Rational LAMOL: A Rationale-Based Lifelong Learning Framework

Kasidis Kanwatchara[1*], **Thanapapas Horsuwan**[1*], **Piyawat Lertvittayakumjorn**[2],
Boonserm Kijsirikul[1], **Peerapon Vateekul**[1†]
[1] Department of Computer Engineering, Faculty of Engineering,
Chulalongkorn University, Thailand
[2] Department of Computing, Imperial College London, UK
{kanwatchara.k, thanapapas.h}@gmail.com, pl1515@imperial.ac.uk,
{boonserm.k, peerapon.v}@chula.ac.th

Abstract

Lifelong learning (LL) aims to train a neural network on a stream of tasks while retaining knowledge from previous tasks. However, many prior attempts in NLP still suffer from the catastrophic forgetting issue, where the model completely forgets what it just learned in the previous tasks. In this paper, we introduce Rational LAMOL, a novel end-to-end LL framework for language models. In order to alleviate catastrophic forgetting, Rational LAMOL enhances LAMOL, a recent LL model, by applying critical freezing guided by human rationales. When the human rationales are not available, we propose exploiting unsupervised generated rationales as substitutions. In the experiment, we tested Rational LAMOL on permutations of three datasets from the ERASER benchmark. The results show that our proposed framework outperformed vanilla LAMOL on most permutations. Furthermore, unsupervised rationale generation was able to consistently improve the overall LL performance from the baseline without relying on human-annotated rationales. We made our code publicly available at `https://github.com/kanwatchara-k/r_lamol`.

1 Introduction

The grounds of lifelong learning (LL) stem from the ability of humans to continually acquire, consolidate, and transfer knowledge and skills throughout their lifespan. This ability is also important for real-world natural language processing (NLP) applications, where autonomous agents are required to interact with users from various domains through continuous streams of information and language semantic drifts occur over time. The existing dominant paradigm for machine learning, however, is isolated learning (Chen and Liu, 2016). While isolated learning has shown some successes in a variety of domains, their applicability remains limited to the assumption that all samples are available during the learning phase. When a stream of tasks are trained sequentially, machine learning and neural network models face catastrophic forgetting or interference (McCloskey and Cohen, 1989). This occurs due to the non-stationary data distribution that biases the model.

We focus on lifelong language learning (LLL), which is lifelong learning on a stream of NLP tasks. To the best of our knowledge, the grounds of LLL are left largely underexplored. LAMOL is an LLL general framework that has garnered recent interest due to its simplicity (Sun et al., 2020). In particular, LAMOL transforms all NLP tasks into the question answering (QA) format according to McCann et al. (2018) and generates pseudo-samples of old tasks using its language modeling (LM) capability to refresh the learned knowledge. However, there is still a gap between the performance of LAMOL and the result of multi-task learning which is generally considered as the upper bound of LLL performance. This indicates that only pseudo-samples generation may not be sufficient to prevent catastrophic forgetting.

In this paper, we improve existing LLL strategies by proposing Rational LAMOL, a rationale-based lifelong learning framework which equips the original LAMOL with critical freezing (Nguyen et al., 2020) to further prevent catastrophic forgetting. Particularly, we devise an algorithm to identify critical components in transformer-based language models using rationales, and the selected compo-

[*] Equal contributions
[†] Corresponding author

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics
and the 11th International Joint Conference on Natural Language Processing, pages 2942–2953
August 1–6, 2021. ©2021 Association for Computational Linguistics

nents will be frozen to maintain learned knowledge while being trained on a new task.

The contributions of our paper are listed below:

- We demonstrate the importance of freezing plastic components (i.e., components that are most susceptible to change) in transformer-based models to strengthen memories of the previously learned tasks in the LLL setting.

- We propose *critical component identification* algorithm which analyzes the transformer-based LLL model with rationales so as to find the most plastic component to freeze. This step is so called critical freezing, firstly devised in computer vision (Nguyen et al., 2020) but we adapted it to NLP.

- We propose that unsupervised generated rationales by InvRat (Chang et al., 2020) can be effectively used as substitutions of human rationales, allowing our framework to be applied to generic NLP datasets.

We evaluated Rational LAMOL on six task order permutations of three datasets from the ERASER benchmark (DeYoung et al., 2020). The results show that our proposed framework outperformed the original LAMOL on five out of the six permutations, achieving average improvements of 1.83% with a lower standard deviation of 4.57%. Moreover, using unsupervised rationale generation instead of human rationales also yielded competitive performance, achieving average improvements of 2.67% from original LAMOL.

2 Background and Related Work

In this section, we briefly introduce the concept of lifelong learning, catastrophic forgetting, and component freezing which are relevant to the core idea of Rational LAMOL. We also briefly summarize prominent researches related to rationales.

Lifelong Learning and Catastrophic Forgetting While people fine tune a pre-trained model to perform a single task, lifelong learning (LL) is a setting in which a learner performs sequential learning of infinitely incoming tasks $\tau = \{\tau_1, \tau_2, ..., \tau_i, ..., \}$, where τ_i is the i-th task to learn at a particular point in time. The objective of the LL learner is to ideally both optimize the performance on the new task and maintain optimal performance on previous tasks τ_t for $t = 0, 1, ..., i$. Moreover, the ability to transfer knowledge across different tasks is also desired. However, naively training on a sequence of tasks without accounting for the difference in data distributions would result in an abrupt decrease in old tasks performance. This phenomenon is known as *Catastrophic Forgetting* (McCloskey and Cohen, 1989). There are multiple existing works that aim to mitigate catastrophic forgetting in LL. They can be categorized into three major approaches. First, regularization methods use a regularization term to constrain changes when updating weights in a new task (Kirkpatrick et al., 2017; Aljundi et al., 2017; Lee et al., 2017). Second, data-based methods disallow significant deviation of weights from previous tasks by keeping a small subset of data from the previous tasks or generating pseudo-data to refresh the learned knowledge (Lopez-Paz and Ranzato, 2017; Chaudhry et al., 2019; de Masson d'Autume et al., 2019; Li and Hoiem, 2018). Third, architecture-based methods dynamically transform the neural network architectures in order to accommodate new knowledge (Rusu et al., 2016; Chen et al., 2016).

Lifelong Language Learning or LLL is a scenario where a model sequentially learns from a stream of NLP tasks in an LL manner. To the best of our knowledge, LLL has rarely been studied and previous works usually target a single type of NLP tasks (Chen et al., 2015; Liu et al., 2019; de Masson d'Autume et al., 2019). To go beyond this limitation, Sun et al. (2020) proposed LAMOL, a learning framework that utilizes a language model to simultaneously predict outputs and learn to generate pseudo-training examples, which are exploited to alleviate catastrophic forgetting. Hence, LAMOL, as well as our Rational LAMOL, naturally falls into the data-based LL approach since data from previous tasks, albeit generated, is utilized to constrain a model.

Component Freezing While component freezing is also a common practice in the fine-tuning process, it is done to prevent loss in general knowledge in lower layers of the model (Raganato and Tiedemann, 2018).

By contrast, many architecture-based LL methods, for example Rusu et al. (2016), utilize component freezing to prevent changes to learned knowledge from previous tasks and enlarge the model to accommodate new tasks, thereby making the model immune to forgetting. Our Rational LAMOL also uses component freezing, but unlike architecture-based methods, only a small part of the model is frozen and its size is constant throughout the learning process.

Rationales Rationales are reasons for labels or predictions. In NLP, they are usually parts of the input texts which support or contribute to the class labels. Rationales could be either annotated by humans or generated by machine learning models. Human rationales have been used to enhance machine learning in multiple studies. For instance, Rajani et al. (2019) used the rationales to guide a neural network toward better reasoning. Bao et al. (2018) utilized rationales as auxiliary information to train a neural network model, reducing training examples required to achieve good results. Recently, DeYoung et al. (2020) introduced the ERASER benchmark consisting of multiple datasets, all of which are annotated with human rationales. This facilitates the advancement of research on interpretable NLP. In the experiment, we used human rationales from ERASER in the critical component identification step to find the most plastic component to be frozen.

Meanwhile, some researchers attempt to design architectures to predict rationales from labelled data. Existing rationalization techniques commonly use the maximum mutual information (MMI) criterion to select rationales, which is prone to choosing spurious correlation between input features and outputs as rationales (Lei et al., 2016; Yu et al., 2019). To fix this issue, Invariant Rationalization (InvRat) (Chang et al., 2020) follows the invariant risk minimization (IRM) paradigm, as introduced by Arjovsky et al. (2019). It utilizes the *environment* variable to isolate and select the causal features that faithfully explain the output. In order to allow Rational LAMOL to be applied to any NLP dataset, we choose to leverage InvRat to automatically produce rationales due to its superior performance and straightforward application, removing the need for human rationales.

3 Methodology

We introduce Rational LAMOL and its detailed implementation in this section. As Rational LAMOL is based from LAMOL (Sun et al., 2020), we briefly explain LAMOL in Section 3.1. Then we introduce the core lifelong learning framework of Rational LAMOL in Section 3.2. This is followed by two proposed enhancements including *critical component identification* and *unsupervised rationale generation*, detailed in Section 3.3 and 3.4, respectively.

3.1 LAMOL

Language Modeling for Lifelong Language Learning (LAMOL) (Sun et al., 2020) utilizes a single language model (LM) as a multipurpose model. Framing all tasks as question answering (QA), the LM now poses as a generic task-agnostic model. In addition, LAMOL trains the LM as a generative model upon receiving a special generation token. Using a single model for both providing answers and generating pseudo-samples, LAMOL truly exhibits a model of LM and QA duality.

The benefit that comes with the generative part of the model tackles the long-standing issue of LL– catastrophic forgetting. While other methods make use of extra memory or model capacity to preserve a subset of real samples (Lopez-Paz and Ranzato, 2017; Chaudhry et al., 2019) or to accomodate a separate generator (Shin et al., 2017; Kemker and Kanan, 2017), LAMOL transfers all the responsibilities into a single model. It learns the ability to select potentially prominent features befitting learning by modeling the input. This allows the model to replay meaningful pseudo-samples from previous tasks while forcing the model to memorize knowledge acquired from previous tasks tied to the generation token. In this paper, we propose exploiting rationales with LAMOL to further improve the LLL performance, discussed next.

3.2 Rational LAMOL

Rational LAMOL, illustrated in Figure 1 (right), is a learning framework revolving around the original methodologies of LAMOL. We consider an LL setting where $\tau = \{\tau_1, \tau_2, ..., \tau_i, ...\}$ is a stream of learning tasks and τ_i is the i-th task to train at a particular point in time. Let M_i denote the model M after being trained for task i, where M_0 is the initialized pre-trained model. Using these notations and starting from M_0, Rational LAMOL works iteratively in four steps as follows. First, given a model M_i, it trains M_i with the task τ_{i+1} using LAMOL's training procedure to obtain $\hat{M}_{i+1}$. Second, for $i > 0$, it applies critical component identification, which is described in Section 3.3, on M_i and $\hat{M}_{i+1}$ with the rationales of task τ_i to dissect the most plastic layers or blocks. Third, we take a step back to work at M_i and apply critical freezing, i.e., freezing the most plastic components,

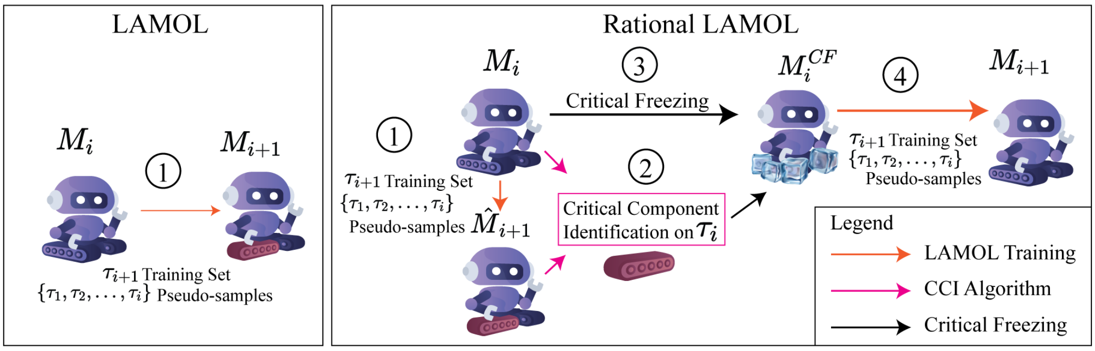
LAMOL
M_i
①
M_{i+1}
$\mathcal{T}_{i+1}$ Training Set
$\{\tau_1, \tau_2, \ldots, \tau_i\}$ Pseudo-samples
Rational LAMOL
M_i
③
Critical Freezing
M_i^{CF}
④
①
$\mathcal{T}_{i+1}$ Training Set
$\{\tau_1, \tau_2, \ldots, \tau_i\}$
Pseudo-samples $\hat{M}_{i+1}$
②
Critical Component Identification on $\mathcal{T}_i$
M_{i+1}
$\mathcal{T}_{i+1}$ Training Set
$\{\tau_1, \tau_2, \ldots, \tau_i\}$
Pseudo-samples
Legend
LAMOL Training
CCI Algorithm
Critical Freezing

Figure 1: **Left**: The overview of LAMOL. **Right**: The overview of Rational LAMOL, our proposed framework that aims to alleviate catastrophic forgetting by freezing the critical component.

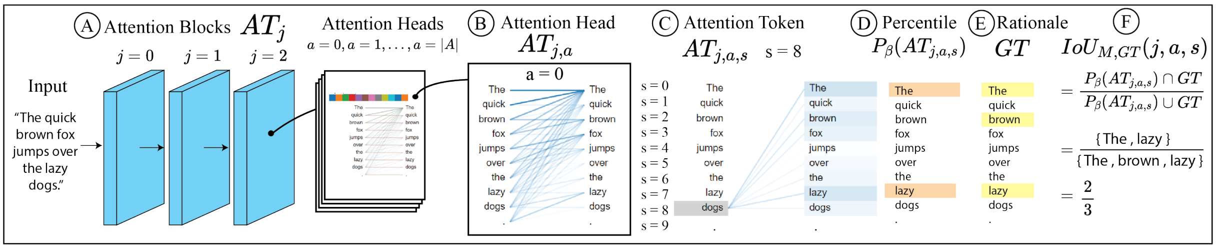
Ⓐ Attention Blocks AT_j
$j = 0$ $j = 1$ $j = 2$
Input
"The quick brown fox jumps over the lazy dogs."
Attention Heads
$a = 0, a = 1, \ldots, a = |A|$
Ⓑ Attention Head
$AT_{j,a}$
a = 0
Ⓒ Attention Token
$AT_{j,a,s}$ s = 8
s = 0 s = 1 s = 2 s = 3 s = 4 s = 5 s = 6 s = 7 s = 8 s = 9
The quick brown fox jumps over the lazy dogs
Ⓓ Percentile
$P_\beta(AT_{j,a,s})$
Ⓔ Rationale
GT
Ⓕ
$IoU_{M,GT}(j, a, s)$
$= \frac{P_\beta(AT_{j,a,s}) \cap GT}{P_\beta(AT_{j,a,s}) \cup GT}$
$= \frac{\{\text{The , lazy}\}}{\{\text{The , brown , lazy}\}}$
$= \frac{2}{3}$

Figure 2: Schematic illustration of the calculation of $IoU_{M,GT}$. **A**: The input is fed through each attention block AT_j, where each block j has multiple heads. **B**: A single attention head $AT_{j,a}$ consists of the attention of the sequence in relation to all other tokens, as shown in **C**. Finally, the IoU calculation **F** is applied on the hard selection of attention token with percentiles **D** and the rationale ground truth in **E**.

to obtain M_i^{CF}. Lastly, we train M_i^{CF} through the task τ_{i+1} again to get a new model M_{i+1} that retains the most plastic memories. Note that despite the unique nature of LAMOL, our Rational LAMOL does not limit its usage to a single model architecture. It has potential applications to general attention-based models suffering from catastrophic forgetting through domain shifts across tasks.

3.3 Critical Component Identification (CCI)

We propose the Critical Component Identification (CCI) algorithm, pointing out the most plastic block of our transformer-based LL model before moving on to a new task completely. (This shares the same spirit as Nguyen et al. (2020), proposing Auto DeepVis to find the most plastic blocks of CNN models for image classification.) The chosen block is the one that forgets what it has learned from the recent task the most when being introduced a new task, so we will freeze the block to prevent catastrophic forgetting in Rational LAMOL.

As shown in Algorithm 1, for each validation sample $x \in X$ of task i, the CCI compares the attention maps AT produced by the model M_i (i.e., the old model M_O in Algorithm 1) and $\hat{M}_{i+1}$ (i.e., the new model M_N in Algorithm 1) to find the most plastic block b with respect to this sample. Then it returns the block $\mathbb{F}$ which is the mode of all b, voted by most of the samples in X. Note that most of the variable names are preserved similar to Nguyen et al. (2020) for ease of reference, and some sections are refactored for readability.

In particular, to find b for the sample x, we iterate over all blocks $j = 1, ..., K$ and perform two steps. First, we find the representative map of the block j in M_O with respect to the ground truth GT (i.e., $RM_{M_O,GT}(j)$) by selecting the attention map of the attention head a^* and the token s^* in x from the block j that is most similar to the human rationale for the sample x (i.e., ground truth GT in Algorithm 1). Although interpretable NLP stands to be a nascent subfield for exploration (DeYoung et al., 2020), elementary visualization of attentions are possible in Transformers (Vig, 2019; Hoover et al., 2020). These self-attention mechanisms associate distant positions of a single sequence and many appear to exhibit behavior related to the sentences' syntactic and semantic structure (Vaswani et al., 2017). We hypothesize that the semantic nature of the self-attention mechanisms would opt for tokens most relating to positive evidence vital for predictions, being analogous to *rationales*–snippets that

Algorithm 1 Critical Component Identification

```
Input: Validation set $X$, ground truth rationale $GT$, old model $M_O$, new model $M_N$, number of blocks $K$
Output: Critical block $\mathbb{F}$
  Ł← $\emptyset$
  for all validation sample $x \in X$ do:
      IoUs $\leftarrow \emptyset$
      $AT_O, AT_N \leftarrow [M_O(x), M_N(x)]$
      for $j = 1, K$ do:
          $RM_{M_O,GT} \leftarrow$
              $AT_{j,a^*,s^*}$ with highest $IoU_{M_O,GT}$
          $RM_{M_N,M_O} \leftarrow$
              $AT_{j,a^*,s^*}$ with highest $IoU_{M_N,M_O}$
          APPEND(IoUs, max($IoU_{M_N,M_O}$))
      end for
      $b \leftarrow \arg\min_j$ IoUs
      APPEND(Ł, b)
  end for
  $\mathbb{F}$ = MODE(Ł)
  return $\mathbb{F}$
```

support outputs. To compute the similarity between attention maps and human rationales, we use Intersection over Union (IoU). Formally, the following equations explain this step.

$$RM_{M,GT}(j) = AT_{j,a^*,s^*} \tag{1}$$

where

$$(a^*, s^*) = \underset{a \in A, s \in S}{\arg\max}(IoU_{M,GT}(j,a,s)) \tag{2}$$

and

$$IoU_{M,GT}(j,a,s) = \frac{P_\beta(AT_{j,a,s}) \cap GT}{P_\beta(AT_{j,a,s}) \cup GT} \tag{3}$$

A is the set of all attention heads in the block, and S is the set of all tokens in x. $IoU_{M,GT}(j,a,s)$ reflects the similarity between the ground truth and the attention map of the block j, head a, and token s in x. Since the ground truth contains binary labels indicating whether a token is a part of the rationale or not, we need to convert the attention map $AT_{j,a,s}$ into binary labels using P_β – a simple binary thresholding which returns 1 for the value greater than the β-th percentile on the entire sequence (otherwise, 0). This is required as IoU works for comparing two binary masks. Figure 2 visualizes how to compute the IoU score by drilling down each component of the model.

After we obtain $RM_{M_O,GT}(j)$ of the block j, the second step finds the representative map of the block j in M_N with respect to M_O (i.e., $RM_{M_N,M_O}(j)$). This can be done by replacing M and GT in Equation 1-3 by M_N and M_O, respectively, and replacing GT on the right side of Equation 3 to be $P_\beta(RM_{M_O,GT}(j))$. After that, we collect the maximum IoU_{M_N,M_O} of the block j which represents the amount of knowledge of task i held in the model after we introduce task $i+1$. Therefore, the most plastic block b for this sample x is the block with the lowest maximum IoU_{M_N,M_O}.

Actually, transformer blocks are not the finest granularity that we could freeze. Since each block contains several attention heads, it is possible to freeze some attention heads individually. Hence, we propose another algorithm, applying to heads. This is similar to Algorithm 1, but instead of searching for blocks with lowest maximum IoU, the algorithm searches using both the attention blocks and attention heads together as keys. Although the definition of IoU stays the same, the definition of the representative map will be at a higher granularity. Formally, for a block index j and attention head a, $RM_{M,GT}$ will be computed as:

$$RM_{M,GT}(j,a) = AT_{j,a,s^*} \tag{4}$$

where

$$(s^*) = \underset{s \in S}{\arg\max}(IoU_{M,GT}(j,a,s)) \tag{5}$$

and we can freeze top n heads that receives most votes from the samples in the validation set X.

3.4 Unsupervised Rationale Generation

As described in Section 3.2, our framework requires rationales as an input. However, most existing NLP datasets are not annotated with rationales. To overcome the limitation, we leverage a recent unsupervised rationale generation framework, InvRat (Chang et al., 2020) to generate rationales as substitutions. Originally, InvRat was designed for single-input tasks such as sentiment analysis. However, since some of the datasets we experimented with are text-pair classification, we append the query (or question) at the end of each sample to accommodate these tasks.

Dataset	# Train	# Val	# Test	Metric
BoolQ	6,363	1,491	2,817	EM
Movie	1,600	200	200	
SciFact	405	100	188	

Table 1: Summary of datasets, dataset sizes, and their corresponding metrics. EM represents an exact match between texts.

4 Experimental Setup

4.1 Datasets

To evaluate our proposed framework, we conducted an experiment on three English text classification datasets, curated and made publicly available by ERASER[1] (DeYoung et al., 2020). All of the three datasets, as listed below, are provided with rationales marked by humans. Table 1 contains a summary of the datasets, dataset sizes, and metrics.

- BoolQ (Clark et al., 2019): a dataset comprises selected passages from Wikipedia and naturally occurring yes/no questions to be answered by the model.

- Movie Reviews (Zaidan and Eisner, 2008): a dataset composed of movie reviews. It contains positive and negative sentiment labels to be predicted by the model.

- SciFact (Wadden et al., 2020): a dataset containing expert-written scientific claims coupled with evidence-containing abstracts. Given a claim, the model has to identify if the abstract supports or refutes the claim.

We ran our proposed framework on all six permutations of task order for three times with different random seeds. The average results are then reported in Section 5.

4.2 Implementation Details

We followed the best LAMOL configuration from Sun et al. (2020). All parameters were kept at the default values. For all methods, we use the small GPT-2 model (Radford et al., 2019) as the language model. Each task was trained for five epochs. We applied greedy decoding during inference. Due to fine-tuning instability of neural network, in each task order, we used the same first task model M_1 for all methods in each run for fair comparison.

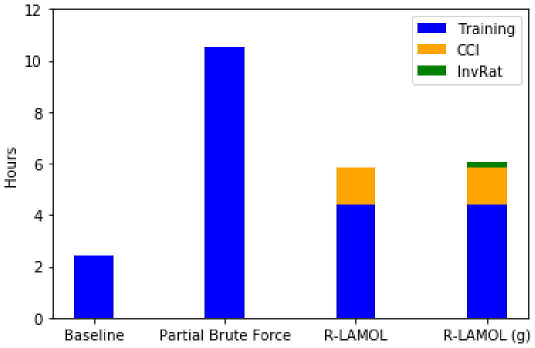

Figure 3: Average runtime in hours of various methods. R-LAMOL, R-LAMOL (g), and Partial Brute Force refer to Rational LAMOL, Generated Rational LAMOL, and Partial Brute Force $_{\text{block}}$ respectively.

Critical freezing was applied to a model with two different levels of granularity: block level and head level. The validation set of each task was used as input to Algorithm 1. For block level granularity, we chose to freeze the most frequent block obtained from the algorithm, while for head level granularity, 12 heads chosen returned by the algorithm were kept frozen during training. We used $\beta = 80$, i.e., selecting the top 20 percentile of attention scores to compare with ground truth rationales. As the ERASER benchmark has an average ratio of rationale tokens to document tokens of around 9.4%, we allowed rationale selection to be two times the average ratio (i.e., 20%).

For InvRat, we opted for 300-dimensional GloVe embeddings (Pennington et al., 2014). The generator and the predictor modules of InvRat were based on 1-layer bidirectional gated recurrent units (Chung et al., 2014) with 256 hidden units as in Chang et al. (2020). Maximum model input was set to 1,024 tokens. All hyperparameters for each task were tuned on the validation set.

5 Results and Discussion

This section reports the performance of Rationale LAMOL and compares it with LAMOL as the baseline as well as multitask learning, which is considered as the upper bound of LL. We also analyze the effect of each component in the proposed framework.

5.1 Effect of Component Freezing

In order to validate if component freezing truly helps reduce catastrophic forgetting, we performed partial brute force block-level freezing on each task permutation to approximately determine the upper

[1] https://www.eraserbenchmark.com/

Methods	BMS	BSM	MBS	MSB	SBM	SMB	Average	Std.
LAMOL	57.39	55.98	65.89	66.71	67.63	60.08	62.28	5.09
Partial Brute Force $_{\text{block}}$	62.97	64.05	66.73	67.75	65.22	69.05	65.96	2.30
Rational LAMOL $_{\text{block}}$	62.49	59.55	66.09	68.04	68.55	59.94	64.11	4.57
Rational LAMOL $_{\text{head}}$	64.35	61.70	65.22	67.76	56.59	60.62	62.71	3.93
Gen-Rational LAMOL $_{\text{block}}$	66.82	59.97	66.38	65.11	66.94	64.49	64.95	2.63
Gen-Rational LAMOL $_{\text{head}}$	67.35	57.36	66.51	63.85	63.98	65.52	64.10	3.57
Multitask	67.32							

Table 2: Accuracy of different methods evaluated on the models at the last epoch of the last task, averaged over three seeds. Each column refers to the order of tasks on which the methods were trained. B, M and, S refer to BoolQ, Movie Reviews, and SciFact, respectively. The Average and Std columns respectively are the average and standard deviation of the accuracy scores for each row of the methods.

bound of our Rationale LAMOL $_{\text{block}}$. Due to limited computing resources, we compromised with searching for all even-numbered block indices, and choosing the model with maximum average score of the first two tasks to do the brute force on the latter two tasks. Since brute force was performed on a per-task basis, our search space would be 6+6, the first six being the six blocks on the first two tasks, and the latter six being the six blocks on the last two tasks. Do note that true brute force would be 12×12. Although it is possible that our partial brute force is sub-optimal, we find that it is a good compromise due to limited computing resources. The results are presented in Table 2. Brute force was able to outperform vanilla LAMOL by a substantial margin of 3.68%, only 1.36% from the multitask upper bound. This suggests that component freezing is able to further nullify the effect of catastrophic forgetting from LAMOL. It also achieved a standard deviation of only 2.3% compared with LAMOL's 5.28%. This suggests that freezing the right component helps with task order resilience.

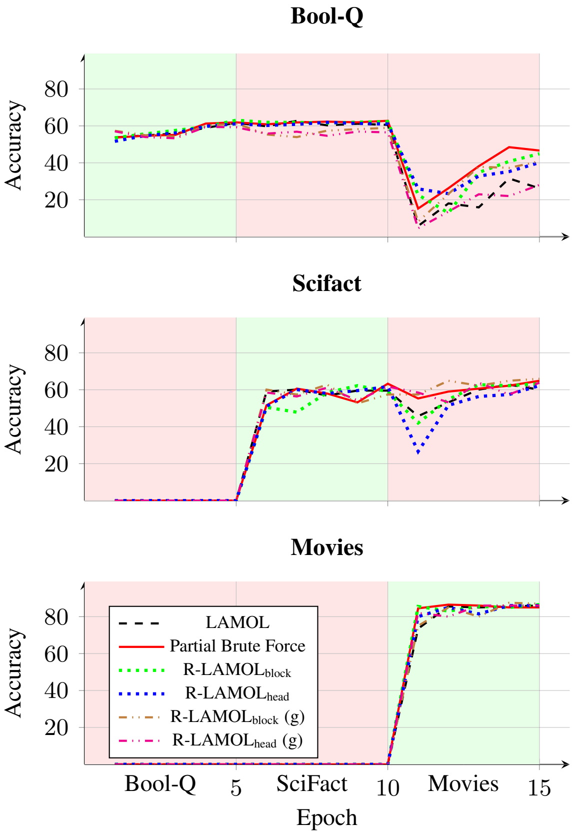

Figure 4: Learning curves of task order BSM. The graphs show accuracy at each epoch for each task. Green background refers to the epochs on which the model is first introduced with a particular task. In this figure, for example, the model is trained on Bool-Q and evaluated on all the three tasks during epoch 1-5.

A sample of accuracy graphs (as the learning progressed) of the compared methods, with the BoolQ → SciFact → Movies (BSM) task order is shown in Figure 4 from top to bottom, respectively. As the first task, BoolQ was not really affected by SciFact, but encountered a heavy drop during the third task of Movies. In the baseline, BoolQ dropped from 61% to a mere 6%, while only rebounding up to 26% at the end. However, after freezing the most plastic block identified by partial brute forcing, BoolQ dropped from 62% to 15%, and rebounding up to 47%. Comparatively, in the second task, SciFact encountered a smaller drop during the third task from 63% to 55%, and then rebounded back to 65%. As the last task, movies was not affected by catastrophic forgetting.

Accuracy graphs for all permutation of tasks is available in Appendix 6 from which we make several observations concerning the effect of task orders on the overall performance:

- There is evidence that Movies accelerate the forgetting process of first task due to the abrupt change in data distribution.
- However, the performance on the task Movies itself is barely affected by the task order. We attribute it to the low difficulty of the task.
- There is usually no interference between the tasks Bool-Q and SciFact when these tasks are trained in adjacency since they are similar.

5.2 Effect of Critical Component Identification (CCI)

It is unrealistic to perform brute force in every single setting. So, it is crucial that our algorithm uses reasonable amount of time while still maintaining improvements from the baseline. The CCI algorithm requires each task except task 1 to be repeated twice. This doubles the time needed to train a single task. Combined with time required for CCI, Rational LAMOL required approximately 2.4 times more time than vanilla LAMOL to completely train a model as shown in Figure 3. On the other hand, our algorithm used only approximately half of the time it took to train in the partial brute force fashion. Currently, CCI only measures plasticity in between two models (M_i and $\hat{M}_{i+1}$). Single model analysis for layer plasticity evaluation is left for future work.

From Table 2, Rational LAMOL$_{\text{block}}$ outperformed LAMOL by 1.83% average accuracy (0.97% average Macro-F1) over all permutations while having smaller standard deviation, indicating that it is also more robust to task orders. Rational LAMOL$_{\text{head}}$ was able to match or outperform LAMOL in five out of six task orders, but the significant decrease in the SBM order lowered the average to a 0.43% gain (and a slight decrease in Macro-F1) from the baseline. Upon further inspection, we found that the pseudo-samples of SciFact contained high variance in quality during pseudo-data replay. In addition to generation token mismatch, i.e., a situation where a pseudo-sample has an answer token from a wrong task, the low volume of SciFact training data affected the quality of the pseudo-samples generated. So, this accelerated catastrophic forgetting rather than alleviating. Without the SBM drop, Rational LAMOL$_{\text{head}}$ performed comparatively well or slightly higher with the block-level. Performing a one-tailed paired t-test on all data points of the total 3 random seeds, we observed that block-level freezing is able to win against the original LAMOL with statistical significance (p-value of 0.023 and 0.042 for block-level and generated block-level respectively). With the SBM result neglected as an outlier, both block-level and head-level significantly improved the results compared with the original LAMOL (p-value of 0.015, 0.014, 0.010, 0.049 for block-level, generated block-level, head-level, and generated head-level respectively). However, there is no conclusive evidence of which method (head-level or block-level freezing) being significantly better (p-value of 0.133). Even though our Rationale LAMOL outperformed the baseline, there was still a gap from the brute force upper bound. This could be due to many incompatibilities between human rationales and machine attention scores, as mentioned in Bao et al. (2018), which made our algorithm choose sub-optimal layers/heads.

5.3 Effect of Unsupervised Rationale Generation

Due to the difference in focus between human and machines, it is conceivable that the rationales generated by InvRat would be mostly misaligned with human rationales. This is shown in Table 3, where the F1 scores of InvRat are quite low when compared with human rationales. Figure 5 shows an example of generated rationales output by InvRat compared with human rationales.

Despite that, Generated Rational LAMOL$_{\text{block}}$ outperformed both Rational LAMOL and LAMOL baseline by 0.84% accuracy (0.31% Macro-F1) and 2.67% accuracy (1.27% Macro-F1) respectively, further reducing the gap to Brute Force, the approximate upper bound of the proposed CCI. This suggests that rationales chosen by InvRat, regardless of how nonsensical they appear, still carry information that eliminates the need for human rationales. The results are consistent with Bao et al. (2018) who showed that significant gains are achieved when using machines attention scores as an additional supervision signal instead of using human rationales.

Last but not least, Figure 3 shows that the process of generating rationales using InvRat, including training and inference, contributed only marginally, about 15 minutes, to the total time used in the training process.

both freeman and tandy seem to be in their element , in full command of their natural charisma . they flirt with the camera and dominate scenes without overtly calling attention to themselves .

Figure 5: An example of rationales from the Movies task. The sentiment for this particular example is negative. The underlined text is a human rationale, while rationales generated by InvRat are shown in red.

	P	R	F1
BoolQ	14.70	18.48	14.57
Movie	5.71	17.89	5.90
SciFact	4.90	5.41	4.99

Table 3: Token-based precision, recall, and F1 showing the agreement between the rationales generated by InvRat and the human-annotated rationales.

6 Conclusion

To effectively retain learned knowledge in LL for NLP tasks, we proposed Rational LAMOL, a learning framework that uses rationales to identify and freeze the most critical components of the model while being trained on a new task. We showed that Rational LAMOL is able to outperform LAMOL by a significant margin. Furthermore, our framework can be applied to any NLP datasets by leveraging unsupervised rationale generation, eliminating the need for human rationales while maintaining comparable improvements. Overall, Rational LAMOL bridges the gap between LL in NLP with model understanding through rationales, exhibiting potential for a true lifelong language learning as well as limiting catastrophic forgetting.

References

Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. 2017. Memory aware synapses: Learning what (not) to forget. *CoRR*, abs/1711.09601.

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. 2019. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*. Cite arxiv:1907.02893.

Yujia Bao, Shiyu Chang, Mo Yu, and Regina Barzilay. 2018. Deriving machine attention from human rationales. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1903–1913, Brussels, Belgium. Association for Computational Linguistics.

Shiyu Chang, Yang Zhang, Mo Yu, and Tommi Jaakkola. 2020. Invariant rationalization. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1448–1458. PMLR.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. 2019. Efficient lifelong learning with a-gem. In *International Conference on Learning Representations (ICLR)*.

Tianqi Chen, Ian Goodfellow, and Jonathon Shlens. 2016. Net2net: Accelerating learning via knowledge transfer. *CoRR*, abs/1511.05641.

Zhiyuan Chen and Bing Liu. 2016. *Lifelong Machine Learning*. Morgan & Claypool Publishers.

Zhiyuan Chen, Nianzu Ma, and Bing Liu. 2015. Lifelong learning for sentiment classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 750–756, Beijing, China. Association for Computational Linguistics.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. Cite arxiv:1412.3555Comment: Presented in NIPS 2014 Deep Learning and Representation Learning Workshop.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. 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)*, pages 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. 2020. ERASER: A benchmark to evaluate rationalized NLP models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4443–4458, Online. Association for Computational Linguistics.

Benjamin Hoover, Hendrik Strobelt, and Sebastian Gehrmann. 2020. exBERT: A visual analysis tool to explore learned representations in Transformer models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 187–196, Online. Association for Computational Linguistics.

Ronald Kemker and Christopher Kanan. 2017. Fearnet: Brain-inspired model for incremental learning. *CoRR*, abs/1711.10563.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526.

Sang-Woo Lee Lee, Jin-Hwa Kim, Jaehyun Jun, Jung-Woo Ha, and Byoung-Tak Zhang. 2017. Overcoming Catastrophic Forgetting by Incremental Moment Matching (IMM). In *Advances In Neural Information Processing Systems 30*.

Tao Lei, Regina Barzilay, and Tommi S. Jaakkola. 2016. Rationalizing neural predictions. In *EMNLP*, pages 107–117. The Association for Computational Linguistics.

Z. Li and D. Hoiem. 2018. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947.

Tianlin Liu, Lyle Ungar, and João Sedoc. 2019. Continual learning for sentence representations using conceptors. 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)*, pages 3274–3279, Minneapolis, Minnesota. Association for Computational Linguistics.

David Lopez-Paz and Marc'Aurelio Ranzato. 2017. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, page 6467–6476.

Cyprien de Masson d'Autume, Sebastian Ruder, Lingpeng Kong, and Dani Yogatama. 2019. Episodic memory in lifelong language learning. *CoRR*, abs/1906.01076.

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. The natural language decathlon: Multitask learning as question answering. *CoRR*, abs/1806.08730.

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

Giang Nguyen, Shuan Chen, Thao Do, Tae Joon Jun, Ho-Jin Choi, and Daeyoung Kim. 2020. Dissecting catastrophic forgetting in continual learning by deep visualization. *CoRR*, abs/2001.01578.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Alessandro Raganato and Jörg Tiedemann. 2018. An analysis of encoder representations in transformer-based machine translation. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 287–297, Brussels, Belgium. Association for Computational Linguistics.

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Explain yourself! leveraging language models for commonsense reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4932–4942, Florence, Italy. Association for Computational Linguistics.

Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. 2016. Progressive neural networks. *CoRR*, abs/1606.04671.

Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. 2017. Continual learning with deep generative replay. In *Advances in Neural Information Processing Systems*, volume 30, pages 2990–2999. Curran Associates, Inc.

Fan-Keng Sun, Cheng-Hao Ho, and Hung-Yi Lee. 2020. Lamol: Language modeling for lifelong language learning. In *International Conference on Learning Representations*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, undefinedukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Jesse Vig. 2019. A multiscale visualization of attention in the transformer model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 37–42, Florence, Italy. Association for Computational Linguistics.

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7534–7550, Online. Association for Computational Linguistics.

Mo Yu, Shiyu Chang, Yang Zhang, and Tommi Jaakkola. 2019. Rethinking cooperative rationalization: Introspective extraction and complement control. 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), pages 4094–4103, Hong Kong, China. Association for Computational Linguistics.

Omar Zaidan and Jason Eisner. 2008. Modeling annotators: A generative approach to learning from annotator rationales. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 31–40, Honolulu, Hawaii. Association for Computational Linguistics.

A Learning Curves of All Task Permutations

Figure 6 to Figure 10 show the learning curves of all task order permutations of the compared methods.

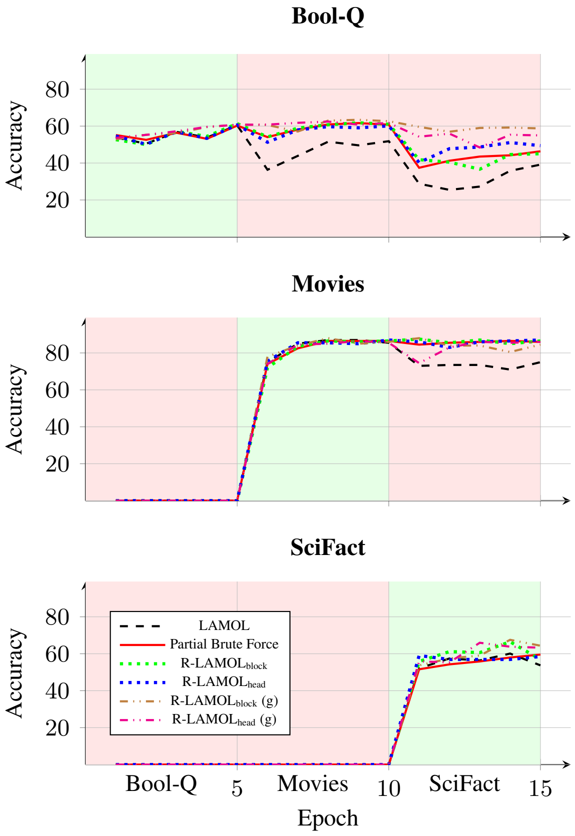

Figure 6: Learning Curves for task order BMS

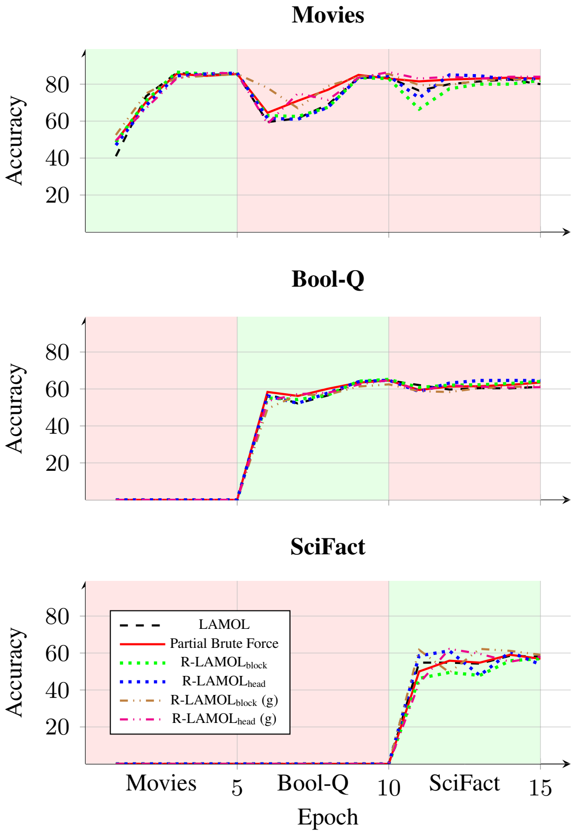

Figure 7: Learning Curves for task order MBS

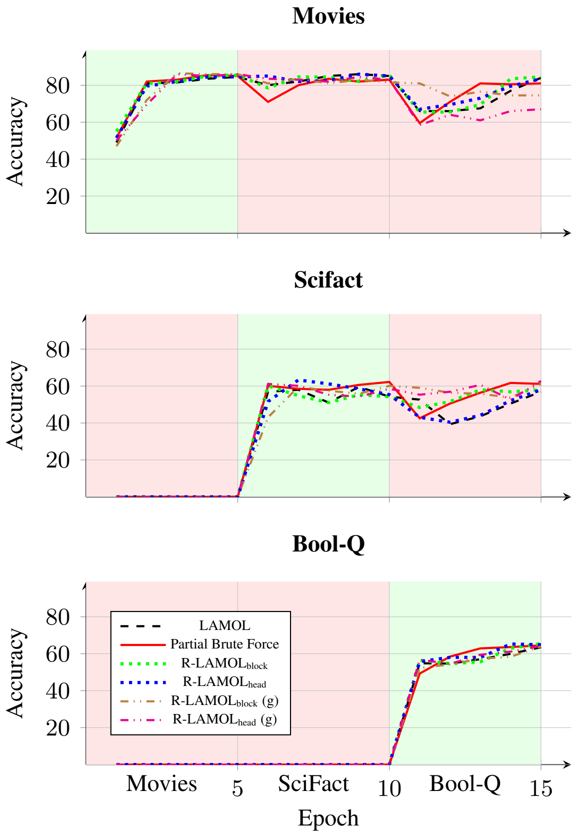

Figure 8: Learning Curves for task order MSB

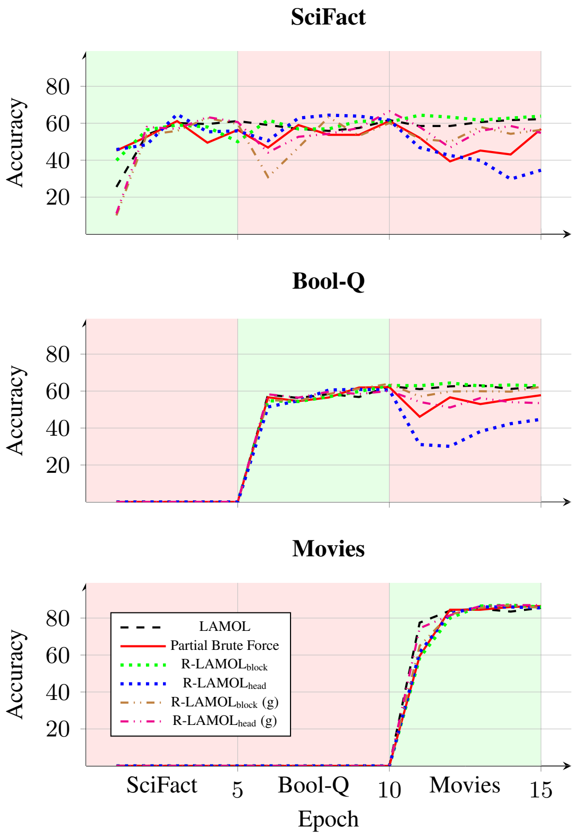

Figure 9: Learning Curves for task order SBM

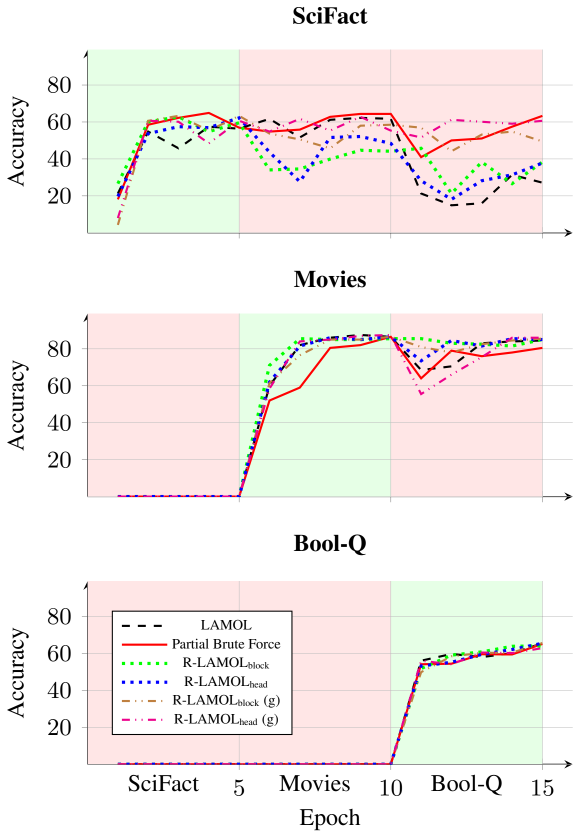

Figure 10: Learning Curves for task order SMB