CONTINUAL MODEL EVOLVEMENT WITH INNER-PRODUCT RESTRICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

With pre-trained model's rapid deployment in natural language processing (NLP) applications, it is intuitive to expect these models can continuously evolve when the task requires more complicated inference abilities of the model. Existing continual learning (CL) problem setups and methods focus on fixing out-of-distribution (OOD) data streams which cannot solve such a new challenge. We propose a continual model evolvement problem formulation (CME) that introduces a new challenge for fine-tuned pre-trained models that requires them to evolve during deployment. We formulate the problem and introduce multiple metrics to assess current CL methods from different aspects. Further, we propose a strong method dubbed inner-product restriction as a headstart in solving the CME problem. Experimental results indicate that the CME is still challenging to current deployed pre-trained models while our proposed method can provide a strong boost based on previous CL methods, supporting that it is of great need to explore the CME challenge for better deployment of pre-trained models in NLP applications.

1 INTRODUCTION

Recent development of pre-trained language models stimulates the NLP field with the deployment of fine-tuned pre-trained models Devlin et al. (2019). Once these models are deployed, they cannot be changed and therefore become substantially incomparable to humans. That is, these models cannot evolve even in the range of the given assignment (a specific downstream task such as machine translation, text classification, reading comprehension, etc.). It is highly preferred to explore the possibility of continual model evolvement when the deployed models encounter more challenging samples which cannot be properly predicted. In previous practice, researchers hope to teach these deep neural networks to adapt to new datastreams or refine the reported errors during deployment. These new datastreams can be out-of-distribution samples or corrected knowledge (e.g. transferring a wikipedia-based SQuAD dataset trained model to solve HotPotQA task). Therefore, such a learning process falls into a continual learning paradigm where deployed models are further refined through small batches of samples and able to solve these new samples from another domain.

However, the abilities of error fixings or adaptions to new datastreams cannot cover real-world NLP applications especially when users are having extremely high expectations of these newly invented pre-trained language models. For example, model refinement to new datastreams tackles problems such as converting a SQuAD Rajpurkar et al. (2016) dataset based reading comprehension model to a model that can answer HotPotQA Yang et al. (2018) questions. Real-world users may expect the models to fully understand the provided context and the questions so they may challenge the model with unanswerable questions. In such a scenario, the model initially trained without the ability to recognize unanswerable questions will give unreasonable predictions which seriously depress users. Therefore, deployed models are supposed to learn to continually evolve and recognize these challenging examples such as unanswerable questions while they are in service instead of training a new model as a brand new task. Such a challenge is not covered by the current continual learning paradigm since the required ability such as recognizing the unanswerable questions are more challenging than previous domain-shift setups.

Therefore, in this paper, we propose a continual model evolvement (CME) problem formulation, which tests whether fine-tuned pre-trained models can *evolve* and deal with more challenging in-

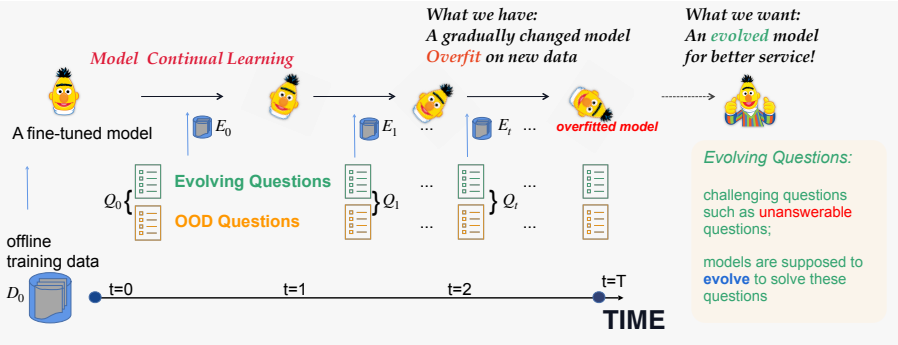

Figure 1: Illustration of the Continual Model Evolvement Challenge Setup.

coming datastream without re-training the entire model with another round of massive data collection and annotation. Specifically, we use the generative framework in the question answering task as a testbed to construct the CME challenge. We use the reading comprehension task as an example to test whether a question answering (QA) model can evolve to learn to solve more challenging tasks such as recognizing unanswerable questions. It is worth mentioning that we take the evolving ability to recognize unanswerable questions as an example. The real-world scenario might require more complicated questions that the model has to evolve to solve.

Considering that the Continual Model Evolvement is a new challenge to pre-trained models, we provide several strong methods as well as strong baselines as a headstart for the challenge. One major concern in CME is to maintain the original task performance which is to solve the catastrophic forgetting problem in the traditional CL paradigm. Therefore, we introduce an **Inner Product Restriction** strategy for the model learning process to narrow the gap between the original task and the evolved task.

Specifically, we dive into the experience replay method Rolnick et al. (2019) which is widely used in continual learning problems, and consider that in the CME challenge, the replay task and the continual learning goal might contradict each other. Therefore, we calculate the gradient between two tasks and restrict the inner product between task gradients. We propose two strategies: (1) we can add such an inner product restriction as a regularization term in the replay process; (2) we use the inner product restriction to optimize the model directly. Through our proposed inner-product restriction algorithm, we can mitigate the catastrophic forgetting problem which is a key challenge in the Continual Model Evolvement setup.

We conduct extensive experiments and obtain many non-trivial observations. We first construct experiments on general Continual Learning methods to study the Continual Model Evolvement setup. Then we test our headstart method and compare the performances with CL baselines. Experimental results reveal that (1) current CL methods cannot tackle the CME challenge which may hurt user experience in real-world NLP applications; (2) our proposed headstart method successfully mitigates the catastrophic forgetting problem and obtains significant improvement over the traditional CL method in the CME challenge without introducing additional parameters or meta-networks.

2 Continual Model Evolvement

2.1 Problem Formulation

Upstream Model In continual model evolvement, a deployed model f is trained offline with annotated data which could be time-consuming. Following the continual model refinement setup, we name the offline-trained model as upstream model f_0 and the original data used to train the upstream model as upstream data D_0.

Challenging Input Streams After deployment, the deployed model will encounter massive input samples and among them some *challenging* input samples that cannot give correct predictions. Supposing that the incoming samples are in T episodes $\{S_1, ..., S_T\}$ and among these samples, there

are errors $\{E_1, ..., E_T\}$ the upstream model cannot solve which we consider as challenging samples. The goal of Continual Model Evolvement is to tune the upstream model f_0 with the error questions E_t to evolve to model f_t that can answer these challenging incoming samples. Meanwhile, catastrophic forgetting is one major problem in continual learning tasks. That is, the model needs to learn to solve the challenging incoming samples, while it should not forget the ability to deal with the upstream samples. In the model evolving challenge, such a problem is more severe since the evolving task can be a brand new task from the perspective of a neural model while it may be considered similar to the upstream task from the perspective of human users.

2.2 IN-PRACTICE TESTBED FORMULATION

Generally, sequence-to-sequence problems can be used in solving a great number of NLP tasks including but not limited to translation, summarization, dialogue and text classification tasks. Therefore, we formulate the question answering task as a sequence-to-sequence generation task and use question answering task as a testbed to test the CME challenge. Then we suppose that deployed upstream model f_0 is a question answering (QA) model trained with the most widely used datasets such as the SQuAD dataset Rajpurkar et al. (2016) as upstream data D_0 using sequence-to-sequence pre-trained models exemplified by BART Lewis et al. (2020) as a generative task.

We assume that during the deployment of the upstream model f_0, the upstream model will encounter different user questions which may contain many challenging samples that current upstream model cannot tackle. As a testbed, we simplify the challenging input streams to two types of questions that the model should evolve and manage to solve.

Evolving Question In question answering systems, unanswerable questions are very challenging for comprehension systems trained without knowing the existence of such patterns. Therefore, *such patterns can be considered as a form of questions that the model needs to evolve to solve.* That is, the input domain is similar to the upstream data yet the incoming questions cannot be properly answered within the given context. In real-world applications, users might count the neural models and require answers which is challenging *including but not limited to* asking unanswerable questions. These unanswerable questions are considered as knowledge that the model needs to evolve to learn.

OOD Question Besides unanswerable questions as challenging questions for the upstream model, out-of-distribution questions can also be asked by users since the upstream data obviously cannot cover all user needs. Therefore we also include some OOD questions in the CME challenge. These out-of-distribution questions can be considered as examples from a similar dataset to the upstream data (e.g. a HotPotQA question is an OOD sample for the upstream model trained with the SQuAD dataset.) These questions are considered as knowledge that the model can learn through domain adaptation or generalization.

2.3 EVALUATION OF CME

In the formulation of the Continual Model Evolvement challenge, it is of great importance to maintain the original upstream model performance when evolving the model for new-coming samples. Considering that new-coming challenging samples are errors to the upstream model, we introduce Error Fixing Rate (EFR) and Upstream Knowledge Retention (UKR) scores to test the new-coming sample fixing ability and the original task performance. During the entire evolving period, we need to test how much knowledge is learned during the evolving process and we use knowledge retention of the challenge and the OOD samples, which are Evolved Knowledge Retention (EKR) and OOD Knowledge Retention (OKR) to evaluate. Further, the evolved model is supposed to generalize the evolved ability. That is, besides testing how well the model solves the new-coming challenging samples, we preserve a testset of both evolved knowledge samples and OOD knowledge samples and use Evolved Knowledge Generalization score (EKG) and OOD Knowledge Generalization score (OKG) to evaluate how well the evolved model generalizes to these samples.

• **Error Fixing Rate (EFR)** **EFR**(t) is to test the model response to fixing the input streams S_t of the t^{th} episode. Here, label y is the exact match label of a input sample in the question answering task: **EFR**$(t) =:$ **Acc**$(f_t, E_t) =: \frac{|\{(x,y)\in E_t \mid f_t(x)=y\}|}{|E_t|}$.

• **Upstream Knowledge Retention (UKR)** The upstream knowledge retention is the most important metric in the CME challenge since not losing accuracy of the original job is a must-to-do in real-world applications. We select a certain testset $D_0^{'}$ to test the upstream knowledge retention during continual model evolving: $\mathbf{UKR}(t) =: \mathbf{Acc}(f_t, D_0^{'})$

• **Evolved/OOD Knowledge Retention and Generalization (E/OKR and E/OKG)** Knowledge retention is to test how well the evolved model learns the samples encountered and the knowledge generalization is to test on a stream of unseen samples assuming that the evolved model can generalize to the new challenges. Therefore, we calculate Evolved/OOD Knowledge Retention (EKR and OKR) on all encountered challenging samples $S^c_{<t}$ and OOD samples $S^o_{<t}$ and Evolved/OOD Knowledge Generalization (EKG and OKG) on additional subsets S^c_{t+1} and S^o_{t+1}: $\mathbf{EKR}(t) =: \mathrm{Acc}(f_t, S^c_t), \mathbf{OKR}(t) =: \mathrm{Acc}(f_t, S^o_t), \mathbf{EKG}(t) =: \mathrm{Acc}(f_t, S^c_{t+1})$. $\mathbf{OKG}(t) =: \mathrm{Acc}(f_t, S^o_{t+1})$.

2.4 Exclusive Challenge of CME

Compared with previous continual learning and model refinement challenges, the unique challeng of Continual Model Evolvement is that during the continual learning process, the learning object is more realistic and challenging. Compared with previous continual learning paradigm Biesialska et al. (2020), where the continual learning objects are different downstream tasks, the CME challenge is a realistic scenario that constantly happens in real-world applications. The concept of model evolving task is a scenario that users might ask unexpected questions that might not be seen in the training set patterns. On the contrary, users might not ask two entirely different tasks when they are told the AI service is designed to solve a certain task such as question answering. Compared with continual model refinement Lin et al. (2022) where the continual data stream only focuses on different domains of question answering scenarios, where the task is less challenging and the catastrophic forgetting problem is less harmful considering that the general task setup does not change. The model evolving concept can be expanded with future deployment of pre-trained NLP models in more challenging scenarios where the refinement is relatively limited. Compared with knowledge editing in pre-trained models Zhu et al. (2020); De Cao et al. (2021); Mitchell et al. (2021), continual model evolvement focuses on the fine-tuned model evolving instead of editing a specific knowledge learned during the pre-training process.

3 Inner-Product Restriction

As a headstart in solving the Continual Model Evolvement challenge, we introduce an inner-product restriction algorithm to explore the potential in using current neural network construction and training methods.

3.1 Base Model and Basic CL Methods

As described in Continual Model Evolvement, we use question answering task as a testbed. Therefore, we incorporate a widely used pre-trained sequence-to-sequence language model BART-Base Lewis et al. (2020) as the base model.

Continual learning (CL) is the most straightforward solution that further fine-tune the upstream model to evolve to the new-coming streams. The core of CL methods is to avoid catastrophic forgetting problem McCloskey & Cohen (1989) during the continuous tuning of the upstream model. Generally, experience replay Rolnick et al. (2019) and regularization methods Kirkpatrick et al. (2017) are widely used in mitigating the problem. We give a brief summarization of these methods and then build the inner-product algorithm based on these CL methods.

Experience Replay Experience Replay McCloskey & Cohen (1989) is an effective replay method that maintain a memory module and save previously encountered samples as well as samples from the upstream data. Then we randomly select a subset from the memory to train the model at a certain period which is called replay. Such a method can mitigate the catastrophic forgetting problem in the traditional continual learning task setup by using previous encountered samples and upstream samples to tune the model. In the CME challenge testbed, with upstream training examples involved,

during the model evolving process, the model can learn the new ability (recognizing unanswerable questions) without forgetting the original purpose (answering questions).

Maximally Interfered Replay (MIR) A simple modification of the experience replay method is the Maximally Interfered Replay (MIR) method Aljundi et al. (2019). The MIR method replays the most forgettable examples from the memory. In the CME challenge testbed, we follow Lin et al. (2022) to design the most forgettable example retrieval rule. We first fine-tune model f_{t-1} with incoming data S_t to obtain a virtual model $f_t^{'}$ and calculate interference scores:

$$\text{score}(x_i, y_i) =: \text{loss}(f_t^{'}(x_i), y_i) - \text{loss}(f_{t-1}(x_i), y_i).$$

We consider replaying the samples with largest interference scores.

Regularization Methods In the traditional CL task, regularization method is another common solution. The core idea is to introduce a regularization term during the continual learning process.

The most straightforward regularization method is the L2-regularization between the parameter θ_t of model f_t: $\mathcal{L}_{\text{L2Reg}}(t) = \sum_i \left(\theta_t^i - \theta_{t-1}^i\right)^2$. Such a loss term can be added to the continual learning loss with a weight η as a constraint to avoid overfitting on the new-coming samples. One weakness of regularization methods is that regularization methods only restrict the continual optimized model during time so when the time step grows, the model behavior might be far from the upstream model. Therefore, regularization methods can be combined with replay methods in the continual learning tasks to obtain a better model.

3.2 Inner Product Restriction

In continual model evolvement challenge, the model evolving process might cause severe parameter change compared with domain adaption in traditional continual learning paradigm since the evolving task in CME can be significantly different from the upstream task, which could lead to a more challenging catastrophic forgetting problem.

To mitigate the catastrophic forgetting problem in the CME challenge, one possible solution is to assume that the upstream task loss and the evolving task loss could be at odds, though the evolving task is constructed from the upstream task. Therefore, we aim to narrow down the optimization contradiction between the upstream model and the evolved model.

Suppose the experience replay loss that optimizes the upstream task is $\mathcal{L}_{ft}(\theta)$ and the continual learning loss that optimizes the evolving task is $\mathcal{L}_{cl}(\theta)$ on the model f_{t-1} with parameter θ, for a timestep with both replay loss and continual learning loss, the total training loss is $\mathcal{L} = \mathcal{L}_{ft}(\theta) + \mathcal{L}_{cl}(\theta)$. These two losses can be contradictory to each other given the huge variance between the evolving task goal and the upstream task goal. Therefore, we are hoping that we can mitigate the conflict between these two losses.

In a continuous time step with learning rate η, we can re-write the experience replay loss $\mathcal{L}_{ft}$ to:

$$\mathcal{L}_{ft}(\theta) = \mathcal{L}_{ft}(\theta - \eta\nabla_\theta\mathcal{L}_{cl}(\theta)) = \mathcal{L}_{ft}(\theta) - \eta\nabla_\theta\mathcal{L}_{ft}(\theta)^T\nabla_\theta\mathcal{L}_{cl}(\theta) + \mathcal{O}(\eta^2). \tag{1}$$

The expanded term is $-\eta\nabla_\theta\mathcal{L}_{cl}(\theta)^T\nabla_\theta\mathcal{L}_{ft}(\theta)$ will increase the experience replay loss $\mathcal{L}_{ft}$ when the term $\nabla_\theta\mathcal{L}_{ft}(\theta)^T$ and the term $\nabla_\theta\mathcal{L}_{cl}(\theta)$ are in opposite directions and could cause a serious catastrophic forgetting problem. We denote the inner product as $\mathcal{I}(\theta) = \nabla_\theta\mathcal{L}_{cl}(\theta)^T\nabla_\theta\mathcal{L}_{ft}(\theta)$.

Therefore, we design two different strategies to mitigate the catastrophic forgetting problem caused by the conflict between these opposite direction gradients.

Inner Product Regularization From the perspective of using regularization methods when training the network to mitigate the catastrophic problem, we directly optimize the product term as a penalty with a hyperparameter λ to the continual learning process. We re-write the total loss to:

$$\mathcal{L} = \mathcal{L}_{ft}(\theta) + \mathcal{L}_{cl}(\theta) + \lambda max(0, -\eta\nabla_\theta\mathcal{L}_{ft}(\theta)^T\nabla_\theta\mathcal{L}_{cl}(\theta)). \tag{2}$$

In this way, we can add a penalty to the gradients that might hurt the upstream model behavior in the evolving loss to mitigate the catastrophic forgetting problem in the CME challenge.

Inner Product Restricted Optimization From perspective of editing the model directly without shifting the model through two conflict loss backwards, we introduce a new model optimizing strategy and only edit the model parameters with the same direction gradients.

Supposing that the calculated loss is $\mathcal{L} = \mathcal{L}_{ft}(\theta) + \mathcal{L}_{cl}(\theta)$, the gradients on parameter θ^i is $\nabla_{\theta^i}\mathcal{L}(\theta)$ and the inner product term on parameter θ^i is $\nabla_{\theta^i}\mathcal{L}_{ft}(\theta)^T\nabla_{\theta}\mathcal{L}_{cl}(\theta)$. Instead of following traditional optimization process $\theta^i = \theta^i - \eta\nabla_{\theta^i}\mathcal{L}(\theta)$, we optimize the gradients with positive inner-product restriction and fix the parameter where inner-product is negative:

$$\theta^i = \begin{cases} \theta^i - \eta \cdot \mathcal{I}(\theta^i) \cdot \nabla_{\theta^i}\mathcal{L}(\theta), & \mathcal{I}(\theta^i) < 0 \\ \theta^i & \mathcal{I}(\theta^i) \geq 0 \end{cases} \tag{3}$$

In this way, only the parameters with no gradient conflicts are edited therefore the model is less shifted and can maintain the upstream performances.

Compared with similar parameter editing strategies such as Editable Neural Networks Sinitsin et al. (2020), MEND Mitchell et al. (2021), the inner-product optimization process does not rely on a meta network or another pack of parameters to edit the model therefore can be used in the continual learning process. These learnable edit methods cannot be applied in solving the unknown and unlimited continual model evolving process. The inner-product term can guide the model-editing process unlike a whole-model updating with conflict loss backwards, which is a simple method as a headstart to solve the Continual Model Evolvement challenge.

4 EXPERIMENTS

4.1 BASELINES AND REFERENCE

We construct several strong baselines to explore the CME challenge. The lower bound of the CME challenge is a **FrozenUpstream** model. That is, the upstream model f_0 is not changed overtime (i.e., $f_t \equiv f_0$). A straightforward upper bound of the CME challenge is to consider that we collect all new-coming samples S as well as the upstream data D to train a model f_T offline and test the last episode. In practice, we use a subset of D.

We construct baselines using Continual Fine-Tuning, L2-Regularization, Experience Replay (ER) and Maximally Interfered Replay (MIR) methods. For the experience replay method (ER and MIR), we set replay frequency k to 1, that is, for each continual learning episode, we apply a round of replay.

4.2 DATASET CONSTRUCTION

In the Continual Model Evolvement challenge, we consider using the question answering task as the testbed. Specifically, we use the SQuAD v1.1 dataset Rajpurkar et al. (2016) as the upstream data D_0 and use the SQuAD v2.0 dataset Rajpurkar et al. (2018) and the HotPot QA dataset Yang et al. (2018) to construct the challenging input stream as the evolving challenge for the model continual learning process. We make a special modification to the HotPot QA dataset to construct unanswerable questions as the evolving task. In the SQuAD 2.0 task, the unanswerable questions are improper for the given context. Instead of constructing unanswerable questions as the evolving task, we construct unanswerable contexts in the HotPotQA dataset to construct a type of challenging sample. In the HotPotQA dataset, there are multiple paragraphs since it requires a reasoning process to answer the question. We randomly drop one or two key paragraphs in the given passage therefore the original question becomes unanswerable. Specifically, we use the development set with distractions of the official HotPotQA dataset. For the OOD input stream, following Lin et al. (2022), we introduce several question answering datasets including Natrual Questions (NQ) Kwiatkowski et al. (2019), SearchQA Dunn et al. (2017) and TriviaQA Trischler et al. (2017). We use these datasets based on the MRQA benchmark Fisch et al. (2019). Therefore, the new-coming data stream S_i is the combination of the challenging samples and OOD samples. For each time step, the datastream S_i includes samples with partition conditioned to timestep t and we set the total timestep $T = 100$.

Table 1: Results (%) in multiple metrics based on exact match in the generative question answering task: **EFR**=Error-Fixing Rate; **UKR**=Upstream Knowledge Retention; **E/OKR**= Evolving/OOD Knowledge Retention; **E/OKG**= Evolving/OOD Knowledge Generalization. Column names with bars are the average of all periods. The ones with '$^{(T)}$' are the scores at the final step. **Comb.** is the combination of L2 and IP regularization terms. k is the replay interval, c is the candidate pool size.

Methods ↓ Metrics →	$\overline{\text{EFR}}$	$\overline{\text{UKR}}$	$\overline{\text{EKR}}$	$\overline{\text{EKG}}$	$\overline{\text{OKR}}$	$\overline{\text{OKG}}$	$\text{UKR}^{(T)}$	$\text{EKR}^{(T)}$	$\text{EKG}^{(T)}$	$\text{OKR}^{(T)}$	$\text{OKG}^{(T)}$
Frozen ($f_t \equiv f_0$)	0.0	86.52	0.07	0.0	58.0	41.67	86.52	0.0	0.0	47.21	41.67
● Continual Fine-Tuning	98.85	63.0	82.12	52.89	81.93	45.51	60.16	90.26	69.44	80.17	49.04
■ Online L2-Reg.	98.0	59.64	88.99	58.6	79.78	46.14	43.16	96.1	77.78	69.27	45.67
▲ Exp. Replay (k=1)	95.03	76.22	93.42	48.77	88.78	51.06	75.98	96.1	61.46	85.2	53.45
✚ L2-Reg.	96.33	78.1	90.24	44.95	89.14	52.23	76.37	96.75	59.38	82.12	56.17
▶ MIR (k=1,c=256)	97.81	59.85	87.5	53.47	80.67	45.65	39.06	98.05	77.43	73.18	44.55
✚ L2-Reg.	97.45	63.74	85.55	52.24	80.09	47.0	44.92	97.4	76.39	70.11	45.43
Our methods ↓											
▲ ER ✚ IP-Reg. ($\eta = 1$)	91.99	80.12	85.4	35.15	89.69	52.46	80.47	92.21	48.26	88.83	56.57
▲ ER ✚ Comb. ($\eta = 1$)	88.19	81.05	89.22	40.82	87.89	51.96	81.25	90.91	51.39	86.03	54.57
▲ ER ✚ Comb. ($\eta = 0.1$)	89.16	82.23	86.36	35.88	88.44	52.31	81.25	94.16	48.61	84.92	55.05
▲ ER ✚ IP-Optim. ($\eta = 1$)	6.19	85.53	25.02	15.51	63.12	48.9	86.13	47.4	29.51	53.07	50.24
▲ ($\eta = 10$)	6.93	86.15	21.28	13.35	62.66	48.97	85.55	38.31	26.39	52.51	49.44
▲ ($\eta = 100$)	4.05	85.74	26.99	17.36	63.11	48.6	85.55	44.16	30.21	53.07	50.72
▶ MIR ✚ IP-Reg.($\eta = 1$)	94.94	79.51	78.72	33.95	86.3	50.19	81.25	88.31	51.74	77.37	53.37
▶ MIR ✚ Comb.($\eta = 1$)	96.53	80.3	82.5	37.42	85.19	51.33	81.84	92.86	53.47	75.42	52.24
▶ MIR ✚ IP-Optim.($\eta = 1$)	2.66	85.33	8.43	5.05	61.75	46.55	85.94	20.78	14.58	52.51	48.72
Offline Fine-Tuning	*97.22*	-	-	-	-	-	*81.25*	*99.33*	*53.47*	*90.91*	*56.49*

Similar to the continual model refinement data stream setup, in the continual data streams, the data stream S_t has K samples in total. These samples includes $K_u = K * \alpha^{t-1}$ samples from the upstream data, and $K_n = K - K_u$ new-coming samples including $K_c = K_n * \gamma$ samples that require model evolving and $K_n - K_c$ samples that are from OOD datasets. Here, α and γ are hyper-parameters for the dataset construction. We use $\alpha = 0.9$ and $\gamma = 0.5$ in all our experiments.

A practical detail is that we mix normal samples of the evolving datasets (answerable samples) in the incoming samples S which is more realistic. But during evaluation, we evaluate EKR and EKG over challenging samples only. That is, the challenging set S_t^c during evaluation are all unanswerable questions. In this way, the EKR and EKG results reflect how well the model evolved during the continual learning process.

4.3 Main Results

CL methods in CME We first explore how traditional continual methods behave in the CME challenge.

As seen in Table 1, when we do not tune the upstream model, the frozen upstream model f_0 cannot fix any incoming errors and cannot tackle either the evolving or the OOD task. The offline finetuning can achieve very high retention scores yet still struggle in generalizing the new-coming samples.

The most fundamental baseline continual fine-tuning, on the other hand, can successfully fix most incoming errors. While the continual fine-tuning method can successfully fix the incoming samples, it loses accuracy on the upstream samples, indicating that continual fine-tuning faces serious catastrophic forgetting problem, which makes it unsuitable for training real-world deployed models for evolving. Plus, though it can obtain some improvements in the evolving knowledge learning, the drop in the upstream task and the poor improvements in learning the OOD knowledge indicates that the continual fine-tuning overfits the evolving task.

As for the regularization-based method L2-regularized continual fine-tuning, we can observe that L2-regularization suffers in the continual evolving task, though it achieves high EKR and EKG scores. L2-regularization cannot maintain the upstream performances in the evolving process which is a different result compared with using L2-regularization in the model refinement task only. Therefore, we may conclude that L2-regularization cannot cope with two different task goals even though the evolving task is derived from the upstream task.

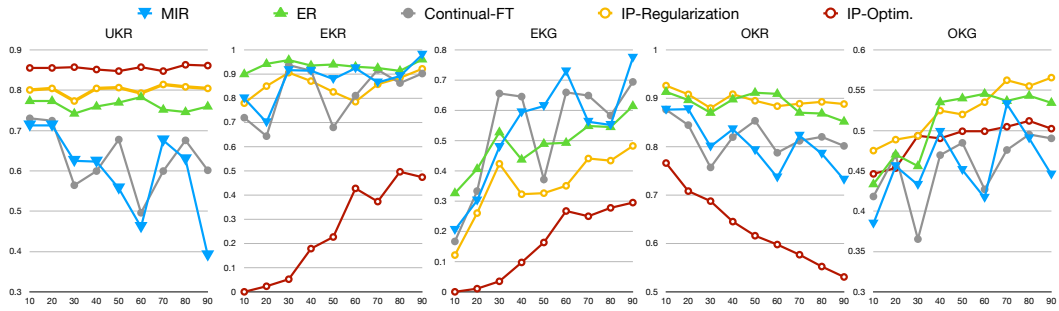

Figure 2: The curves of key metrics over time of CME methods. The x-axis is the time step.

Experience replay methods, which uses both upstream data and new-incoming samples during the continual process, can obtain a relatively better performances compared with the baselines above. As seen, experience replay method can obtain a significant improvement compared with the continual fine-tuning method in the upstream task performances. As for the MIR method, which is also a form of experience replay, the performance is not promising, that is, the upstream task performance is hurt seriously. In the MIR method, the replay samples are those with the largest losses, therefore, the replay samples will focus on the evolving task since the evolving samples are new to the upstream model. We can conclude that methods such as MIR might not be suitable for the CME challenge given the unique challenge of the evolving task.

Restricted Regularization While current continual learning methods cannot obtain promising results, the inner-product restricted regularization method can obtain a significant improvement: that is, though the evolving task performance is lower than unrestricted replay methods, the inner-product regularization method can maintain the upstream knowledge (the average UKR is 80.12 compared with 63.00 in the simple experience replay method) while it still can learn a considerable amount of evolving and OOD knowledge. Compared with previous regularization methods such as L2-regularization, we can conclude that inner-product regularization is effective in mitigating the catastrophic forgetting problem, which takes precedence in the CME challenge since we must maintain the upstream performance in a deployed model during the continual learning process. Also, we can observe that when we add the L2-regularization with the inner-product regularization, the performance is further improved, which indicates that L2-regularization method can be used as an auxiliary penalty term to constrain models during continual learning when the continual learning task gap are already narrowed by the inner-product constraint.

Restricted Optimization As mentioned, instead of regularization method, we can directly edit the optimizing process using the inner-product restriction. As seen, the restricted optimization method can maintain almost all the knowledge in the upstream task. As for the new-coming samples, a relatively smaller proportion of the evolving task samples can also be correctly recognized and a considerable amount of OOD samples can be correctly understood, indicating that the inner-product optimization can improve the model performances on the new-coming samples with no deterioration on the upstream task. However, the restricted optimization, unlike direct tuning methods, cannot properly fix the errors in the incoming samples, the learning process is different from fitting the incoming samples. When using different weight parameters η, the performance difference is not large, indicating that the inner-product restriction is stable when added to the continual learning process. To summarize, though error fixing and the evolving task learning ability is not ideal, inner-product optimization is a way to learn new tasks without catastrophic forgetting which is important in tuning deployed models.

Time Stream Variance In figure 2, we plot the metric changes during the continual learning process. We can observe that generally, CL methods can learn the new-coming knowledge overtime. In our IP-optimization method, though the knowledge retention seems to drop during learning, the generalization ability is improving, indicating that the optimization is effective especially when the UKR maintains a high-level.

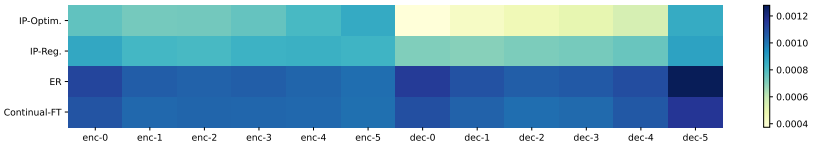

Figure 3: The parameter variance between f_0 and continual learned model f_T.

Parameter Shift Comparison To explore the parameter shift in the continual learning process, we calculate the parameter weight change of different methods between the upstream model f_0 and the last-episode model f_T. We calculate the average absolute deviation between each parameter between f_0 and f_T of each different layers of the model.

As seen in Figure. 3, the inner-product based methods can maintain the model less perturbed, which directs to better upstream performances. As seen, the decoder parameters are less changed in the IP-Optimization method, which is because the contradicted gradients tend to exist in the decoder layers, causing a serious catastrophic forgetting problem.

5 Related Work

Continual Learning in NLP With wide deployments of pre-trained model applications, continual learning of NLP applications are drawing more and more attention Biesialska et al. (2020); Sun et al. (2020); Wang et al. (2019); Huang et al. (2021); Jin et al. (2021). Most of these works still use limited experiment setup without considering the real-world application scenario needs. That is, their setups are focusing on no-revisiting upstream data, incremental tasks or limited range of new-coming data. Recent work Lin et al. (2022) introduces a boundary-agnostic continual learning framework that aims to find a more realistic scenario in NLP applications, while it is also limited to solving out-of-distribution data. Compared to all these continual learning frameworks and methods, our proposed CME challenge introduces a high-level evolving task and provides a realistic testbed using the unanswerable questions as an example, which is a challenging yet very realistic setup in real-world NLP applications.

Model Refinement and Editing Recent trends of refining or editing pre-trained models in NLP have brought many works to the community. A major trend is to explore how to edit a model to adjust factual knowledge in pre-trained language models Jang et al. (2021). Generally, model editing Sinitsin et al. (2020); Mitchell et al. (2021) usually focuses on designing a meta-network to assist the model parameter editing which cannot be used in the continual learning paradigm. Time-sensitive knowledge edit Zhu et al. (2020); De Cao et al. (2021) usually focuses on factual knowledge in pre-trained models which is different from our CME setup that focuses on deployed models for a certain downstream task.

6 Conclusion and Future Directions

In this paper, we propose a new challenge in the NLP application deployment, Continual Model Evolvement. We aim to explore a real challenge that fine-tuned models would face during their usage in deployments. We modify the unanswerable questions as an *evolving task* for the continual learning setup and find that previous continual learning methods suffer from catastrophic forgetting problem which cannot be tolerated in real-world applications. We further propose a simple yet effective inner-product restriction algorithm to significantly mitigate the catastrophic problem while there is still space for a better continual learning method to cope with both catastrophic forgetting problem and the learning of new challenging tasks. We are hoping that in the future, the proposed CME challenge and the proposed headstart method can be used in assisting real-world NLP applications and future work will be expansions of the CME challenge in different types of applications and more effective methods to solve such a challenge.

REFERENCES

Rahaf Aljundi, Lucas Caccia, Eugene Belilovsky, Massimo Caccia, Min Lin, Laurent Charlin, and Tinne Tuytelaars. Online continual learning with maximally interfered retrieval. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2019. Curran Associates Inc. URL https://dl.acm.org/doi/abs/10.5555/3454287.3455350.

Magdalena Biesialska, Katarzyna Biesialska, and Marta R. Costa-jussà. Continual lifelong learning in natural language processing: A survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6523–6541, Barcelona, Spain (Online), 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.574. URL https://aclanthology.org/2020.coling-main.574.

Nicola De Cao, Wilker Aziz, and Ivan Titov. Editing factual knowledge in language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 6491–6506, Online and Punta Cana, Dominican Republic, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.522. URL https://aclanthology.org/2021.emnlp-main.522.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

Matthew Dunn, Levent Sagun, Mike Higgins, V. Ugur Güney, Volkan Cirik, and Kyunghyun Cho. Searchqa: A new q&a dataset augmented with context from a search engine. *ArXiv preprint*, abs/1704.05179, 2017. URL https://arxiv.org/abs/1704.05179.

Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. MRQA 2019 shared task: Evaluating generalization in reading comprehension. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pp. 1–13, Hong Kong, China, 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-5801. URL https://aclanthology.org/D19-5801.

Yufan Huang, Yanzhe Zhang, Jiaao Chen, Xuezhi Wang, and Diyi Yang. 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*, pp. 2736–2746, Online, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.218. URL https://aclanthology.org/2021.naacl-main.218.

Joel Jang, Seonghyeon Ye, Sohee Yang, Joongbo Shin, Janghoon Han, Gyeonghun Kim, Stanley Jungkyu Choi, and Minjoon Seo. Towards continual knowledge learning of language models. *ArXiv preprint*, abs/2110.03215, 2021. URL https://arxiv.org/abs/2110.03215.

Xisen Jin, Bill Yuchen Lin, Mohammad Rostami, and Xiang Ren. Learn continually, generalize rapidly: Lifelong knowledge accumulation for few-shot learning. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 714–729, Punta Cana, Dominican Republic, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.62. URL https://aclanthology.org/2021.findings-emnlp.62.

James Kirkpatrick, Razvan Pascanu, Neil C. Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114:3521 – 3526, 2017.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion

Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. doi: 10.1162/tacl_a_00276. URL https://aclanthology.org/Q19-1026.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7871–7880, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.703. URL https://aclanthology.org/2020.acl-main.703.

Bill Yuchen Lin, Sida I. Wang, Xi Victoria Lin, Robin Jia, Lin Xiao, Xiang Ren, and Wen tau Yih. On continual model refinement in out-of-distribution data streams. In *ACL*, 2022.

Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pp. 109–165. Elsevier, 1989.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D. Manning. Fast model editing at scale. *ArXiv preprint*, abs/2110.11309, 2021. URL https://arxiv.org/abs/2110.11309.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2383–2392, Austin, Texas, 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1264. URL https://aclanthology.org/D16-1264.

Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for squad. In *ACL*, 2018.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy P. Lillicrap, and Gregory Wayne. Experience replay for continual learning. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 348–358, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/fa7cdfad1a5aaf8370ebeda47a1ff1c3-Abstract.html.

Anton Sinitsin, Vsevolod Plokhotnyuk, Dmitriy Pyrkin, Sergei Popov, and Artem Babenko. Editable neural networks. *ArXiv*, abs/2004.00345, 2020.

Fan-Keng Sun, Cheng-Hao Ho, and Hung-Yi Lee. LAMOL: language modeling for lifelong language learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=Skgxcn4YDS.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. NewsQA: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pp. 191–200, Vancouver, Canada, 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-2623. URL https://aclanthology.org/W17-2623.

Hong Wang, Wenhan Xiong, Mo Yu, Xiaoxiao Guo, Shiyu Chang, and William Yang Wang. Sentence embedding alignment for lifelong relation extraction. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 796–806, Minneapolis, Minnesota, 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1086. URL https://aclanthology.org/N19-1086.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6. URL https://aclanthology.org/2020.emnlp-demos.6.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, Brussels, Belgium, 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1259. URL https://aclanthology.org/D18-1259.

Chen Zhu, Ankit Singh Rawat, Manzil Zaheer, Srinadh Bhojanapalli, Daliang Li, Felix Yu, and Sanjiv Kumar. Modifying memories in transformer models. *ArXiv preprint*, abs/2012.00363, 2020. URL https://arxiv.org/abs/2012.00363.

APPENDIX

IMPLEMENTATIONS

Dataset Details The datasets we use are open-source question answering datasets including SQuAD v1.1, SQuAD v2.0, HotPotQA, SearchQA, Natrual Questions and TriviaQA.

We set the incoming sample K to 64, and set $a\alpha = 0.9$ and $\gamma = 0.5$ and the total timestep $T = 100$ to construct the CME challenge. We run 5 different runs to generate different streams and test the performances accordingly and use the average results. We select these parameters based on a practical prediction and we also conduct some ablation experiments with different K, α, γ which show similar performances.

Upstream Model Details We implement all our methods based on the huggingface transformers Wolf et al. (2020). Specifically, we follow Lin et al. (2022) and implement the question answering task as a text generation task.

We train the upstream model based on the BART-Base model. Such a sequence-to-sequence model is a flexible structure that can be used in various NLP applications. In detail, we concat the question, the contexts (with multiple paragraphs in datasets such as HotPotQA dataset.) and the output is the answer to the question.

We run the upstream model finetuning with batchsize 64 and learning rate 5e-5 and the running epoch is 30. The fine-tuning process is meant to obtain a close score compared with extractive question answering systems. The upstream model achieves an average exact match score of 86.52 on subsets of the SQuAD v1.1 dataset which is similar to the BERT-base performances.

Continual Learning Methods Details In the continual fine-tuning process, we tune different learning rates and running epochs. We search over learning rates in $\{1e-5, 2e-5, 5e-5\}$ and epochs in $\{5, 10, 20\}$ and set learning rate to $2e-5$ and epoch to 20 while these hyper-parameters do not affect the performances by a large margin.

In the L2-regularization method, we set the penalty weight to 1 for all experiments. In the replay methods, we set the replay frequency to 1 for all experiments considering that the inner-product method is based on a continuous replay process, therefore when the frequency $k = 1$, the results are fairly compared.

Runtime Analysis we use 4 Nvidia 3090 GPUs for all experiments. The continual fine-tuning process is a standard pre-train model fine-tuning process. Compared with continual fine-tuning, replay methods require additional computational cost and methods such as MIR requires additional computation (with virtual model update and ranking over the memorized samples).

For the inner-product restriction method, the process needs to maintain the gradients on the previous timestep therefore the GPU memory cost is larger than continual fine-tuning.

Unsatisfied Attempts During the CME challenge method design, we also consider utilizing the unique ability such as prompt learning to assist the evolving task learning. That is, we add a decoder-end prefix to the answer: we set a pattern that generates outputs such as *something is the answer*; when the task evolves, the model should generate patterns such as *something is **not** the answer*. We assume that in this way, the previous learned answer generation pattern can easily adapt to the new task which is recognizing whether the question is answerable. However, such an attempt only achieves similar performance compared with not using any pre-defined patterns, indicating that the evolving task is quite challenging that such simple guidance is not useful.