedding e_l^t is introduced for each layer. The task mask m_l^t is defined as a gated version of the embedding vector e_l^t:

$$m_l^t = \sigma(\frac{e_l^t}{\tau}) \tag{6}$$

where σ represents a gate function, and τ is a temperature variable. We wish to learn a binary task mask that could be employed to allocate feed-forward memory in the same format as described in Eq.2.

To facilitate the efficient learning of task masks, we employ a sigmoid function with a temperature scalar to create a differentiable pseudo-gate function. The temperature scalar regulates the polarization or 'hardness' of the pseudo-step function. As $\tau \to 0$, the values of $m_{l,i}^t$ tend towards either 0 or 1, compelling the model to exploit allocated memories. Conversely, as $\tau \to \infty$, the values of $m_{l,i}^t$ approach 0.5, allowing the model to freely explore memories. Throughout the training process, we implement temperature annealing, transitioning from $\frac{1}{\tau_{max}}$ to τ_{max}. This dynamic adjustment aids the model in cyclically exploring memories while simultaneously exploiting activated memories. During the training process, the mask undergoes a gradual polarization, resulting in the occupation of useful memories. The embedding is initialized with $\alpha m^G - |\mathcal{N}(0,1)|$. This initialization set the extra capacity usage to zero at the beginning of training, promoting low capacity usage. Upon model convergence, we archive the acquired mask for future utilization.

Applying task mask to safeguard 'read-only' memory. To tackle the challenge of CF, our approach involves leveraging task masks acquired from previous tasks to influence the gradient. Before learning a new task, denoted as t, we aggregate all task masks from preceding tasks using an element-wise max (EMAX) operation and subsequently binarize the result with a threshold λ, as expressed by the following equation:

$$m_l^{<t} = \text{I}_\lambda(\text{EMAX}_{j<t}\{m_l^j\}) \tag{7}$$

Here, the subscript l denotes the layer index. In this specific context, task 0 corresponds to the general domain translation task, and the associated mask derived from structural pruning for the general domain is denoted as m^0. The aggregated binary mask $m_l^{<t}$ encapsulates critical memories designated as 'read-only' by previous tasks. The primary objective is to safeguard the parameters in these memories, preserving their functionality for previous tasks. To achieve this, we utilize the mask to adjust the gradient during the training of task t, as articulated in the following equation:

$$g_l^{'t} = g_l^t \odot (1 - m_l^{<t}) \tag{8}$$

where $\odot$ denotes the Hadamard product. This modification guarantees that memories crucial for previous tasks (entries with a value of 1 in $m_l^{<t}$) will have near zero gradients, thereby ensuring their preservation during the training of subsequent tasks.

4 Experiments

In our multi-stage CL experiments for NMT systems, we finetune a pretrained general domain model on T new domains successively[3]. The pretrained model is based on the WMT'19 German-English news translation task winner (Ng et al., 2019). To neutralize the impact of task order, we randomly generate five task order sequences and report the average result.

4.1 Data Preparation

In the context of structure pruning, we employ the WMT14 de-en translation data[4] as the external dataset. Additionally, we combine the WMT newstest datasets from 2019 to 2021[5] to form a comprehensive general domain test set. For the continual domain adaptation experiments, we utilize the OPUS multi-domains dataset (Koehn and Knowles, 2017), which has been re-split by Aharoni and Goldberg (2020). It includes German-English parallel data in five domains: Medical, Law, IT, Koran and Subtitles.

The details of all datasets mentioned above are shown in Appendix A.

4.2 Baseline and Implementation Details

We incorporate eight competitive methods for comparison in our experiments, which can be categorized into two groups: Non-Continual Learning (Non-CL) methods and CL methods. In the Non-CL category, **(1) Single-domain** and **(2) Mixed-domain** directly finetune the pretrained model on single or mixed domain data, achieving the **upper bound** performance. **(3) Adapter (Bapna and Firat, 2019)** inserts Adapters on each transformer block of the general domain model. In the CL category, we have **(4) Sequential Fine-tuning** continual finetunes the pretrained model sequentially; **(5) EWC (Thompson et al., 2019; Saunders et al., 2019)** adds elastic weight consolidation term to regularize loss; **(6) KD(Khayrallah et al., 2018; Dakwale and Monz, 2017)** use knowledge distillation to transfer knowledge; **(7) Dynamic-KD(Cao et al., 2021)** involves dynamic adjustments to the weight of KD loss. **(8) PTE(Gu et al., 2021)** prune the general domain model and learn target domain with free parameters. We have extended the baseline method designed for a single stage to multiple stages. Further details on these methods can be found in Appendix B.

In our proposed method, we configure the pruning sparsity to 0.2 and set the temperature hyperparameter τ_{max} to 400. For embedding initialization, we employ $\alpha = 5.0$, and the binarize threshold λ in Eq. 7 is set to 0.5. For more Implementation Details please refer to Appendix C.

4.3 Metrics

We adopt the BLEU score to evaluate the translation performance. Recognizing that post-training BLEU may not sufficiently capture the nuances in multi-stage CL, we introduce two additional metrics: **Forgetting Ratio (FR)** and **Saturation Ratio (SR)**.

- Inspired by (Liu et al., 2020), FR is defined as:

$$\mathrm{FR}^t = \frac{1}{t-1}\sum_{i=1}^{t-1}\frac{a_i^i - a_i^t}{a_i^i} \tag{9}$$

  where $a_i^j, \forall i \leq j$ represents the BLEU on the i-th domain after learning of j-th domain[6]. This metric is employed to quantify the stability (the ability to prevent forgetting).

- SR is defined as:

$$\mathrm{SR}^t = 1 - \frac{a_t^t}{a_t^M} \tag{10}$$

  where, a_i^M represents the BLEU of i-th domain in a mixed-domain training fashion, commonly regarded as the **upper bound** of CL methods. The saturation rate highlights the system's plasticity (learning ability) when encountering a new task, with a higher rate indicating lower plasticity.

5 Results and Analysis

Table 1 presents the post-training performances of all nine systems across six domains. Notably, F-MALLOC consistently outperforms all

[3] In our experiments, a task is a domain. Hence, we use task and domain interchangeably.

[4] https://www.statmt.org/wmt14/translation-task.html

[5] https://www.statmt.org/

[6] This definition calculates the average proportion of performance degradation over all previously learned domains, excluding the latest one as it experiences no forgetting.

Category	Domain	General	IT	Koran	Law	Medical	Subtitles	Average	FR[%]	Additional storage
	Method	BLEU								
Non-CL	Single-domain	38.00	**48.80**	22.90	57.15	55.93	32.01	**42.47**	-	$T \cdot M$
	Mixed-domain	21.24	46.17	22.97	**60.35**	**55.98**	29.87	39.43	-	0
	Adapter	38.00	44.09	22.48	53.31	51.23	**32.05**	40.19	-	$T \cdot A$
CL	Seq-finetune	15.81	29.29	12.16	23.90	26.50	20.76	21.40	47.80	0
	EWC	24.57	36.93	17.61	46.14	43.92	25.01	32.36	11.47	$2M$
	KD	22.80	34.49	13.93	36.33	38.00	24.88	28.41	32.79	M
	Dynamic-KD	27.88	31.84	14.33	40.05	39.78	23.53	29.57	15.33	M
	PTE	37.00	42.82	**23.06**	52.65	49.59	31.69	39.47	-	$T \cdot M[bit]$
	F-MALLOC(Ours)	**39.54**	44.33	23.02	53.77	51.62	31.16	40.57	0.71	$T \cdot E$

Table 1: BLEU and FR for all domains post-training. The results are averages of 5 different task sequences (Non-CL baselines are independent of task order). The best results are highlighted in bold. The best CL results are highlighted with an underline. '-' indicates the corresponding methods have no forgetting. The specital tokens denote number of seen tasks(T), adapter size(A), model parameter size(M), binary mask size($M[bit]$) and task embedding size(E). Note that $M \gg M[bit] > A \gg E$.

Domain	Genearl	IT	Koran	Law	Medical	Subtitles	Average	FR[%]
Method	BLEU							
Seq-finetune	21.02	23.15	11.80	31.33	36.83	30.65	25.80	37.40
EWC	22.86	45.81	18.96	43.37	39.18	26.16	32.72	10.10
KD	25.91	29.52	13.40	40.31	44.68	31.61	30.91	27.14
Dynamic-KD	30.35	33.83	15.56	41.65	40.90	24.62	31.15	12.22
PTE	37.00	43.28	**22.98**	52.94	49.42	31.87	39.58	-
F-MALLOC(Ours)	**39.54**	**44.19**	22.81	**53.64**	**51.21**	**31.93**	**40.55**	**0.24**

Table 2: BLEU and FR of CL methods for all domains post-training using task sequence 0 (the domain training order corresponds to the sequence in the first row). The best results are highlighted in bold.

CL baselines on average, with an impressively low forgetting rate of 0.71%. In comparison with regularization-based baselines, F-MALLOC demonstrates a better ability to alleviate forgetting and acquire new knowledge. When compared with the SOTA architecture-based method, PTE, F-MALLOC attains higher performance with minimal storage overhead and no prior information about task numbers.

Regrading the Non-CL baselines, although still trailing behind the upper bound performance, F-MALLOC demonstrates comparable performance to the strong baseline method, Adapter. These results collectively underscore the effectiveness of F-MALLOC in Continual Learning scenarios for Transformer-based Neural Machine Translation (NMT) systems.

For a more comprehensive analysis and comparison of various CL methods, the following subsections will use task sequence 0: IT → Koran → Law → Medical → Subtitles as a reference.

5.1 Comparison with CL methods

Table 2 presents the results for task sequence 0. Notably, among the prior CL methods, PTE stands out with the best performance, achieving a BLEU score of 39.58 and zero forgetting. In contrast, regularization-based methods exhibit inferior performance. The suboptimal results of KD-based approaches (KD and Dynamic-KD) can be attributed to the absence of sample replay in our experimental setting. Without a sample cache from previous tasks, KD struggles to effectively transfer knowledge from the preceding model to subsequent ones. Importantly, F-MALLOC surpasses all CL baselines, delivering the best results in both the BLEU score and forgetting rate.

Robustness against domain order. A horizontal comparison between Table 1 and Table 2 for the same method's performance reveals that regularization-base methods such as EWC and KD are sensitive to domain order, resulting in imbalanced performance on the initial and final tasks. In contrast, F-MALLOC exhibits notable resilience to variations in domain order, as evidenced by the balanced performance across different domain orders. This robustness is further substantiated by the low standard deviations presented in Appendix D.

Trade-off between stability and plasticity. As depicted in Fig.3, both EWC and Dynamic-KD

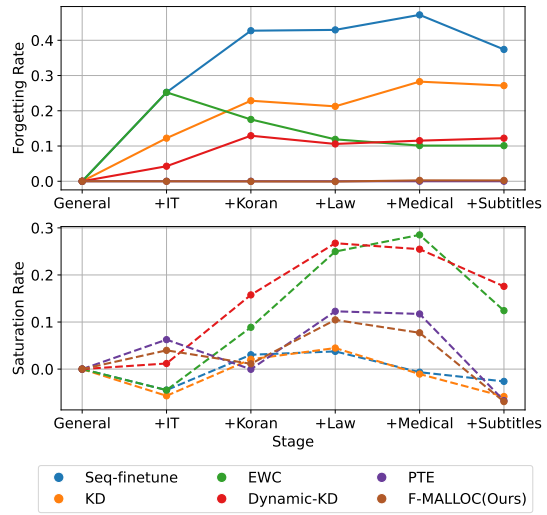

Figure 3: Forgetting rate and saturation rate across different training stages.

exhibit robust abilities to mitigate forgetting. However, they also demonstrate a high saturation rate, suggesting a compromise in their potential to adapt to additional tasks. In contrast, KD achieves a low saturation rate akin to Seq-finetune, but its forgetting rate is notably higher. This observation sheds light on the struggle of regularization-based methods to balance stability and plasticity. Crucially, F-MALLOC excels in both objectives, achieving a harmonious equilibrium between mitigating forgetting and maintaining adaptability.

5.2 Hyperparameter

Temp	BLEU	FR[%]	Sparsity	BLEU	FR[%]
$\tau_{max} = 50$	36.83	10.81	$s = 0.05$	39.09	0.40
$\tau_{max} = 100$	38.33	6.89	$s = 0.1$	39.87	0.15
$\tau_{max} = 200$	40.22	2.37	$\mathbf{s = 0.2}$	**40.55**	**0.24**
$\boldsymbol{\tau_{max} = 400}$	**40.55**	**0.24**	$s = 0.3$	40.01	0.85
$\tau_{max} = 800$	40.60	0.10	$s = 0.4$	39.88	2.02

Table 3: The effect of max temperature τ_{max} (left) and sparsity s (right). The value used in our experiments is highlighted in bold.

We explored the impact of annealing temperature τ_{max} and prune sparsity s. As outlined in Table 3, a small temperature results in a 'soft' mask value, contributing to increased FR. Good results were observed when $\tau_{max} \geq 400$. Continually increasing the temperature renders the annealing strategy ineffective, resulting in a slower convergence speed.

In terms of prune sparsity, low sparsity restricts the available capacity for subsequent tasks, while high sparsity adversely affects general domain performance, both contributing to diminished overall performance. Notably, the performance gap across varying prune sparsity levels is relatively small, highlighting the robustness of F-MALLOC.

5.3 Analyzing Memory Capacity Allocation

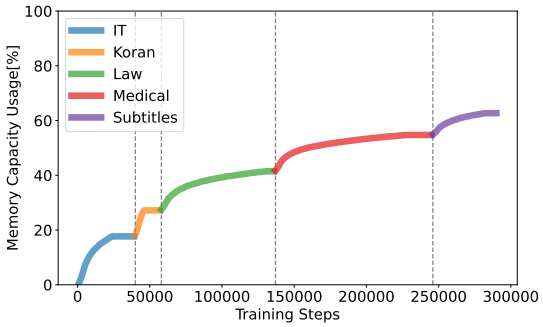

Figure 4: Feed-forward memory capacity usage in the training process of task sequence 0. Vertical dash lines indicate task switches.

F-MALLOC employs a task mask mechanism for the dynamic allocation of feed-forward memories to different tasks. Therefore, by computing the accumulated task mask $m^{<t+1}$ and subsequently binarizing it, we can assess the proportion of allocated memories. As depicted in Fig.4, the capacity usage undergoes rapid growth in the initial training stage for all tasks, gradually converging at a stable rate thereafter.

Upon comparing different tasks, we observed a positive correlation between the usage and the volume of data across diverse domains. However, the capacity usages are not strictly proportional to the dataset size, and with an increasing number of learning tasks, there is a trend of reduced occupancy for new tasks. These phenomena suggest that our proposed method has learned a rational and efficient memory allocation strategy, which leverages the inherent complexity of the tasks. Towards the conclusion of the entire training process, approximately 40% of the feed-forward memory is still 'writable'. However, the best-performing baseline, PTE, has already exhausted model capacity. This emphasizes the potential of our proposed method to effectively accommodate additional tasks.

5.4 Knowledge Transfer and Domain Similarity from Memory Reusing

In our proposed method, we employ non-exclusive task masks, allowing feed-forward memories allocated to previous tasks to be reused by subsequent

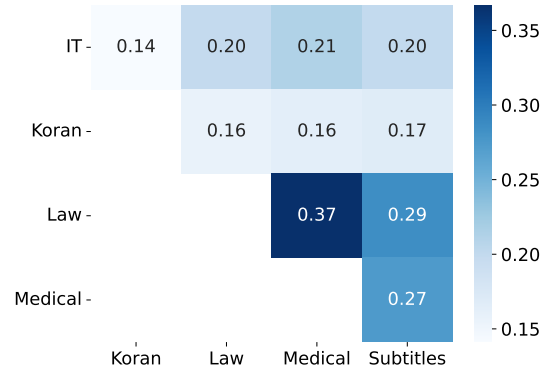

Figure 5: Percentage of memory reuse across tasks.

tasks. To investigate the inter-task relationship regarding the allocation of memories, we visually represent the overlap rate among task masks for different tasks. Specifically, we utilize the Jaccard similarity coefficient, defined as $\frac{|m^i \cap m^j|}{|m^i \cup m^j|}$, to assess the memory reuse between task t_i and t_j, $i < j$. The results, depicted in Fig.5, reveal a substantial proportion of memory reuse between different tasks. This observation underscores the effectiveness of our non-exclusive masking strategy in facilitating knowledge transfer between tasks.

We further conducted a comparative analysis with the unsupervised domain clustering approach proposed by Aharoni and Goldberg (2020). The observed memory reuse rate aligns consistently with domain similarity. For instance, the memory reuse rate of 'Koran' is notably lower than in other domains, reflecting its isolated nature with minimal intersection with other domains. In contrast, the 'IT' domain exhibits a nearly uniform memory reuse rate among 'Law', 'Medical', and 'Subtitles', consistent with the observation that it shares commonalities with these three domains according to the domain cluster result. This alignment highlights the effectiveness of our approach in capturing and leveraging task similarities for improved knowledge transfer.

6 Related Work

CL for NMT. Recent work on CL of NMT can be divided into two categories: regularization-based and architecture-based. Regularization-based techniques address forgetting by incorporating penalty terms to constrain the divergence of model parameters from their previous values. Prominent methods, including Elastic Weight Consolidation (EWC) (Thompson et al., 2019; Saunders et al., 2019) and knowledge distillation (Dakwale and Monz, 2017; Khayrallah et al., 2018; Zhao et al., 2022; Cao et al., 2021), are widely acknowledged for their effectiveness in the fine-tuning process. Gu et al. (2022) introduced a hard Low Forgetting Risk restriction on all parameters. In contrast to these approaches, our method effectively mitigates forgetting by blocking gradients, showcasing a more efficient strategy.

Architecture-based methods involve dividing the model into disjoint task-specific components. For instance, Gu et al. (2021) prune the general domain model and subsequently finetune free parameters to adapt to the target domain. Another approach, as demonstrated by Liang et al. (2021), involves freezing Lottery Ticket Subnetworks to prevent forgetting. Additionally, Huang et al. (2023) propose utilizing external models' feed-forward layers and embeddings as a plug-in for knowledge transfer. In comparison to these methods, our approach stands out as it requires no pre-specification of task numbers or space allocation. Moreover, it avoids the need to store an external model or a mask matrix.

Unstructured Pruning for Transformers. For coarse-grained unstructured pruning of Transformer models, attention-head pruning (Voita et al., 2019; Michel et al., 2019), layer-dropping (Fan et al., 2020) and block pruning (Lagunas et al., 2021) have been popularly used. Our proposed pruning method shares similarities with the approach presented by Kwon et al. (2022), although it differs in the estimation of importance and the selection of modules slated for pruning.

7 Conclusion

This paper introduces F-MALLOC, a pioneering method for CL in NMT systems. By decomposing feed-forward layers into memory cells and implementing a strategic memory allocation approach, F-MALLOC proves effective in simultaneously enhancing new knowledge acquisition and alleviating forgetting. Evaluation with a specialized protocol for CL in NMT, positions F-MALLOC as a superior performer, showcasing substantial improvements, robustness, and extensibility compared to existing approaches. The method's ability to leverage task difficulty and inter-task similarities for enhanced performance represents a significant advancement not seen in previous methods. F-MALLOC not only contribute to the field of CL in NMT but also pave the way for more efficient and adaptable neural network architectures.

8 Limitations

Although our proposed F-MALLOC can effectively alleviate forgetting and exhibits high robustness and extensibility, there are several limitations in our current study: On the one hand, F-MALLOC utilizes a fixed-capacity Transformer, which may limit its capability to adapt to an unrestricted number of tasks. On the other hand, F-MALLOC is designed for domain incremental training. Thus, adding a new language can not be directly solved. We leave these problems for future research.

References

Roee Aharoni and Yoav Goldberg. 2020. Unsupervised domain clusters in pretrained language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7747–7763. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Ankur Bapna and Orhan Firat. 2019. Simple, scalable adaptation for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 1538–1548. Association for Computational Linguistics.

Yue Cao, Hao-Ran Wei, Boxing Chen, and Xiaojun Wan. 2021. Continual learning for neural machine translation. 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 3964–3974. Association for Computational Linguistics.

Chenhui Chu and Raj Dabre. 2019. Multilingual multi-domain adaptation approaches for neural machine translation. *CoRR*, abs/1906.07978.

Chenhui Chu and Rui Wang. 2018. A survey of domain adaptation for neural machine translation. *CoRR*, abs/1806.00258.

Praveen Dakwale and Christof Monz. 2017. Fine-tuning for neural machine translation with limited degradation across in- and out-of-domain data. In *Proceedings of Machine Translation Summit XVI, Volume 1: Research Track, MTSummit 2017, September 18-22, 2017, Nagoya, Aichi, Japan*, pages 156–169.

Angela Fan, Edouard Grave, and Armand Joulin. 2020. Reducing transformer depth on demand with structured dropout. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Markus Freitag and Yaser Al-Onaizan. 2016. Fast domain adaptation for neural machine translation. *CoRR*, abs/1612.06897.

Robert M. French. 1993. Catastrophic interference in connectionist networks: Can it be predicted, can it be prevented? In *Advances in Neural Information Processing Systems 6, [7th NIPS Conference, Denver, Colorado, USA, 1993]*, pages 1176–1177. Morgan Kaufmann.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 5484–5495. Association for Computational Linguistics.

Shuhao Gu, Yang Feng, and Wanying Xie. 2021. Pruning-then-expanding model for domain adaptation of neural machine translation. 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 3942–3952. Association for Computational Linguistics.

Shuhao Gu, Bojie Hu, and Yang Feng. 2022. Continual learning of neural machine translation within low forgetting risk regions. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 1707–1718. Association for Computational Linguistics.

Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580.

Kaiyu Huang, Peng Li, Jin Ma, Ting Yao, and Yang Liu. 2023. Knowledge transfer in incremental learning for multilingual neural machine translation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 15286–15304. Association for Computational Linguistics.

Ferenc Huszár. 2017. On quadratic penalties in elastic weight consolidation. *CoRR*, abs/1712.03847.

Huda Khayrallah, Brian Thompson, Kevin Duh, and Philipp Koehn. 2018. Regularized training objective for continued training for domain adaptation in neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation, NMT@ACL 2018, Melbourne, Australia,*

July 20, 2018, pages 36–44. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation, NMT@ACL 2017, Vancouver, Canada, August 4, 2017*, pages 28–39. Association for Computational Linguistics.

Woosuk Kwon, Sehoon Kim, Michael W. Mahoney, Joseph Hassoun, Kurt Keutzer, and Amir Gholami. 2022. A fast post-training pruning framework for transformers. In *NeurIPS*.

François Lagunas, Ella Charlaix, Victor Sanh, and Alexander M. Rush. 2021. Block pruning for faster transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 10619–10629. Association for Computational Linguistics.

Jianze Liang, Chengqi Zhao, Mingxuan Wang, Xipeng Qiu, and Lei Li. 2021. Finding sparse structures for domain specific neural machine translation. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 13333–13342. AAAI Press.

Liyang Liu, Shilong Zhang, Zhanghui Kuang, Aojun Zhou, Jing-Hao Xue, Xinjiang Wang, Yimin Chen, Wenming Yang, Qingmin Liao, and Wayne Zhang. 2021. Group fisher pruning for practical network compression. 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 7021–7032. PMLR.

Yaoyao Liu, Yuting Su, An-An Liu, Bernt Schiele, and Qianru Sun. 2020. Mnemonics training: Multi-class incremental learning without forgetting. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 12242–12251. Computer Vision Foundation / IEEE.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? 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 14014–14024.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook fair's WMT19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation, WMT 2019, Florence, Italy, August 1-2, 2019 - Volume 2: Shared Task Papers, Day 1*, pages 314–319. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Demonstrations*, pages 48–53. Association for Computational Linguistics.

Danielle Saunders. 2022. Domain adaptation and multi-domain adaptation for neural machine translation: A survey. *J. Artif. Intell. Res.*, 75:351–424.

Danielle Saunders, Felix Stahlberg, Adrià de Gispert, and Bill Byrne. 2019. Domain adaptive inference for neural machine translation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 222–228. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Joan Serrà, Didac Suris, Marius Miron, and Alexandros Karatzoglou. 2018. Overcoming catastrophic forgetting with hard attention to the task. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 4555–4564. PMLR.

Sainbayar Sukhbaatar, Edouard Grave, Guillaume Lample, Hervé Jégou, and Armand Joulin. 2019. Augmenting self-attention with persistent memory. *CoRR*, abs/1907.01470.

Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. 2015. End-to-end memory networks. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 2440–2448.

Brian Thompson, Jeremy Gwinnup, Huda Khayrallah, Kevin Duh, and Philipp Koehn. 2019. Overcoming catastrophic forgetting during domain adaptation of neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics:*

Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), pages 2062–2068. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. 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 5998–6008.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 5797–5808. Association for Computational Linguistics.

Yang Zhao, Junnan Zhu, Lu Xiang, Jiajun Zhang, Yu Zhou, Feifei Zhai, and Chengqing Zong. 2022. Life-long learning for multilingual neural machine translation with knowledge distillation. *CoRR*, abs/2212.02800.

A Dataset Details

	Dataset	Train	Dev.	Test
	WMT17	3.9M	-	-
General	Newstest2019	-	-	2000
	Newstest2020	-	-	1000
	Newstest2021	-	-	785
	IT	223K		
	Koran	17K		
	Law	467K	2000	2000
	Medical	248K		
	Subtitles	500K		

Table 4: Dataset statistics.

Here, we present detailed statistics for the datasets used in our experiments in Table 4, focusing on the translation direction EN → DE. We employ Moses scripts[7] for sentence tokenization and truecasing. Additionally, we utilize FastBPE[8] to apply Byte Pair Encoding (BPE)(Sennrich et al., 2016) to the tokenized data. The dictionary and BPE codes are sourced from the Fairseq WMT19 German-English news translation pre-trained model(Ng et al., 2019).

B Baseline Details

Non-Continual Learning Methods. Each of these baselines constructs a distinct model (or module) for each task independently. Consequently, they do not encounter CF and lack knowledge transfer between tasks.

- **Single-domain** continues to train the general domain model on target domain data, respectively.
- **Mixed-domain** trains the general domain model on combined multi-domain data, which is considered the **upper bound** of CL methods.
- **Adapter (Bapna and Firat, 2019)** inserts adapters on each transformer block of the general domain model as proposed by Bapna and Firat (2019). We set the bottleneck dimension to 64 and only finetune the adapters.

[7] http://www.statmt.org/moses/
[8] https://github.com/glample/fastBPE

Method	Domain					
	General	IT	Koran	Law	Medical	Subtitles
Seq-finetune	6.28	17.62	6.35	12.91	11.69	9.20
EWC	5.23	5.64	1.55	4.70	6.12	5.46
KD	3.97	12.85	4.91	8.69	5.74	6.16
Dynamic-KD	2.59	1.81	0.87	1.85	1.49	0.76
PTE	0.00	0.60	0.23	0.33	0.24	0.40
F-MALLOC(Ours)	0.00	0.71	0.27	0.62	0.61	0.57

Table 5: Standard deviation of BLEU score of the proposed F-MALLOC and CL baselines over 5 random task sequences.

Domain order	EWC		PTE		F-MALLOC	
	BLEU	FR[%]	BLEU	FR[%]	BLEU	FR[%]
IT→Koran→Law→Medical→Subtitles	32.72	10.10	39.58	-	40.55	0.24
Koran→Medical→IT→Law→Subtitles	32.78	12.11	39.36	-	40.70	0.32
Law→IT→Medical→Subtitles→Koran	29.94	16.40	39.55	-	40.76	1.12
Subtitles→Law→Koran→Medical→IT	35.21	5.15	39.48	-	40.39	0.69
Medical→Law→Koran→Subtitles→IT	31.16	13.59	39.37	-	40.47	1.17

Table 6: Result in different domain orders. The best-performing regularization-based baseline, EWC, and architecture-based baseline, PTE, were chosen for comparison.

Continual Learning Methods:

- **Sequential Fine-tuning** continues to train the general domain model on target domains sequentially, without incorporating any mechanism to address CF.

- **Elastic Weight Consolidation (EWC) (Thompson et al., 2019; Saunders et al., 2019)** is a popular regularization-based CL method that adopts elastic weights consolidation to introduce L_2 regularization, penalizing parameter changes. The training objective is:

$$\mathcal{L}_{\text{EWC}}(\theta) = \mathcal{L}_{\text{CE}}(\theta) + \alpha \sum_i F_i(\theta_i - \theta_i^G)^2$$

  In this equation, θ represents the model parameters, F is the diagnosis of the Fisher information matrix, and α is a hyperparameter controlling the strength of regularization. To extend this method to a multi-stage scenario, we adopt the accumulated Fisher information, as proposed by Huszár (2017).

- **Knowledge Distillation (KD) (Khayrallah et al., 2018; Dakwale and Monz, 2017)** introduces a regularization (reg) term into the training objective. The reg term is formulated in the spirit of knowledge distillation, minimizing the cross-entropy between the original (teacher) model's output distribution and that of the new (student) model. A hyperparameter α is introduced to interpolate the reg term and the NLL loss.

$$\mathcal{L}_{\text{EWC}}(\theta) = \mathcal{L}_{\text{CE}}(\theta) + \alpha \mathcal{L}_{\text{KD}}(\theta)$$

  In our experiments, the weight of the KD term is set to 0.1.

- **Dynamic Knowledge Distillation (Dynamic-KD) (Cao et al., 2021)** propose dynamically adjusting the weight of KD loss to better alleviate CF in a multi-stage CL scenario. The bias correction module is omitted due to its incompatibility with the pretrained model.

- **Prune Then Expand (PTE) (Gu et al., 2021)** employs unstructured pruning to trim the general domain model, followed by training the pruned parameters for the target domain. In the context of multi-stage CL, we uniformly distribute the pruned parameters across all subsequent tasks.

C Implementation Details

pretrained Model. All methods are implemented with the Fairseq toolkit (Ott et al., 2019). We adopt the WMT'19 German-English news translation task winner (Ng et al., 2019) as the pretrained general domain model. It is a Transformer encoder-decoder model (Vaswani et al., 2017) with 6 layers, 1,024-dimensional representations, 8,192-dimensional feed-forward layers, and 8 attention heads. Apart from WMT'19 training data, this model is trained on over 10 billion tokens of back-translation data and finetuned on the Newstest test sets from years before 2018. In our experiments, we do not use ensembles or n-best reranking.

Hyper-parameters. Unless explicitly stated otherwise, consistent hyperparameters are applied across all experiments. We utilize the Adam optimizer (Kingma and Ba, 2015) with the same learning rate scheduler as detailed in Vaswani et al. (2017). The learning rate is set to 1e-4 for all systems during the fine-tuning process. Training is stopped when there is no performance improvement for 5 consecutive validation steps.

In our proposed method, we exclusively finetune the Feed-forward layers in Transformers, keeping all other modules frozen throughout the procedure. During the structure pruning stage, we set the pruning sparsity to 0.2 for subsequent CL experiments (the same pruning sparsity is also used in PTE for fair comparison).

In the CL stage, the temperature hyperparameter τ_{max} is set to 400, following previous work (Serrà et al., 2018). We use $\alpha = 5.0$ in the embedding initialization. The binarize threshold λ in Eq.7 is set to 0.5. For inference, we employ beam search with a beam size of 5 for all systems. The default parameter of BLEU is utilized in evaluation.

All experiments are done on 8 NVIDIA RTX 3090 GPUs.

D Standare Deviations

This section reports the standard deviations of the results in Table 1. We only include the CL baselines here, since Non-CL baselines' performance is independent of the domain order. As shown in Table 5, F-MALLOC achieves the lowest standard deviations, indicating its robustness.

E Result in Different Domain Orders

Table 6 shows the performance of F-MALLOC along with two strong baselines, EWC and PTE, in other domain orders. F-MALLOC outperforms both EWC and PTE, highlighting its efficacy across different domain order scenarios.