Lifelong Explainer for Lifelong Learners

Xuelin Situ[†] **Sameen Maruf**[†] **Ingrid Zukerman**[†]
Cecile Paris[‡] **Gholamreza Haffari**[†]
[†]Department of Data Science and Artificial Intelligence, Monash University, Australia
[‡]CSIRO Data61, Australia
[†]{firstname.lastname}@monash.edu
[‡]{firstname.lastname}@data61.csiro.au

Abstract

Lifelong Learning (LL) black-box models are dynamic in that they keep learning from new tasks and constantly update their parameters. Owing to the need to utilize information from previously seen tasks, and capture commonalities in potentially diverse data, it is hard for automatic explanation methods to explain the outcomes of these models. In addition, existing explanation methods, e.g., LIME (Ribeiro et al., 2016), which are computationally expensive when explaining a static black-box model, are even more inefficient in the LL setting. In this paper, we propose a novel Lifelong Explanation (LLE) approach that continuously trains a student explainer under the supervision of a teacher – an arbitrary explanation algorithm – on different tasks undertaken in LL. We also leverage the Experience Replay (ER) mechanism to prevent catastrophic forgetting in the student explainer. Our experiments comparing LLE to three baselines on text classification tasks show that LLE can enhance the stability of the explanations for all seen tasks and maintain the same level of faithfulness to the black-box model as the teacher, while being up to 10^2 times faster at test time. Our ablation study shows that the ER mechanism in our LLE approach enhances the learning capabilities of the student explainer. Our code is available at https://github.com/situsnow/LLE.

1 Introduction

Explaining a model's predictions to practitioners and end users, especially in the case of a blackboxmodel, is non-trivial. Recent research on *eXplainable Artificial Intelligence* usually considers feature attribution as a local explanation, i.e., how much each feature contributes to the outcome of the model. Related works include backpropagation-based methods, where the influence of model outcome is backpropagated according to gradients or layer-wise rules (Bach et al., 2015; Sundararajan et al., 2017; Smilkov et al., 2017; Erion et al., 2021); perturbation-based methods, which observe changes in model performance after feature perturbation (Schwab and Karlen, 2019; Kim et al., 2020), or approximate the local decision boundary through perturbed samples (Ribeiro et al., 2016; Lundberg and Lee, 2017); and model-based methods, which train an explainer model by optimizing an explanation-meritorious objective,[1] such as robustness/stability (Lakkaraju et al., 2020; Alvarez-Melis and Jaakkola, 2018) that requires similar examples to have similar explanations. All these methods aim to explain *static* black-box models, whereas explaining *dynamic* ones, as in the lifelong learning (LL) (Silver et al., 2013) setting, is under-explored.

We propose a Lifelong Explanation (LLE) approach that learns to explain the outcome of a LL black-box under the supervision of a teacher explanation algorithm. The key challenge in LL is to prevent catastrophic forgetting (McCloskey and Cohen, 1989) of knowledge learnt from preceding tasks while learning from a new task. To prevent this, an Experience Replay (ER) mechanism (Li and Hoiem, 2017) is exploited to replay a small amount of past data in order to maintain performance on all seen tasks. However, the dynamically-changing black-box model may make the ER of previously generated explanations sub-optimal. We investigate an ER mechanism that replays previously seen examples together with explanations from the teacher produced in the current step. Specifically, we incorporate the ER mechanism into the training of the student explainer, which focuses on the faithfulness of the generated explanations, i.e., how well an explanation aligns with the LL black-box model outcome.

Our empirical results show that the LLE explainer (i) enhances the stability of explanations, (ii) is as faithful to the black-box model as the teacher, and (iii) is faster than the teacher at test

[1] An objective that maximizes the quality of an explanation.

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 2933–2940
November 7–11, 2021.

time. Our ablation study on ER shows that regenerating the teacher's explanations for past examples significantly improves the faithfulness and stability of the student explanations.

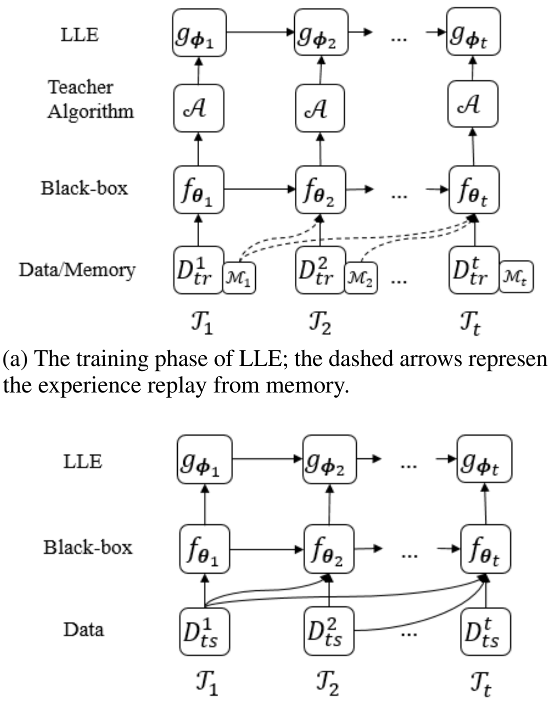
2 Problem Definition

In this paper, we consider a Lifelong Learning (LL) setting comprising a sequence of text classification tasks $\{\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_T\}$. Each task $\mathcal{T}_t$ has its own train/validation/test sets $(D^t_{tr}, D^t_{va}, D^t_{ts})$, each of which contains a set of paired examples $\{(\boldsymbol{x}^t_i, y^t_i)\}^{n_t}_{i=1}$, where $\boldsymbol{x}^t$ is the input (e.g., a document), $y^t \in Y^t$ is the true label (e.g., a topic label), Y^t denotes the label set in task $\mathcal{T}_t$, and n_t is the total number of examples in the set. The goal is to train a classifier $f_{\boldsymbol{\theta}}$ which continuously learns and accumulates knowledge from the data in each task $\mathcal{T}_t$. Specifically, at an arbitrary step t, $f_{\boldsymbol{\theta}}$ optimizes a loss function: $\sum^{n_t}_{i=1} \mathcal{L}(f_{\boldsymbol{\theta}}(\boldsymbol{x}^t_i), y^t_i)$, on the training data D^t_{tr}.

In addition, we require $f_{\boldsymbol{\theta}}$ to remember the preceding knowledge at each step t, so as to maintain its performance on *all* previous tasks, i.e., $\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_{t-1}$. In order to achieve this goal, the classifier $f_{\boldsymbol{\theta}}$ is usually allowed to access a memory that stores a limited number of samples from the previous tasks. The performance measure of $f_{\boldsymbol{\theta}_t}$ at each step t is: $\frac{1}{t}\sum^t_{j=1} acc_{f,j}$, where $acc_{f,j}$ denotes the accuracy of $f_{\boldsymbol{\theta}_t}$ on task $\mathcal{T}_j$.

Lifelong Explanation. To explain a *dynamic* classifier, as in lifelong learning, we consider a new problem setting, called *lifelong explanation*, where at each step t, the input consists of a set of paired examples $\{(\boldsymbol{x}^t_i, f_{\boldsymbol{\theta}_t}(\boldsymbol{x}^t_i))\}^{n_t}_{i=1}$. The goal is to output an explanation $\boldsymbol{r}^t_i$ that indicates how much each dimension of $\boldsymbol{x}^t_i$ contributes to the outcome of $f_{\boldsymbol{\theta}_t}(\boldsymbol{x}^t_i)$. Our approach consists of building an explainer model $g_{\boldsymbol{\phi}}$, i.e., the student, under the supervision of a teacher algorithm, i.e., $g_{\boldsymbol{\phi}}$ uses the explanations generated by the teacher as ground truth.

This approach generalizes to *dynamic* classifiers the learning-to-explain approach in (Situ et al., 2021) for explaining the outcome of *static* classifiers. Since $f_{\boldsymbol{\theta}}$ keeps updating at each step t, we require the explainer $g_{\boldsymbol{\phi}}$ to be able to explain the updated $f_{\boldsymbol{\theta}}$, while maintaining its explanation-meritorious performance, viz faithfulness and stability, on the data from tasks $\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_{t-1}$.


LLE
$g_{\boldsymbol{\phi}_1}$
$g_{\boldsymbol{\phi}_2}$
...
$g_{\boldsymbol{\phi}_t}$
Teacher Algorithm
$\mathcal{A}$
$\mathcal{A}$
$\mathcal{A}$
Black-box
$f_{\boldsymbol{\theta}_1}$
$f_{\boldsymbol{\theta}_2}$
...
$f_{\boldsymbol{\theta}_t}$
Data/Memory
D^1_{tr}
$\mathcal{M}_1$
D^2_{tr}
$\mathcal{M}_2$
...
D^t_{tr}
$\mathcal{M}_t$
$\mathcal{T}_1$
$\mathcal{T}_2$
$\mathcal{T}_t$

(a) The training phase of LLE; the dashed arrows represent the experience replay from memory.

LLE
$g_{\boldsymbol{\phi}_1}$
$g_{\boldsymbol{\phi}_2}$
...
$g_{\boldsymbol{\phi}_t}$
Black-box
$f_{\boldsymbol{\theta}_1}$
$f_{\boldsymbol{\theta}_2}$
...
$f_{\boldsymbol{\theta}_t}$
Data
D^1_{ts}
D^2_{ts}
...
D^t_{ts}
$\mathcal{T}_1$
$\mathcal{T}_2$
$\mathcal{T}_t$

(b) The testing phase of LLE.

Figure 1: Training and testing for LLE. For clarity of exposition, we omit the links from data (D or $\mathcal{M}$) to explanation methods (g or $\mathcal{A}$) — the D or $\mathcal{M}$ inputs to the black-box model f are also inputs to g and $\mathcal{A}$.

3 Lifelong Explanation (LLE)

We now present the training and testing phase of our LLE algorithm (Figure 1) to explain a dynamically-changing black-box classifier.

At time step t in the training phase, we are given a task $\mathcal{T}_t$, its training set D^t_{tr} and the black-box model $f_{\boldsymbol{\theta}_t}$ (Figure 1a). We first collect the explanations $\boldsymbol{r}^t_i$ for each input $\boldsymbol{x}^t_i$ in D^t_{tr} from a teacher algorithm $\mathcal{A}$. Here, $\boldsymbol{r}^t_i$ contains the features (words) in the input $\boldsymbol{x}^t_i$ that are important for the prediction made by $f_{\boldsymbol{\theta}_t}(\boldsymbol{x}^t_i)$. We then train our LLE explainer $g_{\boldsymbol{\phi}_t}$ with the set of explanations $\{\boldsymbol{r}^t_i\}^{n_t}_{i=1}$ for all inputs in D^t_{tr} according to Algorithm 1 and 2.

Training the LLE differs from training the generic LL classifier. Firstly, unlike LL, which predefines task boundaries to determine the memory saving strategy, LLE can simply reuse the same set of memorized examples in LL, and thus is insensitive to this setting; we use *sparse experience replay* (d'Autume et al., 2019), which replays examples from the memory randomly. Secondly, the generic LL algorithm saves the fixed ground-truth label y in the memory. However, in LLE, for an input in the memory $\mathcal{M}$, the ground-truth explanation at time step $t-1$ may differ from the one

Algorithm 1 Lifelong Explanation (LLE)

```
 1: $f_{\boldsymbol{\theta}}$: underlying LL black-box classifier
 2: $g_{\boldsymbol{\phi}}$: explainer model
 3: $\mathcal{A}$: teacher explanation method
 4: $\mathcal{K}$: numbers of randomly selected examples
 5: $\mathcal{M}$: training memory
 6: procedure ExplainerModel($f_{\boldsymbol{\theta}}$)
 7:     $\mathcal{M} \leftarrow \emptyset$
 8:     initialize $\boldsymbol{\theta}_0$ and $\boldsymbol{\phi}_0$ randomly
 9:     for each incoming task $\mathcal{T}_t$ do
10:         $\boldsymbol{\theta}_t \leftarrow$ LLTrain($\boldsymbol{\theta}_{t-1}, D^t_{tr}, \mathcal{M}$)   ▷ training of $f_{\boldsymbol{\theta}}$
11:         $\boldsymbol{\phi}_t \leftarrow$ reTrain($D^t_{tr}, \mathcal{M}, \boldsymbol{\phi}_{t-1}, \mathcal{A}, f_{\boldsymbol{\theta}_t}$)
12:         $\mathcal{M} \leftarrow \mathcal{M} \cup$ randomSubset($D^t_{tr}, \mathcal{K}_t$)
13:         explain $f_{\boldsymbol{\theta}_t}(D^1_{ts}), ..., f_{\boldsymbol{\theta}_t}(D^t_{ts})$ using $g_{\boldsymbol{\phi}_t}$
14:     end for
15: end procedure
```

Algorithm 2 Re-training of $g_{\boldsymbol{\phi}}$

```
 1: procedure reTrain($D_{tr}, \mathcal{M}, \boldsymbol{\phi}, \mathcal{A}, f_{\boldsymbol{\theta}}$)
 2:     $i \leftarrow 0$
 3:     while a stopping condition is not met do
 4:         Randomly pick $\boldsymbol{b} \in D_{tr}, \boldsymbol{b}' \in \mathcal{M}$
 5:         $\mathcal{R} \leftarrow$ consultTeacher($\boldsymbol{b}, \mathcal{A}, f_{\boldsymbol{\theta}}$)   ▷ Appendix A
 6:         $\mathcal{R}' \leftarrow$ consultTeacher($\boldsymbol{b}', \mathcal{A}, f_{\boldsymbol{\theta}}$)
 7:         $\boldsymbol{\phi} \leftarrow \boldsymbol{\phi} - \frac{\eta_i}{|\boldsymbol{b}|+|\boldsymbol{b}'|}\Big(\nabla_{\boldsymbol{\phi}}\mathcal{L}(\boldsymbol{b}, \mathcal{R}, \boldsymbol{\phi}) + \nabla_{\boldsymbol{\phi}}\mathcal{L}(\boldsymbol{b}', \mathcal{R}', \boldsymbol{\phi})\Big)$
 8:         $i \leftarrow i + 1$
 9:     end while
10:     return $\boldsymbol{\phi}$
11: end procedure
```

at time step t, since the black-box $f_{\boldsymbol{\theta}}$ is constantly being updated. Hence, when we train $g_{\boldsymbol{\phi}}$ at time step t (Algorithm 1, line 11), we need to consult the teacher again for the latest explanation (Algorithm 2, lines 5-6). This 'experience replay' approach ensures that $g_{\boldsymbol{\phi}}$ can maintain its explanatory performance on previous examples while learning from new examples. To mitigate catastrophic forgetting, we randomly select a subset of size $\mathcal{K}_t$ from the current training input D^t_{tr} and add it to the memory $\mathcal{M}$ (Algorithm 1, line 12).

In the testing phase (Figure 1b), we no longer require the teacher algorithm $\mathcal{A}$ as the LLE explainer $g_{\boldsymbol{\phi}}$ has already learnt how to produce explanations for unseen examples at each time step t.

4 Experiment

4.1 Dataset and Black-Box Model ($f_{\boldsymbol{\theta}}$)

We randomly select ten tasks from the Amazon Customer Review dataset[2] and fine-tune a pre-trained distilled BERT (Sanh et al., 2019) on these tasks, achieving a 97% test accuracy. Details of the dataset, training of $f_{\boldsymbol{\theta}}$ and accuracies appear in Appendices B.1 and B.2.

4.2 Teacher Explanation Methods ($\mathcal{A}$)

We chose two existing explanation algorithms, LRP (Bach et al., 2015) and LIME (Ribeiro et al., 2016), as the teachers $\mathcal{A}$ in our experiments[3] — experiments in (Montavon et al., 2018) and (Situ et al., 2021) have shown LRP and LIME to be reliable explanation methods in terms of faithfulness and stability. In terms of efficiency, LRP requires one backpropagation pass through the underlying black-box model, and LIME needs to train a linear surrogate model using examples sampled from the neighbourhood of the instance of interest. LPR is time-consuming when the black-box model is large, and LIME is time consuming when the sample size is large. For LIME, we include two baselines, one with sample size 100 (denoted LIME$_s$) and another with sample size 1000 (denoted LIME$_l$), to understand how sample size affects its performance.

4.3 Lifelong Explainer ($g_{\boldsymbol{\phi}}$)

Following the sequence labeling formulation in (Situ et al., 2021), our explainer $g_{\boldsymbol{\phi}}$ takes as input a document $\boldsymbol{x}$ and the outcome predicted by $f_{\boldsymbol{\theta}}$, and outputs a sequence of labels – a label represents the discretized contribution (positive or negative) of a word in $\boldsymbol{x}$ to the outcome.

When $g_{\boldsymbol{\phi}}$ learns from LRP, denoted LLE$_{\text{lrp}}$, the ground-truth explanations from LRP are all positive and categorized into high/medium/low positive based on the thresholds of mean ± standard deviation of all attributions of input $\boldsymbol{x}$. When $g_{\boldsymbol{\phi}}$ learns from LIME, the ground-truth explanations from LIME can be greater, equal or lower than zero. Hence, the categories are taken to be positive, neutral and negative respectively.

We use the Fairseq framework (Ott et al., 2019) to implement the explainer model $g_{\boldsymbol{\phi}}$. Specifically, $g_{\boldsymbol{\phi}}$ is a Transformer encoder (Vaswani et al., 2017) (4 attention heads, 4 blocks) trained with a Stochastic Gradient Descent optimizer and a fixed learning rate (1e-4). For experience replay during training, we randomly select 8 samples from the memory $\mathcal{M}$ on top of the existing mini-batch (size 8). We train the LLE models with three random seeds for 50 epochs each and report the average results with the best checkpoints on the validation set.

[2] We use the datasets provided by HuggingFace datasets API https://huggingface.co/datasets/amazon_us_reviews.

[3] We use the implementations by Wu and Ong (2021) and Ribeiro et al. (2016) for LRP and LIME respectively.

4.4 Performance Metrics

Similarly to (Situ et al., 2021), we compare the faithfulness and stability of explanations produced by our LLE with those produced by the teacher explanation methods $\mathcal{A}$; we also compare the efficiency of the methods.

We measure faithfulness in terms of the Δlog-odds values after masking either the positive or negative contribution words. For an input document $\boldsymbol{x}$ at time step t, Δlog-odds is given by:

$$\text{log-odds}(p(\hat{y}|f_{\boldsymbol{\theta}_t}(\boldsymbol{x}))) - \text{log-odds}(p(\hat{y}|f_{\boldsymbol{\theta}_t}(\tilde{\boldsymbol{x}})))$$

where $\hat{y} = \max_{y \in Y^t} f_{\boldsymbol{\theta}_t}(\boldsymbol{x})$, $\tilde{\boldsymbol{x}}$ is obtained by masking the positive or negative important words in $\boldsymbol{x}$, and $\text{log-odds}(p) = \log \frac{p}{1-p}$.

To measure stability, we first select N (set to 3) most similar test documents to the current test document $\boldsymbol{x}$ based on pairwise ngram similarity. We then compute the Intersection over Union (IoU) according to the positive and negative important words in $\boldsymbol{x}$ and in each of the similar documents $\boldsymbol{x}'$:

$$\frac{1}{|\mathcal{N}(\boldsymbol{x})|} \sum_{\boldsymbol{x}' \in \mathcal{N}(\boldsymbol{x})} \frac{\sum_{\ell \in L} |\boldsymbol{v}_{\boldsymbol{x}}^{\ell} \cap \boldsymbol{v}_{\boldsymbol{x}'}^{\ell}|}{\sum_{\ell \in L} |\boldsymbol{v}_{\boldsymbol{x}}^{\ell} \cup \boldsymbol{v}_{\boldsymbol{x}'}^{\ell}|}$$

where L is the discretized label set. If the teacher is LRP, $L = \{high, low\}$, and if the teacher is LIME, $L = \{positive, negative\}$; $\boldsymbol{v}_{\boldsymbol{x}}^{\ell}$ is the set of words with output label ℓ according to the student explainer $g_{\boldsymbol{\phi}_t}$ or the corresponding teacher $\mathcal{A}$ at time step t.

Efficiency is measured by the average time it takes to produce explanations.

These three metrics are computed for all test sets of tasks seen so far at step t; we report the average values per test sample.

4.5 Results

Figures 2 and 3 respectively display the positive Δlog-odds, measured by masking words with positive attributions, and IoU per test document (higher is better) for all tasks seen so far at each time step (the negative Δlog-odds are shown in Appendix C.1). To evaluate faithfulness, we also include a *Random* baseline, which is the Δlog-odds value obtained by randomly selecting k words in each test sample.[4] When LIME is the teacher, we report the LLE model under the supervision of LIME$_l$ only, denoted LLE$_{\text{lime}}$.

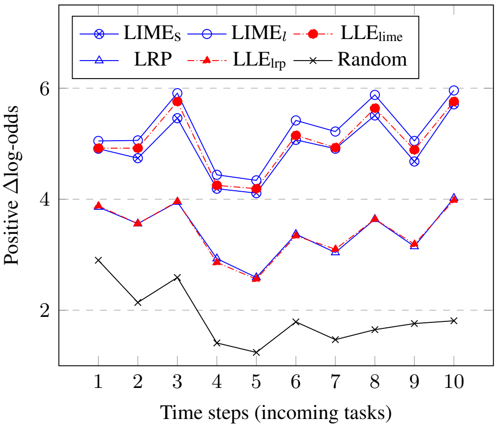


Figure 2: Positive Δlog-odds (higher is better).

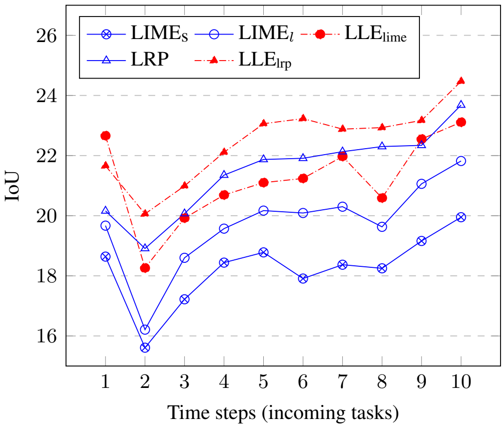


Figure 3: IoU (higher is better).

Faithfulness. As seen in Figure 2, student LLE$_{\text{lrp}}$ and its teacher LRP are almost identically faithful, while LLE$_{\text{lime}}$ never performs significantly worse than LIME$_l$,[5] and performs marginally better than LIME$_s$. We also observe that all methods behave significantly better than the Random baseline. It is worth noting that the LIME family (teacher and student) is consistently and significantly more faithful than the LRP family. In addition, all methods except Random show similar fluctuations in all steps. We hypothesize that both LRP and LIME (and their students) can capture the confidence changes of the underlying black-box $f_{\boldsymbol{\theta}}$ on the examples from tasks seen so far. However, the sampling process in LIME helps capture a smoother local decision boundary than LRP, thus helping it better target the most important features, and thus showing a higher level of faithfulness.

Stability. As shown in Figure 3, students LLE$_{\text{lrp}}$ and LLE$_{\text{lime}}$ achieve higher stability than their teachers LRP and LIME$_l$ respectively. Further, the LRP

[4] We omit the *Random* baseline for stability because the stability of an unfaithful explanation is irrelevant, as shown in Figure 2 and in Figure 6, Appendix C.1.

[5] We use paired t-test with Holm-Bonferroni correction (Holm, 1979) in all our significance tests, with $\alpha < 0.05$.

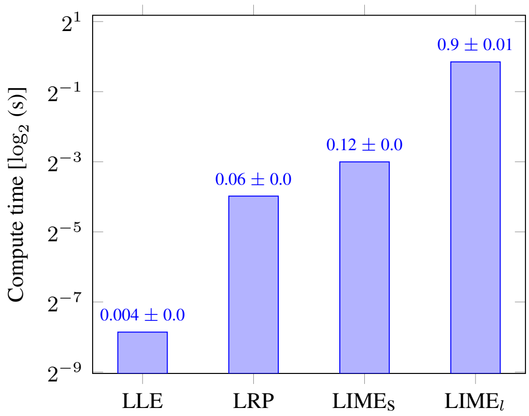


Figure 4: Inference time per test document from all ten tasks (note the log-scale on y-axis).

family outperforms the LIME family, which is in contrast to the trend for faithfulness (Figure 2). However, LLE$_{\text{lime}}$ performs comparably with LRP in most steps, even though its teacher is significantly worse than LRP. This shows that our LLE approach can generate more stable explanations than the teachers while maintaining faithfulness.

Efficiency. Figure 4 shows the processing time of all methods obtained with the same hardware configuration.[6] The size of the black-box model $f_{\boldsymbol{\theta}}$ and the LLE model $g_{\boldsymbol{\phi}}$ are approximately 270MB and 135MB respectively. Given that LRP requires a backward relevance computation per layer in $f_{\boldsymbol{\theta}}$, and LIME requires multiple forward passes (based on sample size), while LLE requires only one forward pass in $g_{\boldsymbol{\phi}}$, it is self evident that LLE is significantly faster than all three baselines.

Experience Replay on LLE. We perform an ablation study to understand the significance of ER in LLE. Specifically, for a particular teacher, we train two other LLE models: (i) without ER during training (denoted LLE-No ER), and (ii) using the explanations generated by the teacher algorithm when the black-box model first sees a task (denoted LLE-Old ER; involves removing line 6 in Algorithm 2). Table 1 shows the Δlog-odds results after masking positive attribution words from these two LLE models and the vanilla LLE, all with LRP as the teacher. The faithfulness of the model with the updated teacher explanations (LLE$_{\text{lrp}}$) is significantly higher than that of the other two LLE variants. Similar observations are obtained in other faithfulness and stability comparisons (Appendix C.2).

	Positive Δlog-odds ↑		
$\mathcal{T}_t$	**LLE$_{\text{lrp}}$-No ER**	**LLE$_{\text{lrp}}$-Old ER**	**LLE$_{\text{lrp}}$**
1	3.8±0.1	3.81±0.1	3.81±0.1
2	3.4±0.07	3.4±0.07	**3.55±0.07**
3	3.82±0.06	3.8±0.06	**3.96±0.06**
4	2.66±0.05	2.68±0.05	**2.85±0.05**
5	2.38±0.04	2.39±0.04	**2.56±0.04**
6	3.17±0.05	3.15±0.05	**3.34±0.04**
7	2.91±0.04	2.9±0.04	**3.1±0.04**
8	3.42±0.04	3.4±0.04	**3.64±0.04**
9	3.03±0.03	3.05±0.03	**3.18±0.03**
10	3.65±0.04	3.63±0.04	**3.98±0.04**

Table 1: Positive Δlog-odds per test document from all seen tasks at each time step; **bold** means the LLE model (learns from teacher LRP) is significantly better than the other two.

5 Conclusion and Future Work

We have proposed a Lifelong Explanation (LLE) method that learns from a teacher and leverages an ER mechanism to explain a constantly-changing black-box. Our experimental results show that LLE can improve the stability of a teacher's explanation, and maintain a comparable level of faithfulness, while performing up to two order of magnitudes faster. Our ablation study has shown the effectiveness of ER using most recently generated explanations.

The performance of LLE in LL settings consisting of problems other than classification, e.g., relation extraction, is still under-explored. The evaluation of LLE based on other merits of explanations, such as simulatability (Hase and Bansal, 2020), can also be an interesting research direction.

Acknowledgements

This research was supported in part by grant DP190100006 from the Australian Research Council. The first author was partly supported by CSIRO/Data61. The computational resources for this work were provided by the Multi-modal Australian ScienceS Imaging and Visualisation Environment (MASSIVE) (www.massive.org.au). We would like to thank the anonymous reviewers for their useful comments to improve the manuscript.

References

David Alvarez-Melis and Tommi S Jaakkola. 2018. On the robustness of interpretability methods. In *Proceedings of the Workshop on Human Interpretability in Machine Learning (WHI 2018)*.

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller,

[6] Intel Xeon Silver 4214R, Quadro RTX 6000, 24GB RAM.

and Wojciech Samek. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140.

Cyprien de Masson d'Autume, Sebastian Ruder, Lingpeng Kong, and Dani Yogatama. 2019. Episodic memory in lifelong language learning. In *Advances in Neural Information Processing Systems*, volume 32.

Gabriel Erion, Joseph D Janizek, Pascal Sturmfels, Scott Lundberg, and Su-In Lee. 2021. Improving performance of deep learning models with axiomatic attribution priors and expected gradients. *Nature Machine Intelligence*.

Peter Hase and Mohit Bansal. 2020. Evaluating explainable AI: Which algorithmic explanations help users predict model behavior? In *Proceedings of ACL*.

Sture Holm. 1979. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70.

Siwon Kim, Jihun Yi, Eunji Kim, and Sungroh Yoon. 2020. Interpretation of NLP models through input marginalization. In *Proceedings of EMNLP*.

Himabindu Lakkaraju, Nino Arsov, and Osbert Bastani. 2020. Robust and stable black box explanations. In *ICML*. PMLR.

Zhizhong Li and Derek Hoiem. 2017. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *Proceedings of ICLR*.

Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*.

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.

Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. 2018. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 73:1–15.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. Fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of SIGKDD*. ACM.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In *NeurIPS Workshop on EMC^2*.

Patrick Schwab and Walter Karlen. 2019. Cxplain: Causal explanations for model interpretation under uncertainty. In *Advances in Neural Information Processing Systems*.

Daniel L Silver, Qiang Yang, and Lianghao Li. 2013. Lifelong machine learning systems: Beyond learning algorithms. In *2013 AAAI spring symposium series*. Citeseer.

Xuelin Situ, Ingrid Zukerman, Cecile Paris, Sameen Maruf, and Gholamreza Haffari. 2021. Learning to explain: Generating stable explanations fast. In *Proceedings of ACL*, pages 5340–5355.

Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. 2017. Smoothgrad: removing noise by adding noise. In *Workshop on Visualization for Deep Learning, ICML*.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of ICML*. JMLR.org.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*.

Zhengxuan Wu and Desmond C Ong. 2021. On explaining your explanations of bert: An empirical study with sequence classification. *arXiv preprint arXiv:2101.00196*.

Appendix A Collecting Teacher Explanations

We present the details of collecting the teacher explanations in Algorithm 3.

Algorithm 3 Collecting teacher explanations

```
procedure CONSULTTEACHER($\boldsymbol{b}, \mathcal{A}, f_{\boldsymbol{\theta}}$)
    $\mathcal{R} \leftarrow \emptyset$
    for each $\boldsymbol{x}$ in $\boldsymbol{b}$ do
        $\hat{y} \leftarrow f_{\boldsymbol{\theta}}(\boldsymbol{x})$
        $\mathcal{R} \leftarrow \mathcal{R} \cup \mathcal{A}(\boldsymbol{x}, \hat{y})$
    end for
    return $\mathcal{R}$
end procedure
```

Appendix B Setup

B.1 Dataset

The Amazon Product Review dataset consists of customer comments on multiple categories of products. We extract the 'review body' and 'star rating' as the input/output for training the classifier. Further, we combine the positive ratings (4 and 5) and negative ratings (1 and 2) to form a binary classification problem. We select the tasks of Home, Outdoors, Wireless, Music, Books, Office products, Luggage, Sports, Jewellery and Video games from this dataset and use them in this order in all experiments. To ensure the classifier learns balanced information from each task, we randomly select 20,000/2,000/2,000 examples as the train/validation/test set, respectively for each of the ten tasks.

B.2 Training of black-box $f_{\boldsymbol{\theta}}$

We train the black-box model $f_{\boldsymbol{\theta}}$ using an Adam optimizer (Loshchilov and Hutter, 2019) (0.1 weight decay and 1e-5 learning rate) for one epoch. To prevent catastrophic forgetting, we randomly save training examples of each task into memory $\mathcal{M}$. We maintain a fixed memory size (64 examples) for each task. We randomly replay 64 examples from $\mathcal{M}$ after every 800 mini-batches which gives us 1% replay rate. The average test accuracy at each time step, as shown in Figure 5, demonstrates that $f_{\boldsymbol{\theta}}$ maintains the performance on seen tasks while learning from new task.

Appendix C Results

C.1 Negative Δlog-odds

Figure 6 compares the Δlog-odds after masking the same k number of words with negative attributions for each of the LIME-based models. We can see that LLE$_{\text{lime}}$ performs very similar to its teacher LIME$_l$ and becomes better than LIME$_s$ after seven tasks. We do not compare LRP-based methods here, as LRP considers all words contribute positively to the final prediction.

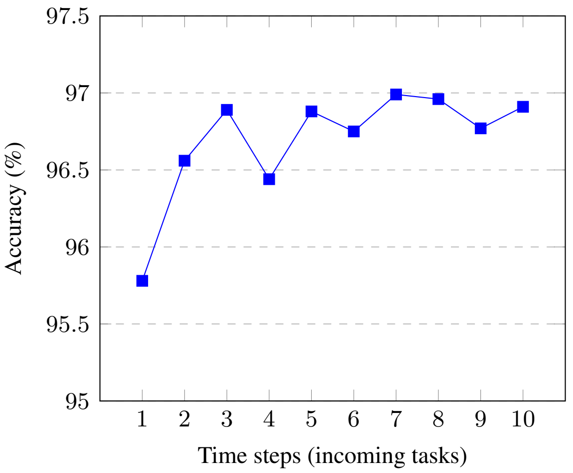


Figure 5: The average test accuracy of the lifelong learning classifier at each time step.

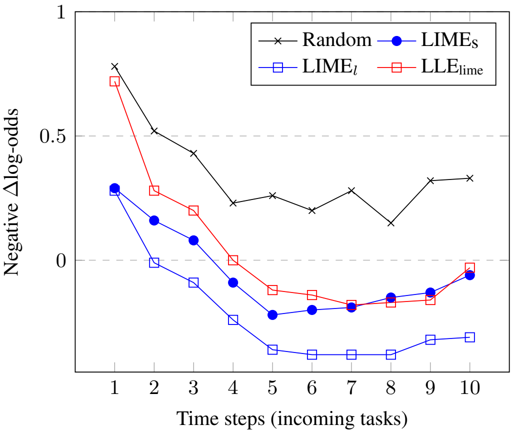


Figure 6: Negative Δlog-odds (lower is better).

C.2 Experience Replay on LLE

The effect of using ER is measured using Δlog-odds and IoU in Tables 2 to 5. These experimental results prove that LLE is able to generate better explanations in terms of faithfulness and stability by leveraging the most recent ground-truth (teacher explanations) in ER.

	Positive Δlog-odds ↑		
$\mathcal{T}_t$	**LLE$_{\text{lime}}$-No ER**	**LLE$_{\text{lime}}$-Old ER**	**LLE$_{\text{lime}}$**
1	4.92±0.09	4.92±0.09	4.92±0.09
2	4.44±0.06	4.53±0.06	**4.92±0.06**
3	5.62±0.05	5.66±0.05	**5.76±0.05**
4	4.13±0.04	4.1±0.04	**4.25±0.04**
5	4.15±0.04	4.03±0.04	**4.19±0.04**
6	5.12±0.04	5.02±0.04	**5.15±0.04**
7	4.96±0.03	4.97±0.03	4.94±0.03
8	5.01±0.04	5.29±0.04	**5.64±0.03**
9	4.57±0.03	4.58±0.03	**4.89±0.03**
10	5.71±0.04	5.66±0.04	**5.77±0.04**

Table 2: Positive Δlog-odds per test document from all seen tasks at each time step; **bold** means the LLE model (learns from teacher LIME$_l$) is significantly better than the other two.

	Negative Δlog-odds ↓		
$\mathcal{T}_t$	**LLE$_{\text{lime}}$-No ER**	**LLE$_{\text{lime}}$-Old ER**	**LLE$_{\text{lime}}$**
1	0.72±0.07	0.72±0.07	0.72±0.07
2	0.25±0.04	0.27±0.04	0.25±0.04
3	0.17±0.03	**0.15±0.03**	0.19±0.03
4	0.0±0.02	-0.0±0.02	-0.01±0.02
5	-0.11±0.02	-0.11±0.02	**-0.13±0.02**
6	-0.13±0.02	-0.12±0.02	**-0.14±0.02**
7	-0.16±0.02	-0.15±0.02	**-0.18±0.02**
8	-0.17±0.02	-0.13±0.02	-0.17±0.02
9	-0.12±0.01	-0.11±0.01	**-0.16±0.02**
10	-0.04±0.01	-0.03±0.02	-0.03±0.02

Table 3: Negative Δlog-odds per test document from all seen tasks at each time step; **bold** means the LLE model (learns from teacher LIME$_l$) is significantly better than the other two.

	IoU		
$\mathcal{T}_t$	**LLE$_{\text{lrp}}$-No ER**	**LLE$_{\text{lrp}}$-Old ER**	**LLE$_{\text{lrp}}$**
1	21.43±0.63	21.43±0.63	21.43±0.63
2	20.8±0.56	20.91±0.6	**21.15±0.63**
3	**20.19±0.49**	19.74±0.5	19.86±0.5
4	22.35±0.6	22.51±0.63	22.52±0.63
5	24.72±0.89	24.54±0.86	**25.44±0.88**
6	22.26±0.66	21.96±0.67	**22.53±0.69**
7	24.31±0.82	24.16±0.81	**24.6±0.82**
8	23.03±0.59	22.77±0.6	22.98±0.6
9	22.11±0.65	22.38±0.64	**22.51±0.66**
10	25.67±0.69	25.54±0.71	**26.08±0.72**

Table 4: IoU per test document from all seen tasks at each time step; **bold** means the LLE model (learns from teacher LRP) is significantly better than the other two.

	IoU		
$\mathcal{T}_t$	**LLE$_{\text{lime}}$-No ER**	**LLE$_{\text{lime}}$-Old ER**	**LLE$_{\text{lime}}$**
1	22.66±0.66	22.66±0.66	22.66±0.66
2	19.28±0.7	18.87±0.72	**19.71±0.7**
3	18.2±0.59	18.0±0.6	**18.44±0.59**
4	20.81±0.73	20.18±0.75	20.79±0.74
5	24.12±0.97	24.05±0.97	24.18±0.96
6	20.09±0.78	19.9±0.79	**20.35±0.79**
7	23.43±0.91	23.07±0.92	**24.07±0.9**
8	19.26±0.71	19.72±0.7	**19.95±0.71**
9	20.7±0.76	20.52±0.76	**21.47±0.74**
10	25.09±0.86	25.28±0.85	**25.72±0.84**

Table 5: IoU per test document from all seen tasks at each time step; **bold** means the LLE model (learns from teacher LIME$_l$) is significantly better than the other two.