Continually Detection, Rapidly React: Unseen Rumors Detection based on Continual Prompt-Tuning

Yuhui Zuo[1], Wei Zhu[2], Guoyong Cai[1],
[1]Guilin University of Electronic Technology, Guilin, China
[2]East China Normal University, Shanghai, China
`zzuuoo2019ok@gmail.com,ccgycai@guet.edu.cn,wzhu@stu.ecnu.edu.cn`

Abstract

Since open social platforms allow for a large and continuous flow of unverified information, rumors can emerge unexpectedly and spread quickly. However, existing rumor detection (RD) models often assume the same training and testing distributions and can not cope with the continuously changing social network environment. This paper proposed a Continual Prompt-Tuning RD (CPT-RD) framework, which avoids catastrophic forgetting (CF) of upstream tasks during sequential task learning and enables bidirectional knowledge transfer between domain tasks. Specifically, we propose the following strategies: (a) Our design explicitly decouples shared and domain-specific knowledge, thus reducing the interference among different domains during optimization; (b) Several technologies aim to transfer knowledge of upstream tasks to deal with emergencies; (c) A task-conditioned prompt-wise hypernetwork (TPHNet) is used to consolidate past domains. In addition, CPT-RD avoids CF without the necessity of a rehearsal buffer. Finally, CPT-RD is evaluated on English and Chinese RD datasets and is effective and efficient compared to prior state-of-the-art methods. [1]

1 Introduction

Online platforms such as social media are facing new and ever-evolving cyber threats at the information level — rumor. A rumor is an unconfirmed claim related to an object, event, or issue of public concern that is spread when its integrity is unknown (Guo et al., 2020). It is very necessary to study automated RD, because rumors are extremely harmful to society and manual detection is time-consuming and labor-intensive (Oshikawa et al., 2018).

However, automated RD has significant challenges and still faces the following difficulties:
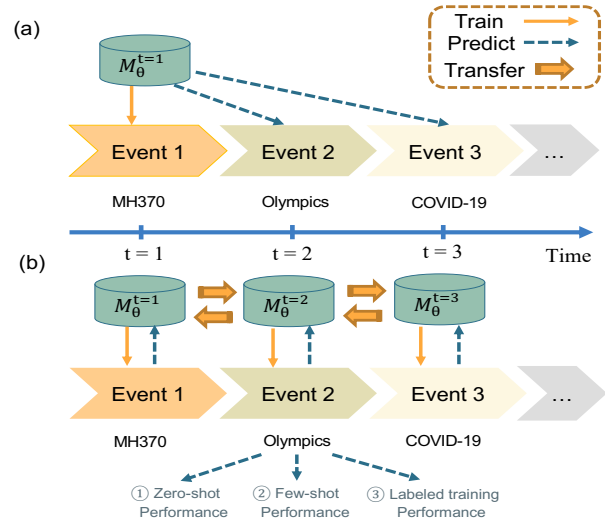

Figure 1: Illustration of the RD model training procedure in the traditional static (a) and continual dynamic event transfer (b) setup.

First, rumors are highly event domain-specific (Wang et al., 2018), each event domain may have a different input text distribution and fraudulent intent. Second, detecting rumors at their early stage of spreading faces the problem of insufficient labeled samples (Zhou et al., 2020). Third, rumor detectors operating on online social platforms often encounter continuous event domain changes. This poses a significant challenge to existing RD models.

Previous work (Wang et al., 2018; Bian et al., 2020; Zhang et al., 2021; Lin et al., 2021; Ben-David et al., 2021) usually assume the same distribution of training and testing data, and have difficulty coping with changing social network environments. In Fig.1 (a), there are no updates to the model regardless of how many unseen events will appear in the future, which requires the unreasonable assumption that the model's generalization capability is enough. In addition, as the social network environment changes, the original training set data will become outdated (Lee et al., 2021), so that the model's ability on more recent

---

[1]Innovation Project of Guangxi Graduate Education

events will diminish. For example, the COVID-19 pandemic caused massive trouble for existing RD models (Lu et al., 2021; Patwa et al., 2021b).

In social media, the ideal RD model must continuously detect event stream and react rapidly to emergencies. To study this ability, we propose a continual dynamic event transfer (CDET) setup (illustrated in Fig.1 (b)), which makes the RD model dynamically updates the parameters θ over a series of sequential events and assume that each event goes through three stages from burst to normalization: zero-shot stage with no samples, few-shot stage with a small number of samples, and full-shot stage with large-scale labeled samples. This is because rumors must be detected early to avoid the social harm caused by their spread, but there are only a few or no labeled samples in the early stages of an event.

Existing RD models still face many challenges in the CDET setup: (1) **Catastrophic forgetting:** When a neural model is trained in a sequence of tasks, the downstream tasks may catastrophically interfere with the upstream tasks. (2) **Knowledge transfer and accumulation:** Transfer the knowledge learned from upstream tasks for rapid generalization, accumulating knowledge from downstream tasks to better cope with upstream tasks; (3) **Parameter explosion:** Previous research (Wang et al., 2021a; Ke et al., 2021a) often require dynamically expanding neural modules for each task, which is undoubtedly aggravating for pre-trained language models (PLM) with billions of parameters, with limited memory; (4) **Data privacy:** After learning a task, training data is usually discarded due to user privacy concerns (Chen and Liu, 2018). This requires models to share learned parameters, rather than saving data to retrain the model.

To address the above challenges, a novel framework called **C**ontinual **P**rompt-**t**uning RD (CPT-RD) has been proposed. Technically, CPT-RD can be seen as a continuously migrated version of P-tuning v2 (Liu et al., 2021b). From Fig.2, we can clearly decouple domain-specific knowledge (tuning parameter) and shared knowledge (frozen parameter). This provides the basis for achieving memory of task-specific parameters and bidirectional domain knowledge transfer. Bidirectional knowledge transfer includes: (1) **Forward Knowledge Transfer**: CPT-RD has various prompt initialization strategies to adapt fast
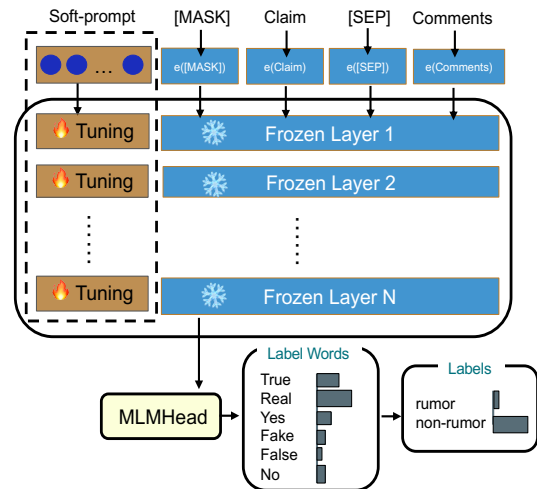

Figure 2: Illustration of deep prompt tuning for RD.

to rumors of emergencies. (2) **Backward Knowledge Transfer**: A task-conditioned prompt-wise hypernetwork (TPHNet) learns latent distribution of soft-prompts, encouraging CPT-RD to accumulate knowledge in sequential events while avoiding data replay.

We collected RD datasets with 14 different domains for both, English and Chinese. Through empirical analysis, we find that CPT-RD essentially avoids catastrophic forgetting. On the English dataset, the knowledge transfer indicators FWT and BWT (Lopez-Paz and Ranzato, 2017) achieve positive indicators of 23.9% and 0.9%, respectively. Finally, the effectiveness of our improvement is demonstrated through ablation experiments.

Our main contributions are: (1) We propose a CDET setup in RD for evaluating rapid generalization and continual detection problems simultaneously. (2) To completely avoid the CF encountered in continual detection, we optimize and store domain-specific soft-prompt for each event domain and use it selectively. (3) We propose various forward knowledge transfer strategies to deal with early emergency rumors and accumulate detection experience through TPHNet for backward knowledge transfer. (4) Our experiments on the collected Chinese and English social media RD datasets demonstrate the superior performance and efficiency of our proposed method.

2 Related work

2.1 Rumors detection

Social media has gained much attention as a source of research rumors, but existing RD meth-

ods still perform poorly in the face of unknown events and struggle to consistently respond to the dynamic and changing social network environment (Zubiaga et al., 2018; Guo et al., 2020). Wang et al. (2018) proposed an event adversarial neural network (EANN) which extracts event-invariant features by removing event-specific features. Lu et al. (2021) consider unseen event RD as a few-shot learning problem, and Wang et al. (2021b) apply meta-learning to obtain the optimal generalization initial parameters. Ben-David et al. (2021) used prompt-tuning to consider RD as domain adaptation and committed to improving out-of-distribution issues. The above methods can alleviate the problem of domain drift, but they are all learned with offline static settings. Lee et al. (2021) considered the continual detection capability on online social media and applied rehearsal-based replay techniques to resist CF. However, this approach does not consider generalizing to the unseen domain quickly and faces problems such as inference efficiency and data privacy. Different from the above methods, our proposed model considers both fast generalization and knowledge transfer for continual detection and is not based on any data replay.

2.2 Prompt-based tuning

Recent research has found that converting downstream tasks to language modeling tasks via textual prompts is more effective to use PLM than typical fine-tuning (Liu et al., 2021a). Early prompting method, GPT-3 (Brown et al., 2020) and PET/iPET (Schick and Schütze, 2020) for example, uses hand-crafted prompt templates. However, the performance of these methods relies heavily on the selection of predefined prompt templates. Hand-crafting prompts are very time-consuming, and the performance may be sub-optimal. Shin et al. (2020) propose AutoPrompt to search for better prompts based on gradient descent approach. Instead of searching for discrete template words, Li and Liang (2021) propose prefix-tuning, where tokens with trainable continuous embeddings are placed at the beginning of the text to perform generate tasks. P-tuning v2 (Liu et al., 2021b) also uses soft-prompt to achieve promising natural language understanding and knowledge probing tasks. Different from the above methods, they studied single-step adaptation, and we are interested in prompt transfer in CL environment.

2.3 Continual learning with fast generalization

Mitigating CF is usually a priority in CL or lifelong learning research proposals (Hadsell et al., 2020). Recently, the demands on CL have increased further, not only to combat CF but also to generalize quickly in unseen tasks. Integration with meta-learning is a promising approach, concerned with balancing stability (preservation of past knowledge) and plasticity (rapid absorption of current knowledge). MER (Riemer et al., 2018) achieves gradient alignment by constraining the direction of the gradient angle between different task samples. Subsequently, OML (Javed and White, 2019) and La-MAML (Gupta et al., 2020) optimize and supplemente the training speed and effect of MER, respectively. Meta-MbPA (Wang et al., 2020) combine the above meta-learning, rehearsal-based replay of CL and BERT. Wang et al. (2021a) uses global and local memory networks to capture different classes of cross-task prototype representations, adds a new frozen classification module for each task, and requires BERT to update slowly. Ke et al. (2021b) and Jin et al. (2021) freezes part of the backbone model against CF. They introduce an additional adapter layer (Houlsby et al., 2019) to learn task-specific knowledge, avoiding inefficient data replay and reducing parameter tuning and growth rates. Different from the above methods, CPT-RD is based on PT. Although it is similar to adapter in terms of parameter tuning, PT is more effective in parameters, and CPT-RD has a more intuitive means of knowledge transfer and avoids data replay.

3 Methodology

In this section, we first present the task definition and execution process of CPT-RD in CDET, then elaborate prompt encoding method and bidirectional knowledge transfer strategy.

3.1 Task definition and description

Suppose RD model $\mathcal{M}_{1:k-1}$ has performed learning on a sequence of tasks from 1 to $k-1$, denoted as $\mathcal{T}_{1:k-1} = \{\mathcal{T}_1, \ldots, \mathcal{T}_{k-1}\}$. Each task is a domain-specific rumor binary classification problem (non-rumor or rumor), and the inputs are claim and comments texts. The goal of the RD model is to use the knowledge gained from upstream $\{\mathcal{T}_{1:k-1}\}$ tasks to help learn a better detector $\mathcal{M}_{1:k}$ for the k-th task $\mathcal{T}_k$ while avoiding
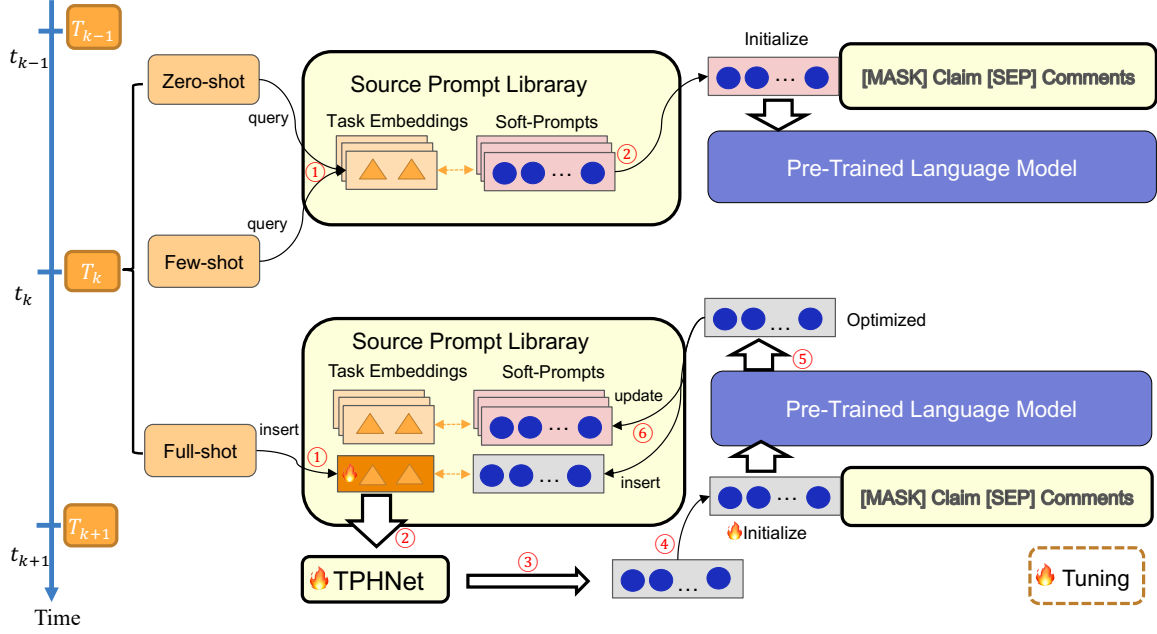
Figure 3: Illustration of the execution of CPT-RD in a single task in CDET.

forgetting knowledge learned from past tasks. We use the terms domain and task interchangeably because each task is from a different domain.

Often, each task consists of three stages: zero-shot stage, few-shot stage, and full-shot (large-scale labeled training) stage, which corresponds to the development process of rumors from germination to normalization as described in Sec.1. As shown in Fig.3, in the zero-shot and few-shot stages, CPT-RD applies various forward knowledge transfer strategies proposed in Sec.3.3 to query the soft-prompt as the prompt initialization of the current task. Note that the soft-prompt of the current task is randomly initialized if the source prompt library (SPL) is empty. In the full-shot phase of the current task, CPT-RD starts training with a randomly initialized soft-prompt instead of using the soft-prompt in SPL because, in experiments, we found that the former works better. CPT-RD obtains knowledge accumulation in full-shot annotated samples through TPHNet. The optimized soft-prompt specific to the current task is stored in the SPL at the end of full-shot training.

3.2 Prompt encoding

Prompt tuning (PT) formalizes RD as a masked language modeling problem using a pre-trained language model (PLM). In Fig.2, given the k-th task input, the pretrained embedding layer e of PLM converts the claim and comments text into token embeddings $\mathbf{X}_k = e(\text{claim}) \in \mathbb{R}^{m \times d}$ and

$\mathbf{C}_k = e(\text{comments}) \in \mathbb{R}^{n \times d}$, m, n is the token length, and d is the hidden dimension of PLM. We prepend l randomly initialized soft-prompt tokens $\mathbf{P}_k = \{P_k^1, P_k^2, \ldots P_k^l\} \in \mathbb{R}^{l \times d}$ before them, where $P_k^i \in \mathbb{R}^d$ is an embedding vector. Then, we add a [MASK] token, which is used to predict the label words $y \in \mathcal{Y}$. The input embedding of PLM is:

$$\mathbf{x} = \mathbf{P}_k, [\text{MASK}], \mathbf{X}_k, [\text{SEP}], \mathbf{C}_k, \qquad (1)$$

and only $\mathbf{P}_k$ is learnable. Hence the tuned parameters in PT are extremely fewer than full-parameter fine-tuning, which is friendly for model deployment.

One key ingredient of PT is the *verbalizer*: a mapping from the class label to a word token in the PLM vocabulary. PLM gives the probability of each word v in the vocabulary being filled in [MASK] token $p([\text{MASK}] = v|\mathbf{x})$. To map the probabilities of words into the probabilities of labels, we define the *verbalizer* as ver, which form the label word set $\mathcal{V}$, to the label space $\mathcal{Y}$, i.e., $ver: \mathcal{V} \mapsto \mathcal{Y}$. We use $\mathcal{V}_y$ to denote the subset of $\mathcal{V}$ that is mapped into a specific label y, $\cup_{y \in \mathcal{Y}} \mathcal{V}_y = \mathcal{V}$. Then the probability of label y, i.e., $p(y|\mathbf{x})$, is calculated as:

$$p(y|\mathbf{x}) = f_{ver}\left(p([\text{MASK}] = v|\mathbf{x})|v \in \mathcal{V}_y\right), \quad (2)$$

where f_{ver} is a function transforming the probability of label words into the probability of the label.

The training objective is:

$$L_{\mathbf{P}_k} = -\sum_{\mathbf{x}_i, y_i \in D_{\mathcal{T}_k}} \log(p(y_i|\mathbf{x}_i)), \qquad (3)$$

where $D_{\mathcal{T}_k}$ is defined as the training data for task $\mathcal{T}_k$.

PT only inserts soft-prompt into the input embedding sequence of PLM. We follow the deep prompt tuning of P-tuning v2, adding the prompts of different layers as prefix tokens to the input sequence independently of other layers to increase tunable task-specific parameters and improve stability. In experiment Sec.4.3, CPT-RD evaluates the above two soft-prompt placement methods respectively.

3.3 Forward knowledge transfer

Due to the lack of samples or even no training samples (zero-shot and few-shot) in the early stage of rumor events, it is necessary to forward transfer the knowledge from upstream tasks for fast detection. An intuitive way of knowledge transfer is to reuse knowledge gained from previous tasks, which often improves and accelerates learning for future tasks.

Therefore, SPL is responsible for storing upstream soft-prompts $\{\mathbf{P}_j\}_{j<k}$ and task embeddings $\{\mathbf{z}_j\}_{j<k}$, where $\mathbf{P}_j$ is obtained after training on $D_{\mathcal{T}_k}$ in the full-shot stage, $\mathbf{z}_j = \frac{1}{|D_{\mathcal{T}_k}|} \sum_{i=1}^{|D_{\mathcal{T}_k}|} f_e(\mathbf{X}_k^i, \mathbf{C}_k^i)$, f_e is an encoder model, i.e. BERT. Based on SPL, three types of prompt initialization are proposed.

CLInit: Use the previous task's soft-prompt $\mathbf{P}_{k-1}$ to initialize the current task's soft prompt $\mathbf{P}_k$. **SimInit:** Select $\mathbf{P}_k$ from $\{\mathbf{P}_j\}_{j<k}$ with the highest similarity to the current task representation $\mathbf{z}_k \in \mathbb{R}^d$ for initialization. Note that $\mathbf{z}_k$ is calculated based on the training data of task $\mathcal{T}_k$, which does not need to be labeled. Straightforwardly, we compute euclidean distances E and cosine similarities C for task embedding pairs in the two groups and use the averaged results as the final similarity metrics:

$$
\begin{aligned}
\mathrm{E}(\mathbf{z}_k, \mathbf{z}_j) &= \frac{1}{1 + \|\mathbf{z}_k - \mathbf{z}_j\|}, \\
\mathrm{C}(\mathbf{z}_k, \mathbf{z}_j) &= \frac{\mathbf{z}_k \cdot \mathbf{z}_j}{\|\mathbf{z}_k\|\|\mathbf{z}_j\|}.
\end{aligned}
\qquad (4)
$$

When CPT-RD faces the upstream task domain samples again, it directly reuses the soft-prompt of the corresponding task, thus completely avoiding the occurrence of CF, but CLInit and SimInit are single-source transfer strategies.

Single-source reuse initialization strategy can completely avoid CF, but only a single task is considered in the forward knowledge transfer. Intuitively, knowledge available for transfer should be present in all upstream tasks. **MeanInit:** Calculate the average of $\{\mathbf{P}_j^u\}_{j<k}$ to obtain $\mathbf{P}_k$. In deep prompt tuning, each layer of soft-prompt is correspondingly averaged. MeanInit considers multi-source transfer, but none of the above strategies can accumulate to transfer backward knowledge. We empirically compare these three strategies in Sec.4.3.

3.4 TPHNet for backward knowledge transfer

Although storing the training-optimized $\mathbf{P}_k$ after labeled train on task $\mathcal{T}_k$ can avoid forgetting, it ignores the backward knowledge transfer of the tasks. Hypernetwork is usually used to consolidate the knowledge of sequence tasks and has a certain ability of backward knowledge transfer (von Oswald et al., 2019; KJ and N Balasubramanian, 2020; Jin et al., 2021). It is a network that explores the meta-parameter space of another network. Similar to Hypernetwork, instead of modeling the final classification result, task-conditioned prompt-wise hypernetwork (TPHNet) learns a latent distribution space of soft-prompt with task-specific priors, aiming to accumulate knowledge in sequential and use it for all future tasks.

Specifically, when the CPT-RD completed the learning of task $\mathcal{T}_{k-1}$, and before the full-shot phase of $\mathcal{T}_k$ starts, the current task embedding $\mathbf{z}_k$ and the soft-prompt set $\{\mathbf{P}_j\}_{j<k}$ already exist in SPL as described in Sec.3.3. Here, the task representation $\mathbf{z}_k$ for task $\mathcal{T}_k$ is optimized jointly while learning the task. Taking the PT case as an example, adding soft-prompt to the embedding layer, deep prompt tuning only needs to modify the generation dimension of soft-prompt. TPHNet g generates a soft-prompt $\mathbf{P}_k$ through an auto-encoder, using $\mathbf{z}_k$ as input:

$$g(\mathbf{z}_k) = W_2(\tanh(W_1\mathbf{z}_k + b_1)) + b_2, \quad (5)$$

where $W_1 \in \mathbb{R}^{d' \times d}, b_1 \in \mathbb{R}^{d'}, W_2 \in \mathbb{R}^{(l \times d) \times d'}$ and $b_2 \in \mathbb{R}^{l \times d'}$ are trainable parameters, $d' = 64$ is the middle dimension.

Then, in each step of learning $\mathcal{T}_k^i$, we randomly sample a prior task soft-prompt $\mathbf{P}_j(j < k)$ to regu-

larize the TPHNet learning. It penalizes the ℓ_2 distance between the soft-prompt generated at the current step $\mathbf{P}_k^i = g(\mathbf{z}_k^i)$ and the pre-computed one, *i.e.*, $||\mathbf{P}_k^i - \mathbf{P}_j||_2^2$. Therefore, we avoid the TPHNet changes its output for a prior task too much during the sequential task learning, so that the knowledge accumulation is better guaranteed for the learned model. The following overall loss function:

$$ L_{\mathcal{T}_k} = L_{\hat{\mathbf{P}}_k} + \frac{\beta}{k-1} \sum_{i=1}^{k-1} ||\mathbf{P}_k^i - \mathbf{P}_j||_2^2, \quad (6) $$

where $L_{\hat{\mathbf{P}}_k}$ is the cross entropy like Eq.3. According to this equation, optimizing the overall loss will update the patameters of the TPHNet g, task embedding $\mathbf{z}_k$ and prompt-based PLM. $\beta = 0.01$ is a hyperparameter that controls the strength of the regularizer. After training, if $\mathbf{P}_k$ obtains better (or the same) performance than $\mathbf{P}_j$ on $\mathcal{T}_j$, we update $\mathbf{P}_j$ to $\mathbf{P}_k$.

4 Experiments

To evaluate the proposed CPT-RD, we closely follow the settings proposed in prior works (Lopez-Paz and Ranzato, 2017; Jin et al., 2021), and conduct comprehensive experiments. In particular, we mainly consider whether CPT-RD effectively addresses the four challenges mentioned in Sec.1. We carefully compare CPT-RD with state-of-the-art methods of different categories under proper experiment settings. Moreover, we conduct extensive ablation studies to provide a deeper understanding of our method.

4.1 Datasets and evaluation metrics

We collected 14 domain events for each dataset, Chinese and English. Each piece of data contains a claim, comments, and label (non-rumor or rumor). We split the data for each event into train/validation/test datasets with a split ratio of 30%/35%/35%. The details about the statistics are shown in Table 6, and the PHEME and Weibo (Lu et al., 2021) datasets are well divided by event domain. We also added the COVID-19 dataset (Patwa et al., 2021a) for the English dataset, which comes from the competition[2]. In addition, for datasets without split of events, including Twitter15 and Twitter16 (Ma et al., 2018), we use Tweeter-LDA (Diao et al., 2012), an LDA variant widely used for short and noisy tweets, to determine topic clusters as well as important words

[2]https://competitions.codalab.org/competitions/26655.

with their weights. We removed the label of unverified rumors in Twitter15 and Twitter16, retained the rumors and non-rumor, and divided them into five-event domains. The similarity between the datasets we calculated using TF-IDF is shown in Fig.5. To prepare the model inputs for both datasets, first, we replace the URLs with the special token [unused10]". Then, we also replace the usernames with the special token [unused11]". These two datasets will be released on GitHub along with our experimental code.

To evaluate the performance of RD using F1 score as a metric, $R_{j,i}$ is defined as the F1 score on the test set of task $\mathcal{T}_i$ after training on task $\mathcal{T}_j$. We follow the two indicators, FWT and BWT, proposed by Lopez-Paz and Ranzato (2017) to evaluate the knowledge transfer ability of CPT-RD in the process of continual learning. We evaluate the average F1 performance for all tasks after full-shot training on the final task $\mathcal{T}_N$:

$$ \text{Avg.F1} = \frac{1}{N} \sum_{i=1}^{N} R_{T,i}. \quad (7) $$

where N is the number of tasks. According to Lopez-Paz and Ranzato (2017), two metrics are defined to measure the effect of forward and backward transfers:

$$ \text{BWT} = \frac{1}{N-1} \sum_{i=1}^{N-1} R_{N,i} - R_{i,i}, \quad (8) $$

$$ \text{FWT} = \frac{1}{N-1} \sum_{i=2}^{N} R_{i-1,i} - R_{0,i}. \quad (9) $$

FWT is the average zero-shot performance on a new task and evaluating the model's generalization ability. BWT assesses the impact of learning on the subsequent task has on the previous task. A negative BWT indicates that the model has forgotten some previously acquired knowledge.

In addition, to evaluate the models performance early in the birth of the rumor, we also recorded the average few-shot performance of each new task during the continual cumulative training, which is a cumulative value:

$$ \text{fs.F1} = \frac{1}{N} \sum_{i=1}^{N} R_{i,i}^{fs}, \quad (10) $$

where $R_{i,i}^{fs}$ denotes the F1 performance of training on the few-shot training set of the i-th task and testing on the test set of the i-th task.

Method	PHEME + Twitter15&16 + Covid19				Weibo					
	Avg.F1	FWT	BWT	fs.F1(val)	Avg.F1	FWT	BWT	fs.F1(val)	+Params	Tune Params
Fine-tuning†	54.6 (± 1.2)	7.0	-24.3	67.5	45.6 (±3.1)	18.3	-23.2	75.6	0	100%
EANN†	57.9 (± 1.5)	8.1	-20.3	68.3	49.4 (±2.1)	18.5	-20.9	77.1	1.8%	100%
Adapter⋆	66.7 (± 2.5)	13.7	-14.2	74.6	58.0 (±2.3)	24.4	-17.5	83.9	2.1%	2.1%
ParallelAdapter⋆	61.5 (± 2.2)	2.8	**-4.7**	68.7	55.8 (±2.5)	8.5	-7.1	75.3	2.3%	2.3%
prompt-tuning (CLS)⋆	60.0 (± 1.5)	7.4	-15.7	75.9	67.8 (±1.8)	18.3	-5.1	84.1	0.03%	0.03%
p-tuning v2 (CLS)⋆	62.9 (± 2.1)	5.1	-9.3	73.0	67.6 (±1.6)	24.4	-3.0	80.4	0.6%	0.6%
prompt-tuning (VER)⋆	65.0 (± 1.5)	7.1	-10.2	75.7	70.5 (±1.8)	18.7	-7.5	82.8	0.03%	0.03%
p-tuning v2 (VER)⋆	**69.7 (± 1.1)**	13.9	-11.0	**76.3**	**71.8 (±1.6)**	29.5	-6.8	82.6	0.6%	0.6%
Prompt-tuning based (PT-based)										
CPT-RD (CLInit)⋆	65.1 (± 0.8)	15.0	0	75.5	70.0 (±0.7)	18.3	0	79.0	0.03%	0.03%
CPT-RD (SimInit)⋆	66.3 (± 0.9)	**15.8**	0	**76.7**	71.6 (±1.6)	19.8	0	83.6	0.03%	0.03%
CPT-RD (MeanInit)⋆	64.5 (± 1.2)	14.8	0	75.6	70.3 (±1.5)	18.5	0	80.9	0.03%	0.03%
CPT-RD (SimInit+TPHNet)⋆	**66.7 (± 1.3)**	15.3	**0.2**	76.0	**71.9 (±1.2)**	20.0	0.5	84.0	0.1%	0.1%
P-tuning v2 based (PTv2-based)										
CPT-RD (CLInit)⋆	68.2 (± 1.0)	19.8	0	76.1	73.1 (±0.7)	30.7	0	83.2	0.6%	0.6%
CPT-RD (SimInit)⋆	72.2 (± 1.1)	23.3	0	79.1	75.0 (±1.2)	31.2	0	84.9	0.6%	0.6%
CPT-RD (MeanInit)⋆	70.6 (± 1.5)	20.5	0	75.5	73.5 (±1.3)	28.7	0	83.0	0.6%	0.6%
CPT-RD (SimInit+TPHNet)⋆	**75.0 (± 1.3)**	23.5	0.9	79.3	**76.5 (±1.6)**	31.4	1.1	**85.7**	1%	1%

Table 1: Results evaluated on test datasets for all tasks in `PHEME + Twitter15&16 + Covid19` and `Weibo`. The following averaged over 5 random task orders (Table 7) are reported, where (⋆) and (†) indicate frozen language model parameters and fine-tuning, (+Params) and (Tune Params) are additional parameters and the tunable parameters for each task. CLS and VER denote the output predictions with CLS token classifier and Verbalizer head, respectively. The fs.F1 is the result on the validation dataset, and the k-shot is 16.
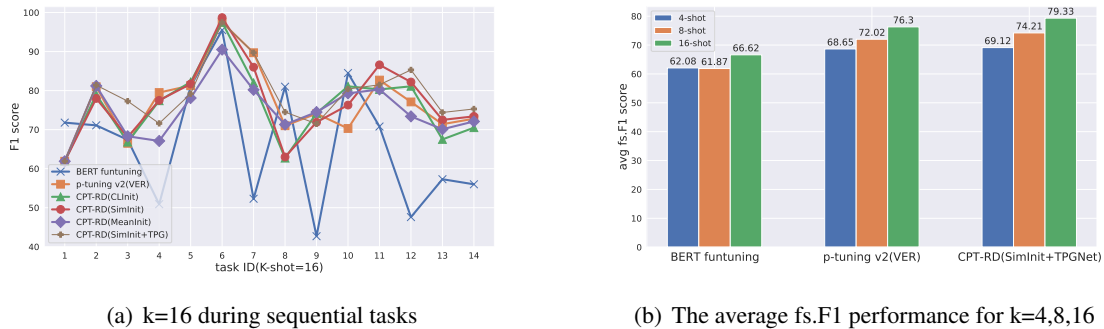

(a) k=16 during sequential tasks

(b) The average fs.F1 performance for k=4,8,16

Figure 4: The few-shot performance of each model on the dataset `PHEME + Twitter15&16 + Covid19`.

4.2 Compare models

In the experiments, BERT-base is used as the PLM weights, and CPT-RD will be compared with the following advanced models: **Fine-tuning:** Fine-tune the model on new task data continually. **EANN:** Wang et al. (2018) uses adversarial networks to extract event invariant features for generalization when new events arrive. For comparison, EANN is modified to use BERT as the encoder and only text data. **Adapter:** Freeze the pre-trained model and train a residual Adapter(Houlsby et al., 2019). **ParallelAdapter:** A variant by transferring the parallel insertion of prefix tuning into adapters (He et al., 2021). **prompt-tuning (CLS/VER):** which only tunes soft-prompts with a frozen language model (Lester et al., 2021), prompt for transformer's first layer. **p-tuning v2 (CLS/VER):** Using multilayer soft-prompts (deep

prompt tuning), where CLS/VER denotes the predicted output with [CLS] token and Linear classification layer and verbalizer MLMhead, respectively. **MTL p-tuning v2 (CLS/VER):** P-tuning v2 in a multi-task manner instead of CL. Train a single prompt using all tasks data concurrently.

4.3 Main results

The higher the FWT score, the better the model works for the unseen domain. A negative BWT indicates that the model produces forgetting, and if it is positive, it can accumulate knowledge.

Compare model performance. In Table 1, unsurprisingly, the Fine-tuning model has a severely CF, the BWT value of -24.3 under the CL setting. The level of catastrophic forgetting in EANN (Wang et al., 2018) is somewhat reduced compared to Fine-tuning but is still severe. Prompt-

Task ID	Task	Fine-tuning	p-tuning v2(VER)	CPT-RD(CLInit)	CPT-RD(SimInit)	CPT-RD(MeanInit)	CPT-RD(SimInit+TPHNet)
1	Charlie Hebdo	22.1	64.1	75.3	77.1	75.3	78.0
2	TwitterEvent4	65.1	88.0	72.1	65.2	70.6	74.4
3	Ferguson	24.7	62.4	70.3	72.7	67.5	72.8
4	Germanwings-crash	50.6	63.4	65.9	60.6	64.7	61.3
5	Ottawa Shooting	52.7	74.3	70.5	72.2	74.6	78.3
6	Prince Toronto	98.7	71.4	98.1	95.6	93.7	94.3
7	Putin missing	53.0	55.3	53.0	66.0	53.5	72.1
8	TwitterEvent1	77.1	77.1	77.1	77.8	70.7	74.5
9	Sydney Siege	42.8	72.5	74.2	65.7	75.8	74.9
10	TwitterEvent5	71.4	67.5	69.8	70.5	71.7	73.5
11	Gurlitt	43.7	48.3	42.2	62.3	52.5	62.5
12	Covid19	47.6	86.7	87.9	84.3	85.3	88.3
13	TwitterEvent2	57.8	69.0	60.5	70.8	65.2	72.4
14	TwitterEvent3	56.9	74.9	37.9	70.9	67.3	71.3
	Avg.	54.6	69.7	68.2	72.2	70.6	75.0

Table 2: The F1 score evaluated on the final model after all 14 tasks are visited in the test set. We use Avg. to represent the average F1 of all tasks for each method. The gray numbers indicates that CPT-RD (p-tuning v2 based) does not perform as well as normal p-tuning v2 in a single task.

tuning and p-tuning v2 are somewhat related to the adapter method in the form of parameter tuning (He et al., 2021), but their performance in CL differs. The prompt-based model is better than the adapter in both datasets. From Tabel 4 in Sec.4.5, consistent with Liu et al. (2021b), CLS token classification is better than the verbalizer when p-tuning v2 is under multi-task learning mode. The performance of multi-task learning is usually considered the upper limit of the CL model. However, verbalizer's FWT and final F1 scores in the CL setting were better than CLS but more prone to CF. Since SPL lets the model not generate CF, we next develop CPT-RD using prompt-tuning (PT) and p-tuningv2 (PTv2) baseline with verbalizer.

CPT-RD performance. We have developed CPT-RD based on PT and PTv2 respectively. We can find that since CPT-RD has SPL, the BWT values are all greater than or equal to 0, which indicates that this fundamentally eliminates the CF. The more stable performance among the three forward transfer strategies is SimInit. CLInit is the most unstable, probably because the soft-prompt from the previous task is not necessarily beneficial to the learning of the next task and is prone to cumulative negative effects on subsequent tasks in CL. It can be observed by Tabel 2 that the F1 score of CLInit on the last two tasks of the last sequence task is significantly lower than other strategies, 60.5 and 37.9 respectively. MeanInit with multi-source prompt does not work as expected and is only slightly better than CLInit. Finally, we added TPHNet to SimInit to achieve backward knowledge transfer, but the performance on PT is not stable enough, probably because PT has fewer tunable parameters. Furthermore, TPHNet incorporates additional tunable parameters but still has

parameter-efficient that is much lower than Fine-tuning.

Final model performance. The final model performance is the model's performance after learning the last task in the sequence. In Table 2, although CPT-RD solves CF, in the ideal case, the final model performance should outperform PTv2. We can find that the final performance of CLinit is inferior to PTv2 in 7 tasks, simInit and meanInit in 6, and SimInit+TPG in 5, which further indicates that TPHNet achieves a certain degree of backward knowledge transfer. Among them, we can find that in CLinit, the last task is as low as 37.9, indicating that directly reusing the soft-prompt of the previous task may have a more significant impact on the subsequent tasks.

Data privacy and Parameter explosion. From Sec.3.4, we propose TPHNet in the context of hypernetwork, which avoids rehearsal-based data replay and thus preserves data privacy. For a more comprehensive evaluation of our method, we replace TPHNet with a rehearsal-based technique, more details in Sec.4.4. From Table 3, it can be found that TPHNet can achieve comparable performance to the rehearsal-based method.

Besides, it can be seen from Sec.3.3 that CPT-RD avoids parameter explosion by using SPL to store soft-prompt (without dynamic model expansion). And the storage space occupied by SPL is negligible. In Table 2, the tuning parameter of the Adapter-based model is 2.1-2.3%, which is more parameter-efficient than the 100% of the Fine-tuning model. However, CPT-RD adds only 0.03-1% tunable parameters based on more parameter-efficient PT and can achieve better performance.

Few-shot performance. There is a lack of samples in the early burst stages, so few-shot perfor-

mance is critical for RD models. The final model is the model obtained after learning the last task. From Fig.4 (a), it can be found that Fine-tuning has a very unstable few-shot performance in CL. PTv2 still shows better few-shot capability than Fine-tuning in CL, which indicates that the PT-based model is more suitable for domain generalization. In Fig.4 (b), the average fs.F1 of CPT-RD (SimInit+TPHNet) is higher than that of PTv2, indicating that continuously accumulated knowledge can be better used for few-shot.

4.4 CPT-RD with rehearsal buffer

To verify the gap between TPHNet and rehearsal-based technology, we randomly sample 50 pieces of data from each domain training set in memory, and jointly trained on the future domain dataset. It can be observed from Table 3 that the rehearsal-based technology reflects the certain ability to backward knowledge transfer by accumulating data in past domain tasks. From Table 1, the BWT value of TPHNet is 0.9, which shows that our method is close to rehearsal-based technology, but our advantage is that the memory occupies small, and there is no data privacy problem.

Buffer size	Methods (PTv2-based)	Avg. F1	FWT	BWT
0/domain	p-tuning v2 (VER)	69.7	13.9	-11.0
	CPT-RD (CLInit)	68.2	19.8	0
	CPT-RD (MeanInit)	70.6	20.5	0
	CPT-RD (SimInit)	72.2	23.3	0
50/domain	p-tuning v2 (VER)	71.2	14.1	1.5
	CPT-RD (CLInit)	72.1	20.1	1.1
	CPT-RD (MeanInit)	71.7	19.4	0.8
	CPT-RD (SimInit)	75.3	23.8	1.2

Table 3: The performance of CPT-RD (PTv2-based) on the `PHEME + Twitter15&16 + Covid19` (English) dataset using rehearsal-based technology.

4.5 Multi-task learning performance

The performance of multi-task learning methods is often defined as an upper bound on the performance of continual learning. Multi-task learning avoids CF by visiting data from different domain tasks at different times. In Table 4, we experimented with the model's performance in the multi-task learning mode under the Chinese and English datasets. Consistent with the experimental conclusion of Liu et al. (2021b), p-tuning v2 using CLS token classification outperforms the verbalizer under multi-task learning. However, the verbalizer outperforms CLS in FWT and final F1 score in the CL setting, but is more prone to CF.

Methods	Avg. F1	+Params	Tune Params
PHEME + Twitter15&16 + Covid19			
MTL p-tuning v2 (CLS)	80.3 (± 0.8)	0.6%	0.6%
MTL p-tuning v2 (VER)	77.9 (± 0.8)	0.6%	0.6%
Weibo			
MTL p-tuning v2 (CLS)	85.3 (± 0.8)	0.6%	0.6%
MTL p-tuning v2 (VER)	83.8 (± 0.8)	0.6%	0.6%

Table 4: The performance of the baseline model p-tuning v2 in the multi-task learning mode.

model	Avg.F1	FWT	BWT
CPT-RD	75.0	23.5	0.9
w/o TPHNet	72.2	23.3	0
w/o SimInit	71.3	14.2	0.9
w/o SimInit+TPHNet	69.7	13.9	-11.0

Table 5: Ablation study on the effectiveness of the CPT-RD (SimInit+TPHNet) PTv2-based on the test set in the `PHEME + Twitter15&16 + Covid19`.

4.6 Ablation study

To understand the effectiveness of the different techniques proposed, we conducted an ablation study. From Table 1, we know that CPT-RD is more stable in PTv2-based than PT-based, which indicates that the model can benefit from more tunable parameters. Conventional PT do not use forward transfer strategy and TPHNet with lower FWT values. In PTv2-based, the overall performance is highest when SimInit and TPHNet are used together. In Table 5, removing either of TPHNet or SimInit will result in a decrease in overall performance. This shows the validity of our improvements, which would degrade to p-tuning v2 if SimInit and TPHNet were both removed.

5 Conclusion

We explore how to continually detect a social network environment with frequent unseen domains in RD. The novel framework Continual Prompt-tuning RD (CPT-RD) is proposed, which includes various knowledge transfer techniques. In the face of emergency domains, CPT-RD can use soft-prompt initialization strategies to achieve fast generalization. There is also a task-conditioned prompt-wise generator network (TPHNet) in terms of continual accumulation of detection knowledge. Future works include extending usage of CPT-RD to task agnostic scenarios and designing more diverse knowledge transfer strategies.

References

Eyal Ben-David, Nadav Oved, and Roi Reichart. 2021. Pada: A prompt-based autoregressive approach for adaptation to unseen domains. *arXiv preprint arXiv:2102.12206.*

Tian Bian, Xi Xiao, Tingyang Xu, Peilin Zhao, Wenbing Huang, Yu Rong, and Junzhou Huang. 2020. Rumor detection on social media with bi-directional graph convolutional networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 549–556.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165.*

Zhiyuan Chen and Bing Liu. 2018. Lifelong machine learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 12(3):1–207.

Qiming Diao, Jing Jiang, Feida Zhu, and Ee Peng LIM. 2012. Finding bursty topics from microblogs. Acl.

Bin Guo, Yasan Ding, Lina Yao, Yunji Liang, and Zhiwen Yu. 2020. The future of false information detection on social media: New perspectives and trends. *ACM Computing Surveys (CSUR)*, 53(4):1–36.

Gunshi Gupta, Karmesh Yadav, and Liam Paull. 2020. La-maml: Look-ahead meta learning for continual learning. *arXiv preprint arXiv:2007.13904.*

Raia Hadsell, Dushyant Rao, Andrei A Rusu, and Razvan Pascanu. 2020. Embracing change: Continual learning in deep neural networks. *Trends in cognitive sciences.*

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2021. Towards a unified view of parameter-efficient transfer learning. *arXiv preprint arXiv:2110.04366.*

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR.

Khurram Javed and Martha White. 2019. Meta-learning representations for continual learning. *arXiv preprint arXiv:1905.12588.*

Xisen Jin, Bill Yuchen Lin, Mohammad Rostami, and Xiang Ren. 2021. Learn continually, generalize rapidly: Lifelong knowledge accumulation for few-shot learning. *arXiv preprint arXiv:2104.08808.*

Zixuan Ke, Bing Liu, Nianzu Ma, Hu Xu, and Lei Shu. 2021a. Achieving forgetting prevention and knowledge transfer in continual learning. *Advances in Neural Information Processing Systems*, 34.

Zixuan Ke, Hu Xu, and Bing Liu. 2021b. Adapting bert for continual learning of a sequence of aspect sentiment classification tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4746–4755.

Joseph KJ and Vineeth N Balasubramanian. 2020. Meta-consolidation for continual learning. *Advances in Neural Information Processing Systems*, 33:14374–14386.

Nayeon Lee, Andrea Madotto, Yejin Bang, and Pascale Fung. 2021. Dynamically addressing unseen rumor via continual learning. *arXiv preprint arXiv:2104.08775.*

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691.*

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190.*

Hongzhan Lin, Jing Ma, Mingfei Cheng, Zhiwei Yang, Liangliang Chen, and Guang Chen. 2021. Rumor detection on twitter with claim-guided hierarchical graph attention networks. *arXiv preprint arXiv:2110.04522.*

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021a. Pretrain, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586.*

Xiao Liu, Kaixuan Ji, Yicheng Fu, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2021b. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv preprint arXiv:2110.07602.*

David Lopez-Paz and Marc'Aurelio Ranzato. 2017. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30:6467–6476.

Heng-yang Lu, Chenyou Fan, Xiaoning Song, and Wei Fang. 2021. A novel few-shot learning based multi-modality fusion model for covid-19 rumor detection from online social media. *PeerJ Computer Science*, 7:e688.

Jing Ma, Wei Gao, and Kam-Fai Wong. 2018. Detect rumor and stance jointly by neural multi-task learning. In *Companion proceedings of the the web conference 2018*, pages 585–593.

Ray Oshikawa, Jing Qian, and William Yang Wang. 2018. A survey on natural language processing for fake news detection. *arXiv preprint arXiv:1811.00770.*

Parth Patwa, Mohit Bhardwaj, Vineeth Guptha, Gitanjali Kumari, Shivam Sharma, Srinivas Pykl, Amitava Das, Asif Ekbal, Md Shad Akhtar, and Tanmoy Chakraborty. 2021a. Overview of constraint 2021 shared tasks: Detecting english covid-19 fake news and hindi hostile posts. In *International Workshop on Combating On line Ho st ile Posts in Regional Languages dur ing Emerge ncy Si tuation*, pages 42–53. Springer.

Parth Patwa, Shivam Sharma, Srinivas Pykl, Vineeth Guptha, Gitanjali Kumari, Md Shad Akhtar, Asif Ekbal, Amitava Das, and Tanmoy Chakraborty. 2021b. Fighting an infodemic: Covid-19 fake news dataset. In *International Workshop on Combating On line Ho st ile Posts in Regional Languages dur ing Emerge ncy Si tuation*, pages 21–29. Springer.

Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. 2018. Learning to learn without forgetting by maximizing transfer and minimizing interference. *arXiv preprint arXiv:1810.11910*.

Timo Schick and Hinrich Schütze. 2020. It's not just size that matters: Small language models are also few-shot learners. *arXiv preprint arXiv:2009.07118*.

Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. 2020. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. *arXiv preprint arXiv:2010.15980*.

Johannes von Oswald, Christian Henning, João Sacramento, and Benjamin F Grewe. 2019. Continual learning with hypernetworks. *arXiv preprint arXiv:1906.00695*.

Chengyu Wang, Haojie Pan, Yuan Liu, Kehan Chen, Minghui Qiu, Wei Zhou, Jun Huang, Haiqing Chen, Wei Lin, and Deng Cai. 2021a. Mell: Large-scale extensible user intent classification for dialogue systems with meta lifelong learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 3649–3659.

Yaqing Wang, Fenglong Ma, Zhiwei Jin, Ye Yuan, Guangxu Xun, Kishlay Jha, Lu Su, and Jing Gao. 2018. Eann: Event adversarial neural networks for multi-modal fake news detection. In *Proceedings of the 24th acm sigkdd international conference on knowledge discovery & data mining*, pages 849–857.

Yaqing Wang, Fenglong Ma, Haoyu Wang, Kishlay Jha, and Jing Gao. 2021b. Multimodal emergent fake news detection via meta neural process networks. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 3708–3716.

Zirui Wang, Sanket Vaibhav Mehta, Barnabás Póczos, and Jaime Carbonell. 2020. Efficient meta lifelong-learning with limited memory. *arXiv preprint arXiv:2010.02500*.

Huaiwen Zhang, Shengsheng Qian, Quan Fang, and Changsheng Xu. 2021. Multi-modal meta multi-task learning for social media rumor detection. *IEEE Transactions on Multimedia*.

Xinyi Zhou, Atishay Jain, Vir V Phoha, and Reza Zafarani. 2020. Fake news early detection: A theory-driven model. *Digital Threats: Research and Practice*, 1(2):1–25.

Arkaitz Zubiaga, Ahmet Aker, Kalina Bontcheva, Maria Liakata, and Rob Procter. 2018. Detection and resolution of rumours in social media: A survey. *ACM Computing Surveys (CSUR)*, 51(2):1–36.

A Appendix

A.1 Event correlation

After filtering the stopping words, we use TF-IDF to calculate the event relevance of each task domain, as shown in Fig.5.

Datasets	tasks	non-rumor	rumor	total
	Gurlitt	77	61	138
	Putin missing	112	126	238
	Prince Toronto	4	229	233
PHEME	Germanwings-crash	231	238	469
	Ferguson	859	284	1143
	Charlie Hebdo	1621	458	2079
	Ottawa Shooting	420	470	890
	Sydney Siege	699	522	1221
	TwitterEvent 1	63	156	219
	TwitterEvent 2	89	164	253
Twitter15&16	TwitterEvent 3	153	198	351
	TwitterEvent 4	146	200	346
	TwitterEvent 5	128	436	564
Covid19	Covid19	3060	3360	6420
	total	7662	6902	14564
	MH370	133	262	395
	Olympics	173	81	254
	Urban managers	94	149	243
	Cola	215	419	634
	Child trafficking	94	172	266
	Waste oil	133	57	190
Weibo	Accident	100	82	182
	Earthquake	117	58	175
	Typhoon	107	64	171
	Rabies	101	42	143
	College entrance exams	147	590	737
	Lockdown the city	86	24	110
	Zhong Nanshan	55	21	76
	Wuhan	167	69	236
	total	1722	2048	3770

Table 6: Statistics of `PHEME + Twitter15&16 + Covid19` (English) and `Weibo` (Chinese) datasets.

A.2 Implementation details

We tune hyperparameters on the `PHEME + Twitter15&16 + Covid19` and `Weibo` validation sets. We tune learning rates by enumerating over [3e-3, 5e-3, 7e-3], and finally use a learning rate of 7e-3 for all CPT-RD approaches where a learning rate of 1e-4 for TPHNet. The learning rate of the fine-tuning approaches is 5e-5, and the learning rate of the adapter approaches is 1e-4. Regarding the PT and PTv2, including the MTL approach, we use a learning rate of 5e-3. We use a batch size of 16 across experiments. We train the model for at most 100 epochs for each training task with a patience of 4 epochs without validation performance improvement. Before training on a new task, we revert the model to the checkpoint with the best validation performance in the previous task. In the few-shot learning stage, we use the same learning rate and train the model for 500 steps ($k \in \{16, 8, 4\}$), assuming no validation sets to perform early stopping. The length of the input sequence on data set `PHEME + Twitter15&16 + Covid19` is 300, and `Weibo` is 128. We also experimented with the length of different soft-prompt tokens, such as $\{20, 40, 60, 80\}$, and found that the performance of 40/60 is relatively stable, so 40 is uniformly used as the length of soft-prompt in the experiment.

Task Order	Tasks
PHEME + Twitter15&16 + Covid19	
Order1	Charlie Hebdo, TwitterEvent4, Ferguson, Germanwings-crash, Ottawa Shooting, Prince Toronto, Putin missing, TwitterEvent1, Sydney Siege, TwitterEvent5, Gurlitt, Covid19, TwitterEvent2, TwitterEvent3
Order2	Sydney Siege, ferguson,TwitterEvent1, Gurlitt, Ottawa Shooting, TwitterEvent3, Prince Toronto, TwitterEvent4, Putin missing, Charlie Hebdo, TwitterEvent5, Germanwings-crash, TwitterEvent2, Covid19
Order3	TwitterEvent3, Charlie Hebdo, TwitterEvent2, Ferguson, TwitterEvent1, TwitterEvent5, TwitterEvent4, Putin missing, Ottawa Shooting, Prince Toronto,Gurlitt, Germanwings-crash, sydneysiege, Covid19
Order4	TwitterEvent1, Germanwings-crash, TwitterEvent4, Ferguson, Gurlitt, Sydney Siege, TwitterEvent3, Charlie Hebdo, Ottawa Shooting, Prince Toronto,TwitterEvent2, TwitterEvent5, Putin missing, Covid19
Order5	Covid19, Germanwings-crash, Prince Toronto, TwitterEvent3, TwitterEvent5, Sydney Siege, Ferguson, Ottawa Shooting, Charlie Hebdo,TwitterEvent4, TwitterEvent2, Gurlitt, Putin missing, TwitterEvent1
Weibo	
Order1	Typhoon, Olympic, MH370, Earthquake, Rabies, College entrance exams, Cola, Urban managers, Child trafficking, Accident, Waste oil, Zhong Nanshan, Wuhan, Lockdown the city
Order2	Olympic, Child trafficking, Rabies, Accident, Earthquake, Cola, College entrance exams,Typhoon, Urban managers, Waste oil, MH370, Zhong Nanshan, Wuhan, Lockdown the city
Order3	Zhong Nanshan, Wuhan, Lockdown the city, Cola, Accident, Urban managers, Waste oil, Earthquake, Olympic, Child trafficking,Typhoon, Rabies, College entrance exams, MH370
Order4	Cola, Accident, Olympic, Waste oil, Typhoon, Zhong Nanshan, Wuhan, Lockdown the city, College entrance exams, Urban managers, Rabies, Child trafficking, MH370, Earthquake
Order5	Zhong Nanshan, Wuhan, Lockdown the city, MH370, Urban managers, Child trafficking, Typhoon, Earthquake, Olympic, Cola, Accident, Rabies, Waste oil, College entrance exams

Table 7: Order of continual learning tasks in `PHEME + Twitter15&16 + Covid19` and `Weibo` datasets.
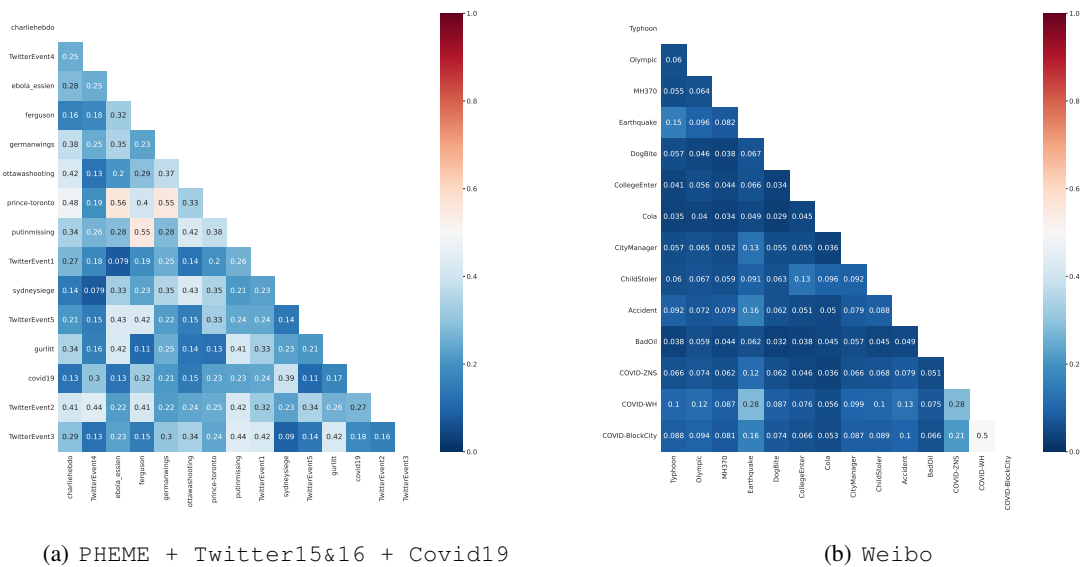

(a) `PHEME + Twitter15&16 + Covid19`   (b) `Weibo`

Figure 5: Domain Task Relevance Heat Map.