ConfliLPC: Logits and Parameter Calibration for Political Conflict Analysis in Continual Learning

Xiaodi Li
Computer Engineering
The University of Texas at Dallas
Richardson, USA
xiaodi.li@utdallas.edu

Niamat Zawad
Computer Science
The University of Texas at Dallas
Richardson, USA
niamat.zawad@utdallas.edu

Patrick T. Brandt
Economic, Political and Policy Science
The University of Texas at Dallas
Richardson, USA
pbrandt@utdallas.edu

Javier Osorio
School of Government and Public Policy
The University of Arizona
Tucson, USA
josorio1@arizona.edu

Vito D'Orazio
Department of Political Science
West Virginia University
Morgantown, USA
vito.dorazio@mail.wvu.edu

Latifur Khan
Computer Science
The University of Texas at Dallas
Richardson, USA
lkhan@utdallas.edu

2024 IEEE International Conference on Big Data (BigData) | 979-8-3503-6248-0/24/$31.00 ©2024 IEEE | DOI: 10.1109/BigData62323.2024.10826026

***Abstract*—The ConfliLPC framework introduces an innovative integration of Logits and Parameter Calibration (LPC) with the ConfliBERT model, tailored specifically for the nuanced analysis of political conflict and violence. This paper details the development and application of ConfliLPC, highlighting its robust capability to adapt to evolving data landscapes without succumbing to catastrophic forgetting (CF), a common challenge in machine learning models applied to dynamic domains such as political science. ConfliLPC enhances accuracy and adaptability by continually adjusting its parameters to accommodate new information while retaining valuable historical insights. The framework has been rigorously tested across various conflict scenarios, demonstrating superior performance in real-time analysis and predictive tasks. This work serves as a significant contribution to the fields of political science, conflict research, and applied machine learning, providing a powerful tool for analysts and policymakers engaged in the understanding and resolution of political conflict. The experimental results highlight the efficiency of the ConfliLPC method and its capability to minimize CF. Our code is publicly available [1].**

***Index Terms*—continual learning, catastrophic forgetting, natural language processing, political conflict analysis**

I. Introduction

The dynamic landscape of global politics, with conflicts involving diverse actors, demands tools that understand historical data and adapt to new, unexpected developments. This need mirrors the concept of continual learning in machine learning, where systems learn from a stream of continuously changing data while retaining previously acquired knowledge. During the 2014 Ukrainian crisis [1], analysts using traditional static models faced challenges adapting to new types of data stemming from Russian cyber attacks and other facets of the conflicts in Crimea and Eastern Ukraine. New events, along with new political actors, required manual recalibration of models to maintain accuracy.

Multi-tasked Learning (MTL) is a strategic machine learning approach that improves models' generalization and efficiency by training them on multiple related tasks simultaneously [2]. This method exploits the commonalities between tasks to construct a shared model architecture, where layers common to all tasks learn general features and task-specific layers adapt to the particularities of each task. For example, there are 3 different tasks binary classification, multi-class classification, and multi-label classification. Multi-task learning will assign different layers for different tasks and learn the three tasks simultaneously. MTL not only boosts model performance by leveraging shared information and reducing the risk of overfitting, but it also enhances computational efficiency by using fewer parameters than would be necessary for separate models for each task. Applied widely in fields like natural language processing and computer vision, MTL is particularly advantageous in environments requiring diverse outputs and where data for some tasks may be limited.

Following the principles of MTL, Continual Learning (CL) presents a different but complementary approach, focusing on the ability of a model to learn continuously from a stream of data while retaining previously acquired knowledge. Unlike MTL, which trains on multiple tasks simultaneously, CL trains on tasks sequentially, updating its knowledge without the need to retain all past data. This method is crucial for applications where data arrives in an ongoing manner or where it is impractical to store all historical data. CL techniques such as experience replay, elastic weight consolidation, and progressive neural networks are designed to prevent catastrophic forgetting (CF), thus enabling a model to adapt to new tasks or data while maintaining proficiency in previously learned tasks [3].

The ConfliLPC framework integrates Logits and Parameter Calibration (LPC) [4] with the ConfliBERT [5] model specifically designed for political conflict and violence texts. ConfliBERT, adept at handling domain-specific nuances through extensive pre-training, is enhanced by LPC to continually refine its predictive performance. This integration allows

[1]https://github.com/Xiaodi-Li/ConfliLPC

ConfliLPC to adapt effectively to new data, similar to how CL systems incorporate new knowledge without forgetting existing information. This represents a significant advancement in NLP applications within political science, providing a robust framework that dynamically adapts to the changing nature of political conflicts.

The dynamic adaptability of ConfliLPC is crucial in real-world applications where political landscapes continually evolve. Traditional models often fail to capture the subtleties and complexities of political language as contexts shift, which can lead to degraded performance over time. ConfliLPC leverages ongoing calibration to adjust its parameters, ensuring consistent performance even as the nature of political discourse changes. This makes it an invaluable tool for political analysts and policymakers who require up-to-date, accurate assessments of political environments for making timely decisions.

The contributions of ConfliLPC are threefold: (1) ConfliLPC is specifically designed to continually adapt to new conflict-related data, enabling it to process evolving political events without the need for frequent retraining. This capability is essential for maintaining the relevance and accuracy of conflict analysis models in rapidly changing global scenarios; (2) ConfliLPC fine-tunes the sensitivity and specificity of the ConfliBERT model based on incoming data. This calibration addresses shifts in discourse, ensuring that the model remains robust across different contexts and time periods; (3) ConfliLPC demonstrates robust performance across all evaluated tasks and significantly reduces forgetting, maintaining its effectiveness over extensive periods and diverse data streams.

II. Background

A. Continual Learning

Continual learning, also known as lifelong learning, represents a vital approach in the domain of machine learning, where algorithms are developed to progressively acquire, fine-tune, and preserve knowledge over time. This paradigm is particularly critical because it addresses a major flaw commonly observed in traditional machine learning models known as catastrophic forgetting. Catastrophic forgetting (CF) occurs when a model, upon learning new information, tends to lose the information it had learned previously.

The overarching aim of continual learning is to mimic the human capacity to persistently gather and refine knowledge throughout the life of an individual without substantial loss of earlier acquired information. This capability is particularly crucial in scenarios where the data environment continuously evolves, or where systems must adapt seamlessly to new data [6]. Such adaptive behavior is imperative for the effective, dynamic deployment of AI systems in real-world conditions.

To overcome the challenges of continual learning, several innovative strategies have been developed: regularization techniques: These methods are designed to maintain the stability of previously learned knowledge while allowing for the acquisition of new information. Techniques such as Elastic Weight Consolidation (EWC) apply a penalty to the learning algorithm, which discourages drastic alterations to weight parameters critical for prior tasks [7]. Another noteworthy method is Synaptic Intelligence (SI), which dynamically evaluates the significance of each parameter throughout the learning process, ensuring that important parameters are preserved as new tasks are learned [8]. A dynamic architectures strategy adapts the structure of neural networks to accommodate new tasks. For example, Progressive Neural Networks introduce new sets of neurons for each new task while maintaining the integrity of networks trained on previous tasks [9]. Other variations include dynamically expandable networks, which increase their capacity based on the complexity of incoming tasks, allowing for scalable and flexible learning structures. Other Memory-based Approaches focus on retaining access to old data through replay mechanisms or experience replay. By periodically rehearsing past instances, these techniques help the model to preserve old knowledge while integrating new information [10]. This can be accomplished by either storing actual samples from previous tasks or generating synthetic samples using generative models, which helps in overcoming the limitations related to direct data retention.

Despite these advancements, continual learning still encounters several hurdles, including effective memory management, scalability of the techniques, and balance between adaptability to new knowledge and stability of the old. Recent research creates sophisticated hybrid models that amalgamate different strategies. For instance, integrating meta-learning with architectural advancements can enable systems to quickly adapt to new tasks with minimal forgetting [11].

The potential applications of continual learning are extensive and cover various sectors. For instance, in autonomous driving, continual learning enables models to perpetually assimilate new data about varying road conditions and driver behaviors. In health care, it allows systems to adapt to new patient data and emerging medical conditions. Additionally, continual learning significantly enhances the personalization of technology services, adapting systems dynamically to changes in user preferences and behaviors. Thus continual learning is not merely an academic concept but a practical necessity for the deployment of robust AI systems in an ever-changing world, ensuring that they remain adaptive and relevant.

B. Political Conflict Analysis

Scholars and professionals in conflict and security circles focus heavily on the study of political violence. Governments and organizations allocate significant resources to observe, understand, and forecast the complexities of social unrest, political violence, and armed conflicts on a global scale [12]–[14]. Conflict analysis, a critical branch of international relations and political science, examines interactions between governments, their adversaries, and civilian populations. These interactions range from tangible actions and verbal confrontations to protests, riots, government suppression, insurgencies, civil wars, terrorism, human rights violations, genocides, criminal activity, forced displacements, traditional and unconventional warfare, nuclear deterrence, peacekeeping efforts, diplomatic tensions, and collaborations.

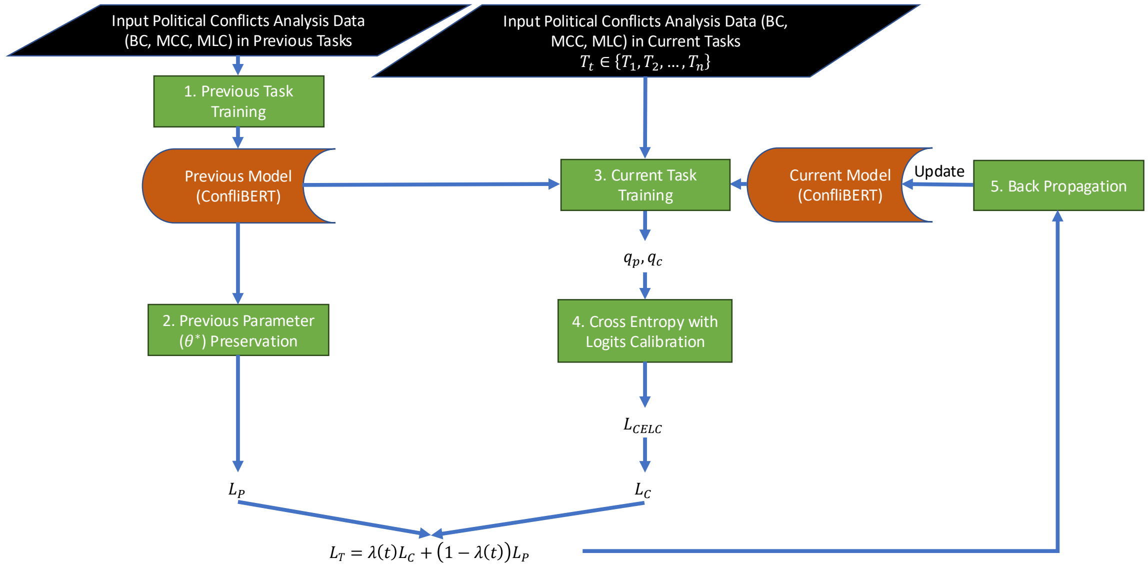

Fig. 1. The Overview of ConfliLPC Framework. (1) We first train the previous model (ConfliBERT) on the large-scale input texts and initialize our model (for current tasks) the same as the previous one. (2) We do previous parameter preservation to preserve the parameters of the trained model and compute the loss for the previous model L_P. (3) During the current task training, we compute logits q_p and q_c. (4) We do logits calibration (e.g., cross entropy with logits calibration for classification tasks) given q_p and q_c using L_{CELC} as the loss for the current model L_C. Then, the objective function drifts from L_P to L_C gradually with the annealing coefficient $\lambda(t)$. (5) Finally, we perform back propagation to update the parameters of the current model. BC represents Binary Classification, MCC represents Multi-Class Classification, and MLC represents Multi-Label Classification.

Traditionally analysts have used manual coding [15] to categorize or label specific events in texts based on predefined criteria. This process requires labor-intensive time and costly financial investments, depending heavily on domain experts. To overcome these limitations, developers created automated systems to classify and extract structured event data. However, these systems often rely on outdated pattern recognition methods and extensive dictionaries, leading to frequent inaccuracies and high maintenance costs.

Recent advancements in natural language processing (NLP) have shifted the landscape, with pre-trained transformer-based language models [16]–[18] leading the charge. Using self-supervised learning with large volumes of unlabeled text, these models significantly reduce the need for labor-intensive annotation through transfer learning. Transformer models, with their parallelized training process, handle large datasets efficiently. Leveraging these advances, researchers developed ConfliBERT [5], a pre-trained language model specifically designed to analyze conflict and political violence. ConfliBERT improves performance on conflict-related tasks and reduces the need for extensive annotation. Its development involved two steps: (1) training a BERT [17]-based model on a domain-specific political violence corpus, and (2) evaluating the model across 12 datasets covering various downstream tasks.

Despite its effectiveness in single-task settings, ConfliBERT struggles in continual learning environments due to CF. CF occurs when a model trained sequentially on new tasks overwrites knowledge gained from earlier tasks. In political conflict analysis, this is particularly problematic because models must adapt to constant shifts, with new events such as protests or civil wars arising while maintaining an understanding of past conflicts. The loss of previously learned information hampers the model's ability to analyze and predict complex, evolving conflicts where historical context is crucial.

To tackle this challenge, we propose an approach to address CF in ConfliBERT, ensuring the model retains prior knowledge while learning new tasks. to improve ConfliBERT's capacity to handle sequential training, we enable it to adapt to new conflict scenarios without sacrificing performance on previously learned tasks. This enhancement strengthens the model's adaptability but also makes it more reliable for conflict analysis, where the ability to retain and build upon prior knowledge is essential for accurate and timely decisions.

III. Problem Statement

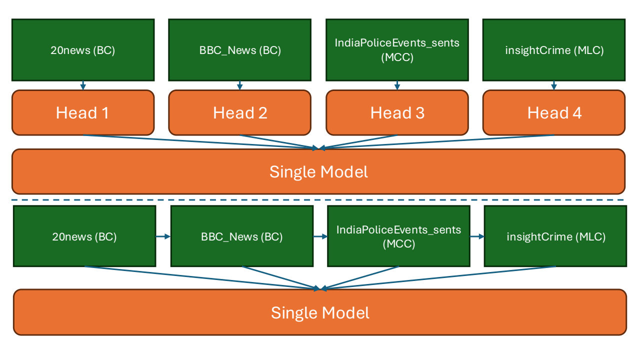

Fig. 2. The Difference between Multi-tasked Learning and Continual Learning. The Multi-tasked Learning is depicted above, while the Continual Learning is shown below.

Fig. 2 shows the difference between multi-tasked learning and continual learning. It shows how multi-tasked learning trains one model for one task simultaneously while continual learning trains a single model for all the tasks sequentially. In CL there will be CF while MTL does not have this problem because in continual learning, the current model is trained based on the previous model while in multi-tasked learning, there is a separate model trained for each task. The detailed differences are as follows:

Multi-tasked Learning (MTL) involves training separate models across multiple tasks simultaneously. This approach leverages the inherent relationships and shared features among the tasks to enhance the performance of the model and generalization capabilities. By training on several tasks at once, the model learns to identify and exploit commonalities across the tasks, which can lead to more robust representations and prevent overfitting specific to one task. This is particularly important in complex environments where tasks share underlying structures or features, allowing the model to perform well across a broader range of tasks than if it were trained separately on each task.

Continual Learning (CL) focuses on training a model to handle new tasks sequentially while retaining the knowledge from previously learned tasks, crucial in dynamic environments where new data or tasks are introduced over time. The ability to train on one task at a time without forgetting previous tasks (avoiding catastrophic forgetting) is vital for applications that require the model to adapt to new conditions continually. Continual learning is especially important for long-lived systems expected to accumulate and refine knowledge progressively without the need for retraining from scratch.

IV. Proposed Approach

The Logits and Parameter Calibration framework for Political Conflict Analysis (ConfliLPC), comprises two essential components: (1) Logits Calibration (LC), designed to adjust logits to reduce logits forgetting and improve accuracy, and (2) Parameter Calibration (PC), intended to adjust parameters to minimize parameter forgetting. For LC, we employ the Cross Entropy with Logits Calibration (CELC) technique in classification tasks. The parameter calibration consists of two key elements: (1) Previous Parameter Preservation (PPP) focuses on maintaining parameters from earlier tasks, and (2) Current Task Training (CTT) reduces the deviation from previous tasks to current tasks during updates. The ConfliLPC framework integrates these components into a novel optimization algorithm that utilizes the Adam optimizer [19].

A. Logits Calibration

The Cross Entropy (CE) Loss [20] is a widely used loss function for classification tasks in deep learning frameworks. It applies a log softmax function to the output logits from the neural network, followed by computing the negative log likelihood (NLL) loss based on the log softmax results. The cross entropy loss is typically defined as follows:

$$L_{CE}(q) = -\sum_{i=1}^{N_C} p_i \log\left(\frac{\exp(q_{c,i})}{\sum_{j=1}^{N_C} \exp(q_{c,j})}\right) \tag{1}$$

Here, N_C denotes the total number of classes for the current tasks. The term $q_{c,i}$ refers to the output logits for class i from the current model on the current tasks. The variable p_i acts as the binary label for class i, where $p_i = 1$ if the input data x belongs to class i, and $p_i = 0$ otherwise. The traditional cross entropy loss focuses only on the performance of the current model, leading to a risk of CF as the model training progresses. We address this by updating the previous model on the current tasks and calculate its output logits q_p.

Drawing inspiration from LCwoF [21], we incorporate the logits from the previous model into the cross entropy loss. Unlike LCwoF, our approach involves adjusting the logits calibration by adding the difference between the logits of the current and previous models $(q_{c,i} - q_{p,i})$ to the current model's logits $q_{c,i}$ for each class. This method helps preserve crucial logits information for each class from the previous model in an element-wise manner. Our modified Cross Entropy with Logits Calibration (CELC) Loss is described in (2):

$$L_{CELC} = -\sum_{i=1}^{N_C} p_i \log\left(\frac{\exp(q_{c,i} + \mu(q_{c,i} - q_{p,i}))}{\sum_{j=1}^{N_C} \exp(q_{c,j} + \mu(q_{c,j} - q_{p,j}))}\right) \tag{2}$$

We scale the difference between the logits of the current and previous models $(q_c - q_p)$ by a weighting factor $\mu \in [0, 1]$, to manage the calibration. This modified loss function enhances the accuracy of the model throughout the training process by either rewarding or penalizing the logits for the correct class. Specifically, a reward is given if the logits of the correct class $q_{c,i}$ exceeds $q_{p,i}$, and a penalty is applied if the logits of the correct class $q_{c,i}$ is less than $q_{p,i}$. In this way, the model will give more propensity to correct class rather than wrong class.

B. Parameter Calibration

Parameter Calibration (PC) mitigates CF by incorporating a penalty into the training loss whenever there is a discrepancy between the current model's parameters and those of the previous model. We add the squared differences between these parameters to achieve this. The Parameter Calibration method consists of two key components: (1) Previous Parameter Preservation (PPP) and (2) Current Task Training (CTT).

To facilitate the transition of the target task from previous tasks to current ones, we propose a technique that enables the objective function to shift smoothly from L_P to L_C using an annealing coefficient, $\lambda(t)$:

$$L_T = \lambda(t) L_C + (1 - \lambda(t)) L_P \tag{3}$$

where t denotes the timestep during the training process. The function $\lambda(t) = \frac{1}{1+\exp(-r\cdot(t-t_0))}$, defined as the sigmoid annealing function [22], where r is the hyperparameter that determines the rate of annealing and t_0 is the hyperparameter that dictates the transition timesteps. Over time, this mechanism allows the objective of the model to gradually transition from previous tasks to current tasks. By implementing backpropagation, we subsequently update the parameters of the current model through parameter calibration. Here, L_P is the loss of the previous task and L_C is the loss of the current task.

1) Previous Parameter Preservation: As illustrated in Fig. 1, the Previous Parameter Preservation (PPP) component focuses on retaining the parameters from the earlier model. This is achieved by applying a regularization to the parameters' posterior based on the data. PPP improves upon the RecAdam method [7], quantifying the significance of each parameter through the assignment of importance weights, Ω. Throughout the training process, the current model significantly protects the most critical parameters by imposing stricter penalties on alterations to these key parameters. Our comprehensive proposed loss function L_P is (4):

$$\begin{aligned} L_P &= -\log p(\theta|D_P) \\ &\approx -\log p(\theta^*|D_P) + \delta(\theta-\theta^*)^T H(\theta^*) \\ &\quad \Omega(\theta)(\theta-\theta^*) \\ &\approx \delta(\theta-\theta^*)^T H(\theta^*)\Omega(\theta)(\theta-\theta^*) \\ &\approx \delta(\theta-\theta^*)^T (NF(\theta^*) + H_{prior}(\theta^*)) \\ &\quad \Omega(\theta)(\theta-\theta^*) \\ &\approx \delta N \sum_{ij} F_{ij}\Omega_{ij}(\theta_{ij}-\theta^*_{ij})^2 \\ &\approx \delta NF \sum_{ij} \Omega_{ij}(\theta_{ij}-\theta^*_{ij})^2 \\ &= \delta\gamma \sum_{ij} \Omega_{ij}(\theta_{ij}-\theta^*_{ij})^2, \end{aligned} \tag{4}$$

where δ serves as a hyperparameter for the regularization. The Hessian matrix of the optimization objective with respect to θ is denoted by $H(\theta)$. This matrix can be approximated using the empirical Fisher information matrix $F(\theta)$ [23]. N represents the total number of data inputs in D_P. $H_{prior}(\theta)$ refers to the Hessian matrix of the negative log prior probability $-\log p(\theta)$. Elastic Weight Consolidation (EWC) typically disregards $H_{prior}(\theta)$ and simplifies $H(\theta)$ by setting its diagonal values equal to those of $F(\theta)$. Consequently, in the final steps of the derivation, NF is substituted with a constant γ, viewed as a coefficient for the quadratic penalty. Throughout the derivation, the term $-\log p(\theta|D_P)$ is omitted since it remains constant with respect to θ^*. The importance weights $\Omega(\theta)$ are calculated based on how sensitive the squared L2 norm of the output of the function is to changes in the parameters. The values Ω_{ij} are derived by aggregating the gradients across the data points as detailed in (5):

$$\Omega_{ij} = \frac{1}{N}\sum_{k=1}^{N} \|g_{ij}(x_k)\| \tag{5}$$

In the formula, $g_{ij}(x_k) = \frac{\partial[l_2^2(f(x_k;\theta))]}{\partial\theta_{ij}}$ represents the gradient of the squared L2 norm of the output of the neural network with respect to the parameter θ_{ij}. Here, the output from $f(x_k;\theta)$ denotes the loss of the network.

As mentioned in (4), θ_{ij} refers to the parameter in the current model that connects pairs of neurons n_i and n_j across two successive layers. θ^* signifies the parameters of the previous model, which are considered to be at a local minimum within the parameter space as indicated in (6):

$$\theta^* = \arg\min_{\theta}\{-\log p(\theta|D_P)\} \tag{6}$$

2) Current Task Training with Continual Learning: During the training process for the current task, we simultaneously train the current model and assess the previous model using the current tasks. In the context of continual learning, we start by training on Task T_1 and then conduct evaluations on the same task. Subsequently, the training shifts to Task T_2, followed by evaluations on tasks T_1 and T_2. This process continues with Task T_3, where after training, evaluations are performed on tasks T_1, T_2, and T_3, and so forth. We detail the procedure for a specific current task here, explaining how we account for drift from previous tasks into the current task. The output of the neural network, which is the loss of the model, can be described as follows:

$$L_C = f_t(x;\theta_{t-1}) = L_{CELC}(Q(x;\theta_{t-1})), \tag{7}$$

where t denotes the timestep, the loss is calculated using the CELC method, specifically designed for classification tasks. $Q(x;\theta_{t-1})$ is the function of both the current and the previous models, which produces logits using data inputs x and the model parameters θ_{t-1} from timestep $t-1$.

C. ConfliLPC Algorithm

Here we integrate Logits Calibration (CELC) and Parameter Calibration (PPP and CTT) into a new optimization algorithm, as detailed in Algorithm 1. The logits calibration (LC) process is outlined in lines 6 to 8 of the algorithm, while parameter calibration (PC) is detailed in lines 9 to 16 and line 22. We present the ConfliLPC Algorithm which combines a quadratic penalty with importance weights and an annealing coefficient into a comprehensive optimization algorithm. This integration is achieved by separating these elements from the gradient update process used in the Adam optimization algorithm [19].

Lines 9 to 14, show how to calculate Ω by initializing Ω as a tensor filled with the scalar value one. The size of Ω is the same as that of the parameter size of the previous model and the current model. From line 10 to line 13, we accumulate the gradients of the squared L2 norm of the learned neural network over the given data inputs to obtain importance weights Ω_{ij} for parameter θ_{ij}. In line 14, we compute the mean value of Ω_{ij} by dividing it by N. Here, N is the total number of data inputs at a given phase. In line 16, we compute the gradients of the loss function as a weighted combination of the gradients of L_C and L_P. In line 22, we update the network parameters θ by the gradient descent method.

Algorithm 1 ConfliLPC

given initial learning rate $\alpha \in \mathbb{R}$, momentum factors $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, pre-trained parameter vector $\theta^* \in \mathbb{R}^n$, hyperparameter for the regularizer $\delta \in \mathbb{R}$, coefficient of the quadratic penalty $\gamma \in \mathbb{R}$, hyperparameter controlling the annealing rate $r \in \mathbb{R}$, hyperparameter controlling the timesteps $t_0 \in \mathbb{N}$.
initialize timestep $t \leftarrow 0$, parameter vector $\theta_{t=0} \in \mathbb{R}^n$, importance weights $\Omega \leftarrow \mathbf{1}$, first moment vector $m_{t=0} \leftarrow 0$, second moment vector $v_{t=0} \leftarrow 0$, schedule multiplier $\eta_{t=0} \in \mathbb{R}$.
repeat
  $t \leftarrow t + 1$ ▷ update timestep
  $x \leftarrow$ SelectBatch($\mathbf{x}$) ▷ select batch data
  $q_{c,t} \leftarrow Q_{c,t}(x, \theta_{t-1})$ ▷ compute output logits for the current model
  $q_{p,t} \leftarrow Q_{p,t}(x, \theta^*)$ ▷ compute output logits for the previous model
  $\nabla(f_t(x; \theta_{t-1})) \leftarrow \nabla(L_{CELC}(q_{c,t}, q_{p,t})$ ▷ compute gradients
  $\Omega_t \leftarrow \Omega_{t-1}$
  for $k \leftarrow 0$ to N **do**
    $g_t(x_k) \leftarrow \nabla l_2^2(f_t(x_k; \theta_{t-1}))$
    $\Omega_t \leftarrow \Omega_t + \|g_t(x_k)\|$
  end for
  $\Omega_t \leftarrow \Omega_t / N$ ▷ compute importance weights after each update epochs
  $\lambda(t) \leftarrow 1/(1 + \exp(-r \cdot (t - t_0))$ ▷ compute annealing coefficient
  $g_t \leftarrow \lambda(t)\nabla f_t(x; \theta_{t-1}) + 2(1 - \lambda(t))\delta\gamma\Omega_t(\theta_{t-1} - \theta^*)$ ▷ compute new gradients
  $m_t \leftarrow \beta_1 m_{t-1} + (1 - \beta_1) g_t$ ▷ update biased first moment estimate
  $v_t \leftarrow \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$ ▷ update biased second raw moment estimate
  $\hat{m}_t \leftarrow m_t / (1 - \beta_1^t)$ ▷ compute bias-corrected first moment estimate
  $\hat{v}_t \leftarrow v_t / (1 - \beta_2^t)$ ▷ compute bias-corrected second raw moment estimate
  $\eta_t \leftarrow$ SetScheduleMultiplier(t) ▷ can be fixed, decay, or also be used for warm restarts
  $\theta_t \leftarrow \theta_{t-1} - \eta_t(\lambda(t)\alpha\hat{m}_t / (\sqrt{\hat{v}_t} + \epsilon) + 2(1 - \lambda(t))\delta\gamma\Omega_t(\theta_{t-1} - \theta^*))$ ▷ update parameters
until stopping criterion is met
return optimized parameters θ_t

TABLE I
DIFFERENT TASK SEQUENCES USED FOR 5-TASK CONTINUAL LEARNING SETTING.

Order	# Task Sequence
1	20news → BBC_News → IndiaPoliceEvents_sents → insightCrime → satp_relevant
2	satp_relevant → insightCrime → IndiaPoliceEvents_sents → BBC_News → 20news
3	20news → IndiaPoliceEvents_sents → BBC_News → insightCrime → satp_relevant
4	20news → IndiaPoliceEvents_sents → insightCrime → BBC_News → satp_relevant
5	satp_relevant → IndiaPoliceEvents_sents → BBC_News → insightCrime → 20news
6	satp_relevant → IndiaPoliceEvents_sents → BBC_News → 20news → insightCrime

TABLE II
DIFFERENT TASK SEQUENCES USED FOR 4-TASK CONTINUAL LEARNING SETTING.

Order	# Task Sequence
1	20news → BBC_News → IndiaPoliceEvents_sents → insightCrime
2	20news → IndiaPoliceEvents_sents → BBC_News → insightCrime
3	BBC_News → 20news → IndiaPoliceEvents_sents → insightCrime
4	BBC_News → IndiaPoliceEvents_sents → insightCrime → 20news
5	IndiaPoliceEvents_sents → BBC_News → 20news → insightCrime
6	IndiaPoliceEvents_sents → 20news → insightCrime → BBC_News
7	insightCrime → IndiaPoliceEvents_sents → 20news → BBC_News
8	insightCrime → 20news → BBC_News → IndiaPoliceEvents_sents

D. Forgetting

We calculate forgetting for the continual learning setting as shown below:

$$F = \frac{1}{T-1} \sum_{i=1}^{T-1} \max_{t \in 1,\ldots,T-1} (\text{Acc}_{t,i} - \text{Acc}_{T,i}) \tag{8}$$

where Acc means accuracy. T represents the total number of tasks. t and i represent the task index. The forgetting results are shown in Table III and Table IV.

V. EXPERIMENTS

In this section, we evaluate ConfliLPC on the Political Conflict Analysis Datasets provided in ConfliBERT [5]. We compare our model with Adam [19], EWC [7], MAS [24], SI [8], and RecAdam [25]. Multitasked is the upper bound in which we train all tasks at the same time. Our work follows task incremental learning and the task id is visible during testing.

A. Datasets

We evaluate our approach ConfliLPC on the Political Conflict Analysis Datasets, which is a collection of resources for training, evaluating, and analyzing provided in ConfliBERT [5]. It contains the following 5 different datasets:

Binary classification (BC). For identifying political news, we collected data from BBC News [26] and the 20 Newsgroups dataset [27]. These BC tasks are crucial for political scientists to classify and filter documents containing political and conflict events from large-scale news wires.

TABLE III
SUMMARY OF AVERAGED METRIC SCORES (INCLUDING FORGETTING) FOR DIFFERENT METHODS UNDER PERMUTED 5-TASK ORDERS. THE AVERAGE AND STD COLUMNS RESPECTIVELY ARE THE AVERAGE AND STANDARD DEVIATION OF THE AVERAGED SCORES FOR EACH ROW OF THE METHODS. MULTITASKED LEARNING AS AN UPPER BOUND IS SHOWN AT THE BOTTOM. ALL OF THE RESULTS ARE THE MEDIANS OVER 5 RUNS. ALL THE METRICS ARE ACC (ACCURACY) / FGT (FORGETTING). WINNING WITH THE HIGHEST ACCURACY AND THE LEAST FORGETTING.

Model	**Order 1** acc/fgt	**Order 2** acc/fgt	**Order 3** acc/fgt	**Order 4** acc/fgt	**Order 5** acc/fgt	**Order 6** acc/fgt	**Average** acc/fgt	**Std** acc/fgt
ConfliBERT + Adam $_{Median}$	56.8/31.4	44.0/47.2	59.6/27.7	37.2/39.5	51.2/38.8	58.0/30.2	51.1/35.8	8.9/7.3
ConfliBERT + EWC $_{Median}$	63.7/20.4	48.0/30.7	58.0/18.0	47.8/25.7	53.2/30.2	56.7/29.6	54.6/25.8	6.2/**5.4**
ConfliBERT + MAS $_{Median}$	66.3/23.6	48.5/35.4	60.1/20.8	52.1/29.6	51.1/**29.1**	54.1/32.7	55.4/28.5	6.6/5.5
ConfliBERT + SI $_{Median}$	65.1/26.7	47.7/40.1	59.4/23.6	49.3/33.6	49.5/33.0	55.5/35.7	54.4/32.1	6.9/6.0
ConfliBERT + RecAdam $_{Median}$	62.6/25.1	46.5/37.8	60.3/22.2	51.0/31.6	51.1/31.0	53.0/34.2	54.1/30.3	6.1/5.8
ConfliBERT + ConfliLPC $_{Median}$	**72.0/12.6**	**59.1/27.7**	**66.7/12.8**	**62.7/15.7**	**56.7/29.1**	**58.6/27.1**	**62.6/20.8**	**5.8**/7.9
Multitasked	79.8							

TABLE IV
SUMMARY OF AVERAGED METRIC SCORES (INCLUDING FORGETTING) FOR DIFFERENT METHODS UNDER PERMUTED 4-TASK ORDERS. THE AVERAGE AND STD COLUMNS RESPECTIVELY ARE THE AVERAGE AND STANDARD DEVIATION OF THE AVERAGED SCORES FOR EACH ROW OF THE METHODS. MULTITASKED LEARNING AS AN UPPER BOUND IS SHOWN AT THE BOTTOM. ALL OF THE RESULTS ARE THE MEDIANS OVER 5 RUNS. ALL THE METRICS ARE ACC (ACCURACY) / FGT (FORGETTING). WINNING WITH THE HIGHEST ACCURACY AND THE LEAST FORGETTING.

Model	**Order 1** acc/fgt	**Order 2** acc/fgt	**Order 3** acc/fgt	**Order 4** acc/fgt	**Order 5** acc/fgt	**Order 6** acc/fgt	**Order 7** acc/fgt	**Order 8** acc/fgt	**Average** acc/fgt	**Std** acc/fgt
ConfliBERT + Adam $_{Median}$	70.7/13.5	70.3/14.6	71.0/13.0	56.3/33.1	68.6/16.0	63.4/22.2	56.4/33.2	69.2/15.5	65.7/20.1	6.3/8.5
ConfliBERT + EWC $_{Median}$	72.2/12.4	69.5/21.1	74.2/8.7	56.0/33.1	71.2/13.7	62.8/24.9	57.5/31.4	69.3/15.8	66.6/20.1	6.9/9.0
ConfliBERT + MAS $_{Median}$	68.2/13.6	60.5/23.2	69.1/9.6	55.7/36.4	68.3/15.1	58.4/27.4	48.6/34.5	63.2/17.4	61.5/22.2	7.2/9.9
ConfliBERT + SI $_{Median}$	67.4/28.6	59.3/24.7	68.2/25.5	54.5/38.3	67.9/20.0	57.8/23.1	47.2/31.4	62.4/21.5	60.6/26.6	7.4/6.0
ConfliBERT + RecAdam $_{Median}$	61.9/24.9	65.7/21.5	59.1/22.2	55.1/33.3	67.7/17.4	66.1/20.1	60.1/27.3	68.0/17.8	63.0/23.1	**4.6/5.3**
ConfliBERT + ConfliLPC $_{Median}$	**78.3/3.4**	**74.7/8.6**	**78.3/4.5**	**64.1/22.1**	**77.5/5.5**	**67.7/17.1**	**60.6/26.2**	**71.7/12.7**	**71.6/12.5**	6.8/8.6
Multitasked	79.5									

TABLE V
THE RESULTS OF ABLATION STUDY ON ADAM, LOGITS CALIBRATION (LC), PARAMETER CALIBRATION (PC), AND LOGITS AND PARAMETER CALIBRATION (CONFLILPC) UNDER PERMUTED 4-TASK ORDERS. THE AVERAGE AND STD COLUMNS RESPECTIVELY ARE THE AVERAGE AND STANDARD DEVIATION OF THE AVERAGED SCORES FOR EACH ROW OF THE METHODS. MULTITASKED LEARNING AS AN UPPER BOUND IS SHOWN AT THE BOTTOM. ALL OF THE RESULTS ARE THE MEDIANS OVER 5 RUNS. ALL THE METRICS ARE ACC (ACCURACY).

Model	**Order 1** acc	**Order 2** acc	**Order 3** acc	**Order 4** acc	**Order 5** acc	**Order 6** acc	**Order 7** acc	**Order 8** acc	**Average** acc	**Std** acc
ConfliBERT + Adam (rerun) $_{Median}$	70.7	70.3	71.0	56.3	68.6	63.4	56.4	69.2	65.7	6.3
ConfliBERT + Adam + LC $_{Median}$	73.5	71.0	71.7	63.0	69.2	64.1	**67.3**	69.8	68.7	**3.7**
ConfliBERT + PC $_{Median}$	77.8	72.5	75.1	**67.6**	69.5	**68.5**	61.4	71.6	70.5	5.0
ConfliBERT + ConfliLPC $_{Median}$	**78.3**	**74.7**	**78.3**	64.1	**77.5**	67.7	60.6	**71.7**	**71.6**	6.8
Multitasked	79.5									

Multi-class classification (MCC). The India Police Events dataset [28] includes sentences from English-language articles in the Times of India about police activity in Gujarat during March 2002—a period marked by widespread Hindu-Muslim violence. The labels cover five categories of police activity: kill, arrest, fail to act, force, and any action.

Multi-label classification (MLC). The South Asia Terrorism Portal (SATP: https://satp.org/) provided a manually annotated sample of 7,445 narratives between 2011 and 2019, focusing on incidents initiated by terrorist organizations. Of these, 23.6% are relevant stories classified into one or more categories such as armed assault, bombing/explosion, kidnapping, and others, while the remainder are unrelated (stories not about terrorist attacks such as arrests or armed clashes). InSight Crime [29] contains annotated stories about organized criminal activity in Latin America and the Caribbean. We applied an MLC task to predict multiple crime categories expressed in the stories, such as drug trafficking, corruption, and law enforcement.

B. Experimental Setup

We have implemented two experimental setups: (1) a 5-task continual learning setting, and (2) a 4-task continual learning setting. In the 5-task setting, we sequentially train the model on five different tasks, in a total of 6 different orders as shown in Table I. Similarly, in the 4-task setting, we train sequentially on four tasks, in a total of 8 different orders as shown in Table II. After each training phase, we assess the performance of the model on all tasks it has been trained on to date. For instance, if the current model is trained on the task IndiaPoliceEvents_sents, having been previously trained on the 20news and BBC_News tasks, we evaluate it using instances from all three tasks. Post-training on the final task, we report the evaluation results across all tasks encountered. In these continual learning settings, our primary focus is on overall classification performance. Task-specific forgetting is

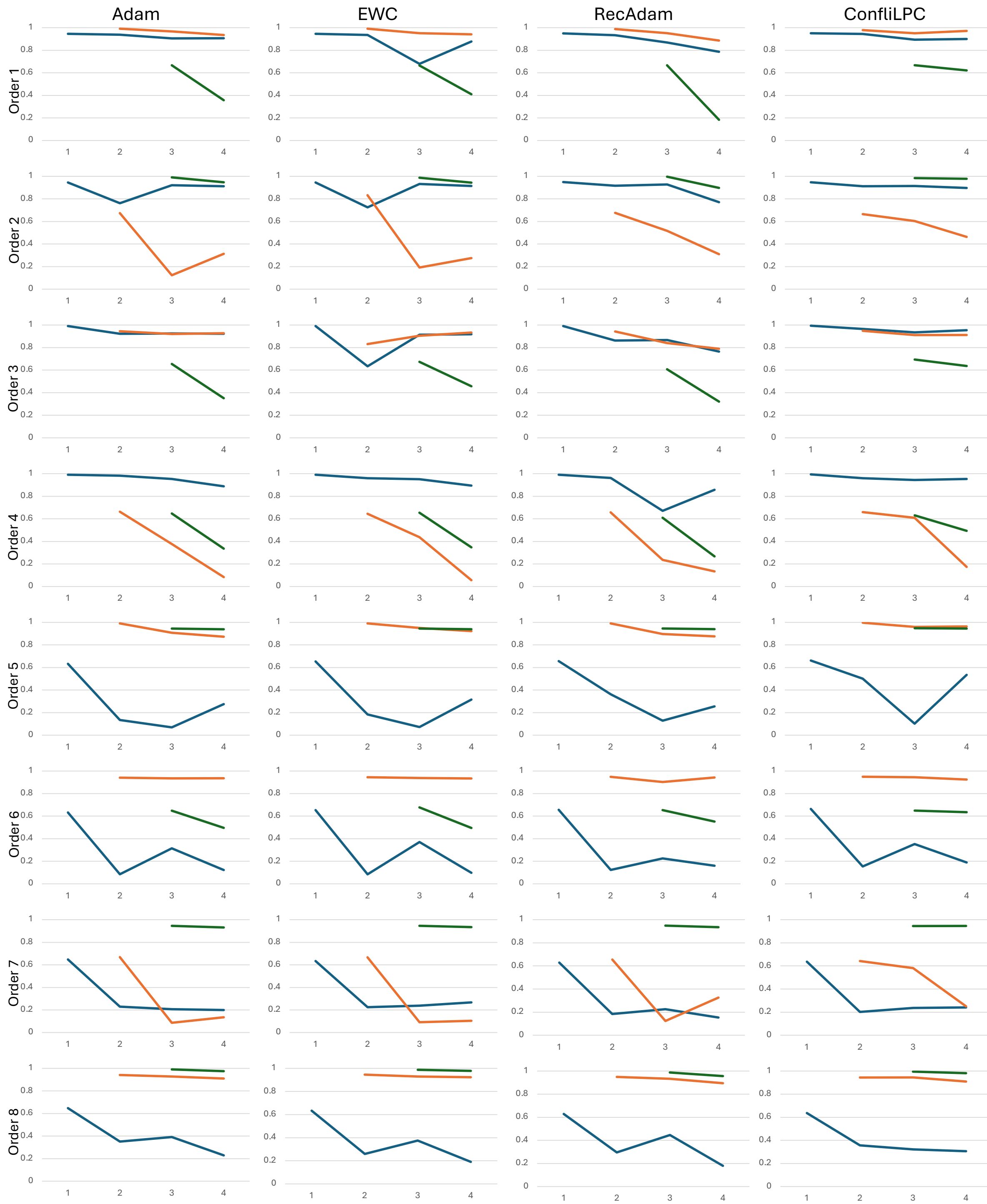

Fig. 3. Overview of the Forgetting Progress for Different Methods and Permuted Orders under Permuted 4-task Orders. The x-axis is the step and the y-axis is the metric (accuracy). The blue line indicates the scores of the first task after training each task. The orange line corresponds to that of the second task. The green line corresponds to that of the third task. The forgetting of our model ConfliLPC is on the rightmost column.

gauged by the decline in performance metrics (accuracy) from the initial to the most recent assessment of each task.

C. Results

The results are shown in Table III and Table IV. Our experimental results show that we outperform ConfliBERT with Adam, EWC, MAS, SI, and RecAdam models on all 5 scenarios of the Political Conflict Analysis Datasets, achieving the best average accuracy across all models.

Under the 5-task continual learning setting, we have 6 different task sequences to evaluate the performance of our work with Adam, EWC, MAS, SI, and RecAdam, the description of task sequences is in Table I. Under the 4-task continual learning setting, we have 8 different task sequences to evaluate the performance of our work with Adam, EWC, MAS, SI, and RecAdam, the description of task sequences is in Table II. We show the result of all the sequence orders in Table III and Table IV. From the experimental results, we can see our model achieves the least forgetting compared with Adam, EWC, MAS, SI, and RecAdam, we achieve the best average acc on both the 5-task setting and 4-task setting. Compared to the upper-bound Multitasked, our highest score even approximates Multitasked with 1.2%. In addition, the sequence order of tasks also matters in the experiment results because when we put more complex tasks late in the order, the average scores will be higher as the forgetting of simple tasks will be less. The results of the continual learning setting show that our method can achieve the best performance and forget less than other methods, which demonstrates the effectiveness of our method in addressing the CF problem in continual learning.

D. Ablation Study

ConfliLPC has two important components, Logits Calibration (LC) and Parameter Calibration (PC). Thus, we do an ablation study on these two components separately with ConfliBERT pre-trained model 5 scenarios of the Political Conflict Analysis Datasets. The results of the ablation study with ConfliBERT model are shown in Table V. From Table V, we can see both LC and PC achieve better results than the baseline Adam. ConfliLPC achieves the best results among all four models In most orders. Compared with Adam, LC achieves 4.6% improvements on average measured by acc on all 8 different orders. Compared with Adam, PC achieves 7.3% improvements on average measured by acc on all 8 different orders. Compared with Adam, ConfliLPC achieves 9.0% improvements on average measured by acc on all 8 different orders. The ablation results demonstrate the importance of both Logit and Parameter Calibration in ConfliLPC.

VI. Forgetting Analysis for Four Tasks

ConfliPLC has the lowest forgetting in all 8 different orders. Forgetting is analyzed by assessing the performance decrease following each new task, as illustrated in Fig. 3. The forgetting of our model ConfliLPC is on the rightmost column. The figure highlights that our model consistently exhibits the lowest level of forgetting in most cases, showcasing the stability and plasticity of the proposed ConfliLPC model.

VII. Related Works

ConfliBERT [5] built on previous automating analysis of political conflict texts by pre-training specifically on a curated corpus, significantly enhancing model performance compared to general-purpose models like BERT [17]. However, ConfliBERT utilized in a fine-tuning scenario where one model is trained for a single task. In a continual learning setting, where data arrives sequentially, we minimize the number of parameters stored and train a single model sequentially across all tasks, developing a more generalized model. Yet, deep neural networks engaging in continual learning often encounter a problem known as *catastrophic forgetting*. Minimizing CF is key to this goal. Current strategies in continual learning are grouped into three main categories: (1) Replay methods [30], [31], (2) Regularization-based methods [4], [24], [32]–[35], and (3) Parameter isolation methods [36], [37]. Replay methods either preserve actual samples or generate synthetic ones using a model. Regularization-based approaches avoid storing direct inputs to protect privacy and reduce memory demands. These methods introduce an additional regularization term in the loss function to help retain prior knowledge when assimilating new data. Parameter isolation methods allocate distinct parameters for each task to prevent forgetting. Additionally, semi-supervised methods in continual learning [38] offer further enhancements. Our approach utilizes an advanced regularization-based method incorporating LCwoF [21] and RecAdam [25]. LCwoF modifies the traditional cross-entropy loss by adding an exponential logit sum from prior class classifiers to the denominator to adjust the normalization scale, although its normalization component has inaccuracies since its total does not sum to one. RecAdam, building on EWC, treats all parameters uniformly, neglecting the varying significance of different network parameters.

VIII. Conclusion

Integrating Logits and Parameter Calibration (LPC) with the ConfliBERT model in the ConfliLPC framework is a significant advancement in applying NLP to analyzing political conflict and violence. Addressing the challenges associated with continual learning, ConfliLPC ensures the model remains adaptive and effective across diverse and evolving scenarios. This capability is crucial to maintain accuracy and less forgetting, allowing the system to process new data streams without losing the valuable context of historical data. The model's ability to fine-tune and recalibrate dynamically in response to emerging data ensures that insights remain relevant and actionable. Looking ahead, there is potential for expanding the application of ConfliLPC to multilingual settings and exploring more complex analytical tasks such as inference and question answering.

Acknowledgement

This research was supported by NSF award OAC-2311142.

REFERENCES

[1] K. Raik, "The ukraine crisis as a conflict over europe's political, economic and security order," *Geopolitics*, vol. 24, no. 1, pp. 51–70, 2019.

[2] R. Caruana, "Multitask learning," *Machine Learning*, vol. 28, pp. 41–75, 1997.

[3] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural Networks*, vol. 113, pp. 54–71, 2019.

[4] X. Li, Z. Wang, D. Li, L. Khan, and B. Thuraisingham, "Lpc: A logits and parameter calibration framework for continual learning," in *Findings of the Association for Computational Linguistics: EMNLP 2022*, 2022, pp. 7142–7155.

[5] Y. Hu, M. Hosseini, E. Skorupa Parolin, J. Osorio, L. Khan, P. Brandt, and V. D'Orazio, "Conflibert: A pre-trained language model for political conflict and violence." Association for Computational Linguistics, 2022.

[6] S. Thrun and L. Pratt, *Learning to learn*. Springer Science & Business Media, 1998.

[7] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the national academy of sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[8] F. Zenke, B. Poole, and S. Ganguli, "Continual learning through synaptic intelligence," in *International Conference on Machine Learning*. PMLR, 2017, pp. 3987–3995.

[9] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell, "Progressive neural networks," *arXiv preprint arXiv:1606.04671*, 2016.

[10] A. Robins, "Catastrophic forgetting, rehearsal and pseudorehearsal," *Connection Science*, vol. 7, no. 2, pp. 123–146, 1995.

[11] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," *arXiv preprint arXiv:1703.03400*, 2017.

[12] G. Palmer, R. W. McManus, V. D'Orazio, M. R. Kenwick, M. Karstens, C. Bloch, N. Dietrich, K. Kahn, K. Ritter, and M. J. Soules, "The mid5 dataset, 2011–2014: Procedures, coding rules, and description," *Conflict Management and Peace Science*, vol. 39, no. 4, pp. 470–482, 2022.

[13] H. Hegre, P. Vesco, and M. Colaresi, "Lessons from an escalation prediction competition," *International Interactions*, vol. 48, no. 4, pp. 521–554, 2022.

[14] J. A. Goldstone, R. H. Bates, D. L. Epstein, T. R. Gurr, M. B. Lustik, M. G. Marshall, J. Ulfelder, and M. Woodward, "A global model for forecasting political instability," *American journal of political science*, vol. 54, no. 1, pp. 190–208, 2010.

[15] C. Raleigh, R. Linke, H. Hegre, and J. Karlsen, "Introducing acled: An armed conflict location and event dataset," *Journal of peace research*, vol. 47, no. 5, pp. 651–660, 2010.

[16] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[17] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[18] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," *Advances in neural information processing systems*, vol. 32, 2019.

[19] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[20] Z. Zhang and M. R. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," in *32nd Conference on Neural Information Processing Systems (NeurIPS)*, 2018.

[21] A. Kukleva, H. Kuehne, and B. Schiele, "Generalized and incremental few-shot learning by explicit learning and calibration without forgetting," *arXiv preprint arXiv:2108.08165*, 2021.

[22] E. Kiperwasser and M. Ballesteros, "Scheduled multi-task learning: From syntax to translation," *Transactions of the Association for Computational Linguistics*, vol. 6, pp. 225–240, 2018.

[23] J. Martens, "New insights and perspectives on the natural gradient method," *arXiv preprint arXiv:1412.1193*, 2014.

[24] R. Aljundi, F. Babiloni, M. Elhoseiny, M. Rohrbach, and T. Tuytelaars, "Memory aware synapses: Learning what (not) to forget," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 139–154.

[25] S. Chen, Y. Hou, Y. Cui, W. Che, T. Liu, and X. Yu, "Recall and learn: Fine-tuning deep pretrained language models with less forgetting," *arXiv preprint arXiv:2004.12651*, 2020.

[26] D. Greene and P. Cunningham, "Practical solutions to the problem of diagonal dominance in kernel document clustering," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 377–384.

[27] K. Lang, "Newsweeder: Learning to filter netnews," in *Machine learning proceedings 1995*. Elsevier, 1995, pp. 331–339.

[28] A. Halterman, K. A. Keith, S. M. Sarwar, and B. O'Connor, "Corpus-level evaluation for event qa: The indiapoliceevents corpus covering the 2002 gujarat violence," *arXiv preprint arXiv:2105.12936*, 2021.

[29] E. S. Parolin, L. Khan, J. Osorio, P. T. Brandt, V. D'Orazio, and J. Holmes, "3m-transformers for event coding on organized crime domain," in *2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, 2021, pp. 1–10.

[30] D. Lopez-Paz and M. Ranzato, "Gradient episodic memory for continual learning," *Advances in neural information processing systems*, vol. 30, pp. 6467–6476, 2017.

[31] A. Chaudhry, M. Ranzato, M. Rohrbach, and M. Elhoseiny, "Efficient lifelong learning with a-gem," *arXiv preprint arXiv:1812.00420*, 2018.

[32] Z. Wang, Y. Chen, C. Zhao, Y. Lin, X. Zhao, H. Tao, Y. Wang, and L. Khan, "Clear: Contrastive-prototype learning with drift estimation for resource constrained stream mining," in *Proceedings of the Web Conference 2021*, 2021, pp. 1351–1362.

[33] H. Yin, D. Li, and P. Li, "Continual learning for natural language generations with transformer calibration," in *The SIGNLL Conference on Computational Natural Language Learning*, 2022.

[34] D. Li, Z. Chen, E. Cho, J. Hao, X. Liu, F. Xing, C. Guo, and Y. Liu, "Overcoming catastrophic forgetting during domain adaptation of seq2seq language generation," in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2022, pp. 5441–5454.

[35] Z. Wang, D. Li, and P. Li, "Latent coreset sampling based data-free continual learning," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022, pp. 2077–2087.

[36] A. Mallya and S. Lazebnik, "Packnet: Adding multiple tasks to a single network by iterative pruning," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 7765–7773.

[37] J. Serra, D. Suris, M. Miron, and A. Karatzoglou, "Overcoming catastrophic forgetting with hard attention to the task," in *International Conference on Machine Learning*. PMLR, 2018, pp. 4548–4557.

[38] X. Li, L. Khan, M. Zamani, S. Wickramasuriya, K. W. Hamlen, and B. Thuraisingham, "Mcom: A semi-supervised method for imbalanced tabular security data," in *IFIP Annual Conference on Data and Applications Security and Privacy*. Springer, 2022, pp. 48–67.