SPARSE MISINFORMATION DETECTOR

Anonymous authors
Paper under double-blind review

ABSTRACT

We present *Sparse Misinformation Detector* (SMD), a new efficient misinformation detection network with regular fine-grained sparsity. We propose two technical components to enable SMD. First, *CircuSparsity*, a new hardware-friendly sparsity pattern, is introduced for improved training and testing efficiency. Second, through dedicated empirical analyses, we discover that document-level misinformation detection is pretty insensitive to a compact model size, which inspires us to make *early exit* for the document-level misinformation classifier. With these two techniques, we successfully achieve efficient misinformation detection on both document and event levels with *one single* model. Empirically, our approach *significantly outperforms* the original dense misinformation detection network while enjoying *50% to 75% sparsity*. Extensive experiments and analyses demonstrate the merits of our method compared to other top-performing counterpart approaches. To our best knowledge, this is the *first* attempt for efficient misinformation detection from the network sparse training perspective.

1 INTRODUCTION

Recent years have seen rampant flooding of misinformation on the Internet. The need of detecting such misinformation is more and more imperative (Wang, 2017; Karimi et al., 2018; Zellers et al., 2019; Yang et al., 2022; Jin et al., 2022; Aneja et al., 2021; Wu et al., 2022). Fake news detection methods have seen considerable progress by employing the deep neural network based models. Their performance is remarkable, yet still, they have many shortcomings.

First, most of these methods can only detect fake news on document level (Wang, 2017; Karimi et al., 2018; Zellers et al., 2019; Tan et al., 2020; Hu et al., 2021; Fung et al., 2021; Huang et al., 2022; Jin et al., 2022); it is desired to know what renders the news fake, *i.e.*, detecting the misinformation on a more fine-grained level – the event level. Second, a few works make advances in the event-level misinformation detection (*e.g.*, Wu et al. (2022)). However, their models are typically very redundant and consume sizable storage (*e.g.*, one of the top-performing misinformation detection model in Wu et al. (2022) consumes more than 300MB on disk), memory footprint, and inference time, making them *rather hard* to be deployed on resource-limited devices, such as smartphones (since people are used to receiving news on their smartphones everyday). Third, these misinformation detection models are usually trained *separately* (see Fig. 1(a)) for document-level and event-level detection, which is a significant waste of resource since these two tasks are inherently related. Intuitively, we may significantly slim the model size by sharing the representations.

In this paper, we present a new and novel Sparse Misinformation Detector (SMD) based on the recent advances in sparse training to resolve these shortcomings. Specifically, our SMD has two major components. First, we propose *CircuSparsity*, a kind of new regular fine-grained sparsity pattern, which has a circulant structure (see Fig. 2). Such regular sparsity pattern can save considerable storage and improve the inference speed significantly. Second, through extensive empirical studies, we find that the document misinformation classifier is much more *insensitive* to the event misinformation classifier, which inspires us to reduce the network depth of the document-level detector by making *early exit* for the document classifier.

Empirically, we conduct extensive benchmarking and analyses to show the effectiveness of our method in the comparison to the original dense model and other counterpart sparse training approaches: At *sparsity 50% to 75%*, our model still outperforms the original dense models and coun-

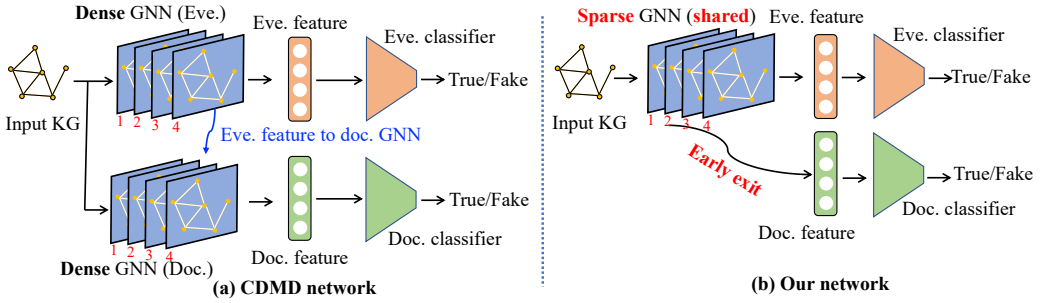


Figure 1: Illustration of the proposed *sparse misinformation detector* (SMD) model (b) compared to the existing method CDMD (Wu et al., 2022) (a). The key technical novelties are highlighted in **bold red** in the figure: **(1)** Our method can achieve misinformation detection at both document and event levels with *one single* GNN; **(2)** the GNN in our model is *sparse* (with a proposed special sparsity pattern); **(3)** we make *early exit* for the document branch, which can reduce the conflict between document-level and event-level losses when they are used within the same model.

terpart sparse training approaches by an obvious margin. If only the document-level detection is considered, our method can achieve *90% sparsity* with performance still improved.

Our contributions in this work can be summarized as follows:

- We integrate sparse training for efficient misinformation detection. A new and novel hardware-friendly sparsity pattern, *CircuSparsity*, is introduced. As far as we know, this is the *first attempt* that marries sparse training to the task of misinformation detection for improved efficiency in both training and testing.
- By dedicated analyses, we propose to make early exit for the document classifier so as to resolve its conflict with the event-level misinformation detection. This simple scheme effectively coordinates the two loss objectives (document-level and event-level detection losses) within one single model.
- Empirically, our method achieves comparable or even better performance that the dense network and other sparse training approaches, *while having sparsity at 50%-75%.* For the document level (*i.e.*, if only fake news detection is considered), our method can sparsify the model at even 90% sparsity while still outperforming the original dense model.

2 Related Work

Misinformation detection. With misinformation becoming an important social issue, increasing effort has been invested in automatic misinformation detection in language-only, cross-modality or cross-lingual documents either on document level (Wang, 2017; Karimi et al., 2018; Zellers et al., 2019; Tan et al., 2020; Hu et al., 2021; Fung et al., 2021; Huang et al., 2022; Jin et al., 2022) or event level (Yang et al., 2022; Jin et al., 2022; Aneja et al., 2021). For instance, Yang et al. (2022) proposes a subgraph reasoning paradigm. The latest and first work on cross-document misinformation detection uses GNNs (Graph Neural Networks) to leverage cross-document information and significantly outperforms previous models (Wu et al., 2022). Despite the encouraging progress, these methods either cannot detect misinformation on fine-grained level (*i.e.*, event level), or suffer from overparameterization that hinders efficient training and deployment in real-world edge devices. Therefore, we aim resolve these issues by detecting misinformation on both document and event level with one single sparse model. In this work, we present a novel sparse training method that can train the sparse model more efficiently while outperforming the original dense model.

Sparse training. Sparse training is a branch of neural network pruning at initialization (Wang et al., 2022), which prunes a *randomly initialized* network (conventional pruning methods typically prunes a pretrained network) and then keep it sparse over the training process. The major merit of sparse training is that it can enjoy the efficiency at the training stage, not only the inference stage. It is pioneered by two works LTH (Frankle & Carbin, 2019) and SNIP Lee et al. (2019). Many sparse training works focus on proposing new pruning criteria (*i.e.*, how to decide which weights should be zeroed), such as GraSP (Wang et al., 2020a), SynFlow (Tanaka et al., 2020). These works derive their

sparsity pattern *based on the random network*, so when the network is re-initialized, the associated mask will change accordingly. This makes the sparsity pattern *unpredictable* in advance, rendering it *very challenging* to exploit such sparsity for actual acceleration without customized hardware and software support (Wen et al., 2016). Different from these works, our method employs a carefully designed sparsity pattern (see Fig. 2), which is *predefined*, not derived from the base model. We will show the advantage of such sparsity pattern in terms of both performance and efficiency.

Sparse transformers. The misinformation detection model used in this work is based on the attention mechanism in transformers (Vaswani et al., 2017). Transformers have seen rapid progress in lager languages models like BERT (Devlin et al., 2018) while being pretty costly. Many works have attempted to reduce the complexity of transformers via sparsity, *e.g.*, Beltagy et al. (2020); Child et al. (2019); Guo et al. (2019); Kitaev et al. (2020); Zaheer et al. (2020). These said, of note, these works exploit sparsity in a *completely different* manner as we do – they introduce sparsity patterns (*e.g.*, sliding windows (Beltagy et al., 2020) and BigBird (Zaheer et al., 2020)) for the dense fully-connected structure of self-attention, in the hopes of avoiding unnecessary attention computation. The network per se in their methods is *not* sparse. In stark contrast, we study sparsity at the *network* level. As far as we know, our work is *the first attempt* to improve the misinformation detection efficiency from the network sparsity perspective.

Other efficient deep learning approaches. In addition to network pruning, there are usually another four categories of efficient deep learning methods: quantization (Courbariaux & Bengio, 2016; Courbariaux et al., 2016; Rastegari et al., 2016), knowledge distillation (Buciluǎ et al., 2006; Hinton et al., 2014; Chen et al., 2017; Wang et al., 2020b; Jiao et al., 2019; Wang & Yoon, 2021), neural architecture design or search (Howard et al., 2017; Sandler et al., 2018; Howard et al., 2019; Zhang et al., 2018; Ma et al., 2018; Tan & Le, 2019), low-rank decomposition (Denton et al., 2014; Jaderberg et al., 2014; Lebedev et al., 2014; Zhang et al., 2015). In general, these methods are orthogonal to the sparsity techniques we employ in this work. Possible integration with them to further enhance the efficiency is left for future work.

3 METHODOLOGY

In this section, we first introduce the problem formulation of misinformation detection via knowledge graph reasoning with GNNs. Then, we elaborate the details of the proposed SMD method. Finally, we explain several key implementation details of SMD.

3.1 KNOWLEDGE GRAPH REASONING FOR MISINFORMATION DETECTION

The input raw news texts are processed by information extraction methods (such as OneIE (Lin et al., 2020)) to get structured information (such as entities, relations, events). Cross-document knowledge graph (KG) is then built upon the structured information. Typically, some language pretrained model (such as BERT (Devlin et al., 2018)) is employed to obtain powerful abstract representation for these structured information. Thus, the input of the misinformation detection system is a KG, of which some nodes and edges are BERT-encoded representations. Such a KG is then fed into a GNN to obtain more abstract representations. Finally, these representations go through a 2-class classifier to predict the input news (or equivalently, its KG) is fake or true.

In total, a KG includes 4 types of nodes (entity, event, event cluster, document) and 10 types of edges (entity-to-entity or relation, entity-to-event or event argument, entity-to-document, event-to-event cluster, event-to-document; and the inverse edges of these 5 kinds of edges). Let r denote an edge type and $\mathcal{R}$ denote the set of all edge types (10 types in total). Consider a node $v \in \mathcal{V}$ in the l-th layer, its feature is obtained by aggregating the output of all edge types (Wu et al., 2022),

$$\mathbf{h}_i^{(l)} = \sum_{r \in \mathcal{R}} \mathbf{h}_{i,r}^{(l)} / |\mathcal{R}|. \tag{1}$$

Each edge is associated with a separate small network, more concretely, a GAT (Graph Attention Network) (Veličković et al., 2018). The type of the GAT is decided by the edge genre: For *relation* and *event argument* edges, since they have concrete physical representations in the input KG, *edge-aware GAT* is employed to model these edges; for the other three edges, they do not have physical representations in the input KG, so the standard GAT is adopted.

GAT and edge-aware GAT. GAT aggregates the representation from its neighbor nodes via the attention mechanism (Vaswani et al., 2017). Edge-aware GAT is an extension of GAT by considering the edge features (apart from the node features) in the graph forward computation (Huang et al., 2020; Yasunaga et al., 2021). Our method builds upon these GNN infrastructure adopted by Wu et al. (2022). We focus on improving the training and inference efficiency of these networks, not proposing new GNN architectures. Interested readers may refer to Wu et al. (2022) for more details.

The GNN processes an input KG to obtain rather abstract representations. The document node representation is then utilized for the document-level misinformation detection, and event node representation for event-level misinformation detection, by training two *separate* classifiers $\mathbf{f}_\mathrm{d}$ and $\mathbf{f}_\mathrm{e}$.

3.2 Sparse Misinformation Detector (SMD)

The goal of this work is to achieve misinformation detection at both the document and event levels *with one single model*. Previous approach CDMD (Wu et al., 2022) provides a strong baseline but their classifiers are trained *separately*, that is, they requires two models for document and event detection (see Fig. 1(a)). This is clearly against our *efficient* misinformation detection goal.

A naive way to achieve the one-model goal is to *share* the GNN part (see Fig. 1(b)) and combine the two classification losses together (with a certain weight to properly balance the two objectives),

$$L = \alpha \cdot \mathrm{CE}(\mathbf{f}_\mathrm{d}(\mathbf{h}_\mathrm{d}), y_\mathrm{d}) + (1-\alpha) \cdot \mathrm{CE}(\mathbf{f}_\mathrm{e}(\mathbf{h}_\mathrm{e}), y_\mathrm{e}), \tag{2}$$

where CE is short for cross-entropy loss; $\mathbf{f}_\mathrm{d}$/$\mathbf{f}_\mathrm{e}$ refers to the document/event classifier; $\mathbf{h}_\mathrm{d}$/$\mathbf{h}_\mathrm{e}$ is the final document/event node feature, and y_d/y_e represents the ground-truth label (0 or 1) for the document/event node; The $\alpha \in [0, 1]$ is a coefficient to balance the two loss objectives.

However, this naive way does not work well. As we will show in the experiments, the two losses are actually *against* each other. Combining them together often *degrades* the performance of document and event misinformation detection (compared to using a separate GNN for each task exclusively). Meanwhile, the GNN itself is pretty redundant. The proposed SMD method is meant to resolve these two problems. Next, we will first introduce a new network sparsity pattern (called *CircuSparsity*) to reduce the GNN redundancy. Then, we explain how to coordinate the two losses well within one network by a simple early exit scheme.

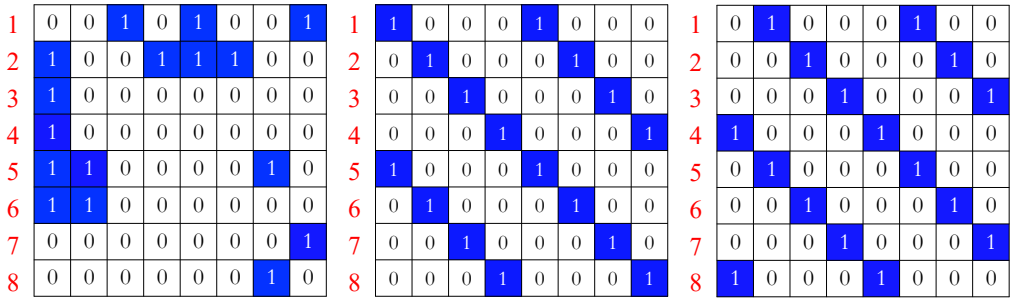
(1) CircuSparsity. As shown in Fig. 2, CircuSparsity is essentially a *circulant matrix* (Gray et al., 2006), namely, each row of it is a cyclic shift of the row above it. The circulant matrix can be fully decided by its first row, so the key is how to design the first row in CircuSparsity.

Consider there are K elements in the first row. We design the sparsity pattern by a *maximal regularity principle*. That is, we seek the *most irreducible* pattern (which we term *base sparsity pattern* in this paper). For a concrete example, for $K = 8$, given a sparsity ratio 75% (*i.e.*, 6 elements are zero), ideally, we have $\binom{8}{2} = 28$ sparsity pattern candidates (they all meet the condition of 75% sparsity). While we prefer the patterns illustrated in Fig. 2 (b) and (c), where their base sparsity pattern is $[1, 0, 0, 0]$ and $[0, 1, 0, 0]$, respectively. Since the sparsity is 75%, *i.e.*, 3 out of 4 elements are zero. Such patterns in Fig. 2 (b) and (c) are irreducible. We repeat these base sparsity patterns to fill up the first row of K elements. Then, all the other rows can be decided by the circulant matrix definition.

(2) One model to solve them all: Early exit. Here we explain how to mitigate the conflict between the two loss objectives of document- and event-level detection when merging them into one model.

By careful sensitivity analyses (see Sec. 4.1), we observe that document-level misinformation detection is *much easier* than the event-level detection. This is also straightforward to understand, since even-level misinformation detection is *more fine-grained* and thus *harder* than document-level. Taking advantage of this discovery, we propose to make *early exit* for the document-level misinformation detection task. The original GNN for the document misinformation detection has four layers (Wu et al., 2022) and we propose to make the exit at the *second* layer for the document misinformation detection classifier, as shown in Fig. 1(b).

This design has two advantages. **First**, for document misinformation detection, it only need to pass two GNN layers, so the computation would be less and speed is faster. **Second**, the higher layers (the third and fourth layers) are not interfered by the document misinformation detection loss gradient, so these layers are purely serving the *event-level* misinformation detection. This can well preserve



(a) Random Sparsity (b) CircuSparsity-1 (ours) (c) CircuSparsity-2 (ours)

Figure 2: Illustration of the proposed *CircuSparsity* (b and c) compared to the random sparsity (a) derived from other sparse training methods. The example sparsity here is 75%. The difference between (b) and (c) is the *base sparsity pattern* (the minimal irreducible sparsity pattern at the most upper-left of the mask matrix): $[1, 0, 0, 0]$ versus $[0, 1, 0, 0]$. CircuSparsity is essentially a kind of circulant matrix with special designs (see Sec. 3.2 for more details), which enjoys the benefits of smaller memory footprint and faster computation with easy implementation.

the event-level misinformation detection performance according to our empirical study. Meanwhile, due to the fact that the GNN is very redundant for document-level misinformation detection, early exit at the second layer does not deteriorate much the document-level performance. In short, this simple design brings considerable gains at only marginal cost.

To further reduce the impact of the document misinformation detection loss to the event-level misinformation detection, we propose to use a lower coefficient for the document misinformation detection (specifically, the document coefficient is 0.2 based on our ablation study; see Tab. 2).

3.3 Implementation Details

We employ *uniform* layerwise sparsity ratios (*i.e.*, each GNN layer is pruned by the same percentage). The mask is *predefined* before the training starts and then *fixed* during training, in contrast to many pruning works (Han et al., 2015; 2016; Li et al., 2017; Wang et al., 2020a; 2021) whose masks are inferred from a pretrained model or learned during training. The fixed sparsity reduces the cost of storing the non-zero weight positions and benefits hardware locality for practical speedup.

Building upon the CDMD model (Wu et al., 2022), we apply the proposed CircuSparsity to all the fully-connected (FC) layers in the GNN of the CDMD model, leaving the *classifier unpruned*. Our primary considerations are, (1) the FC layers in the classifier only account for very few (0.4%) parameters; (2) those layers are pretty close to the final classification; it is better to retain them so as not to affect the classification performance, as many prior pruning works (Li et al., 2017; Gale et al., 2019; Wang et al., 2021) do.

4 Experimental Results

Datasets and networks. There were few open datasets that enable event-level misinformation detection. Wu et al. (2022) recently released three new datasets with baseline benchmarks well established: *IED, TL17, and Crisis*. Therefore, we conduct our empirical analyses on these three datasets. The IED dataset is a complex event corpus. A complex event (Li et al., 2021) refers to a real-world incident that is described by multiple documents. TL17 and Crisis are two datasets of news timeline summarization. One timeline is made up with multiple documents covering a long-term event. All the three datasets can be regarded as multiple *clusters*. Each cluster consists of multiple news documents. IED has 422 clusters for training, 140 clusters for validation, and another 140 clusters for testing. TL17 has 276 clusters for training, 92 clusters for validation, and 92 clusters for testing. Crisis has 1,413 clusters for training, 177 clusters for validation, and 177 clusters for testing. Interested readers may refer to Wu et al. (2022) for more details. Our network, a variant of the CDMD model (Wu et al., 2022), has four major GNN layers, with around 134 FC layers in total. Our model has 95.4M parameters and 0.109G FLOPs (per inference).

Table 1: Analysis of different early exits in our method for the document misinformation detection on the IED dataset at sparsity 75%. The original CDMD model (Wu et al., 2022) has four layers in default, so there are four exits ablated here ("Exit = #4" is the original setting). The best results are in red and second best in blue. Based on this table, we choose early exit at #2 for the document-level misinformation detection in our model (Fig. 1).

Doc. classifier exit	F-1 (doc.)	AUC (doc.)	F-1 (eve.)	AUC (eve.)
#4	$89.72_{\pm 0.83}$	$95.90_{\pm 0.17}$	$45.24_{\pm 1.22}$	$89.03_{\pm 0.47}$
#3	$89.75_{\pm 0.20}$	$95.76_{\pm 0.13}$	$46.16_{\pm 0.34}$	$89.40_{\pm 0.22}$
#2	$90.04_{\pm 0.02}$	$96.10_{\pm 0.14}$	$46.98_{\pm 0.74}$	$89.75_{\pm 0.30}$
#1	$90.35_{\pm 0.31}$	$96.70_{\pm 0.00}$	$45.32_{\pm 0.61}$	$88.95_{\pm 0.19}$

Table 2: Hyper-parameter analysis of α in our method (with document early exit #2) on the IED dataset at sparsity 75%. The best results are in red and second best in blue.

Metric \ α	0.1	0.2	0.3	0.4	0.5	0.6	0.7
F-1 (doc.)	$88.77_{\pm 0.85}$	$89.59_{\pm 0.09}$	$89.01_{\pm 0.24}$	$89.18_{\pm 0.59}$	$90.04_{\pm 0.02}$	$88.75_{\pm 0.85}$	$88.74_{\pm 0.54}$
AUC (doc.)	$95.31_{\pm 0.58}$	$95.78_{\pm 0.19}$	$95.62_{\pm 0.33}$	$95.66_{\pm 0.22}$	$96.10_{\pm 0.14}$	$95.61_{\pm 0.20}$	$95.60_{\pm 0.13}$
F-1 (eve.)	$47.97_{\pm 0.58}$	$48.32_{\pm 0.53}$	$48.06_{\pm 0.87}$	$48.44_{\pm 0.44}$	$46.98_{\pm 0.74}$	$45.75_{\pm 1.01}$	$46.77_{\pm 0.83}$
AUC (eve.)	$90.08_{\pm 0.24}$	$90.26_{\pm 0.25}$	$90.28_{\pm 0.27}$	$90.39_{\pm 0.24}$	$89.75_{\pm 0.30}$	$89.16_{\pm 0.45}$	$89.61_{\pm 0.40}$

Metrics. Following Wu et al. (2022), we employ F-1 score and AUC (area under ROC curve) to evaluate the results. When calculating F-1 score, there is a threshold to balance the precision and recall. Following Wu et al. (2022), the best threshold is selected on the validation set and used on the test set. We report the benchmark results on the test set.

Training settings. Adam optimizer (Kingma & Ba, 2015) is used. Batch size is set to 16 following Wu et al. (2022). The number of total training epochs is set to 120 (for IED and TL17) and 10 (for Crisis). Initial learning rate (LR) is set to 5×10^{-5} after a warm-up period; the LR is then linearly decayed during training. Gradient clipping (threshold set to 1.0) is adopted for improving training stability. We use PyTorch (Paszke et al., 2019) for all of our experiments. We conduct the experiments on NVIDIA RTX 2080Ti and 3090 GPUs. *The code and trained models of this paper will be make publicly available to promote reproducibility*.

Benchmark methods. We compare our approach to CDMD (Wu et al., 2022), HDSF (Karimi & Tang, 2019), GROVER (Zellers et al., 2019), MP (magnitude pruning; without using pretrained model) (Han et al., 2015; 2016) and LTH (Lottery Ticket Hypothesis) (Frankle & Carbin, 2019). The last two are the most representative sparse training schemes to date. For CDMD (Wu et al., 2022), we reproduce their results referring to their official code[1]. In general, we obtain *comparable or even better* results than those reported in the CDMD paper (Wu et al., 2022). For other methods, due to lack of official public implementations, we cite the results from CDMD (Wu et al., 2022).

4.1 Ablation Study

We first present the ablation study results to show how the proposed algorithm works.

Ablation of the early exit of document classifier. There are four layers in the default CDMD model (Wu et al., 2022). We propose to make *early* exit for the document classifier, so there are three choices available – exit after the {first, second, third} GNN layer. In Tab. 1, we show the results of these different exits on the IED dataset.

(1) Of note, when the exit is earlier (*e.g.*, compare #1 to #4), the document-level F-1 and AUC pose a *rising* trend, *i.e.*, using fewer layers surprisingly improves the document-level performance. This suggests that the CDMD model is actually very redundant for the document-level misinformation detection. **(2)** Presumably, we expect document classifier exiting earlier would be better to the event-level performance since it means less interference to the event-level loss. However, this presumption

[1] https://github.com/shirley-wu/cross-doc-misinfo-detection

Table 3: Comparison of different *base sparsity patterns* in our proposed CircuSparsity (see Fig. 2) at sparsity 75% on the IED dataset (Wu et al., 2022). Row 1 is the default setting. The best of each column is highlighted in red and worst in orange.

Base sparsity pattern	F-1 (doc.)	AUC (doc.)	F-1 (eve.)	AUC (eve.)
$[1,0,0,0]$	$89.59_{\pm 0.09}$	$95.78_{\pm 0.19}$	$48.32_{\pm 0.53}$	$90.26_{\pm 0.25}$
$[0,1,0,0]$	$89.03_{\pm 0.46}$	$95.44_{\pm 0.42}$	$47.77_{\pm 0.11}$	$90.21_{\pm 0.06}$
$[0,0,1,0]$	$89.50_{\pm 0.31}$	$95.90_{\pm 0.04}$	$48.54_{\pm 0.24}$	$90.50_{\pm 0.14}$
$[0,0,0,1]$	$89.64_{\pm 0.68}$	$95.72_{\pm 0.20}$	$48.39_{\pm 0.67}$	$90.31_{\pm 0.19}$

Table 4: Evaluation on the **IED** test set. HDSF (Karimi & Tang, 2019), GROVER (Zellers et al., 2019), CDMD (Wu et al., 2022), MP (Han et al., 2015; 2016), LTH (Frankle & Carbin, 2019). MP and LTH are the most prevailing sparse training methods now. The mark † in LTH† indicates that LTH requires a *pretrained* model (others do not). We thereby include a case (SMD†) that uses a pretrained model too for a fair comparison to LTH. The best results are highlighted in red and second best in blue. The results with a gray background are at the most fair comparison setting of our interest (*i.e.*, no pretrained model used for efficiency; one model achieves misinformation detection at both document (doc.) and event (eve.) levels). The mark * indicates the results are directly cited from Wu et al. (2022) (others are obtained through our experiments); NA means not available; the mark / means no result for this entry.

Method	Level	Sparsity	F-1 (doc.)	AUC (doc.)	F-1 (eve.)	AUC (eve.)
HDSF		0	78.42*	NA	/	/
GROVER-mega		0	82.90*	NA	/	/
CDMD		0	$89.09_{\pm 0.71}$	$96.07_{\pm 0.40}$	/	/
MP	Doc.	90%	$88.23_{\pm 1.17}$	$95.29_{\pm 0.21}$	/	/
SMD (ours)		90%	$90.91_{\pm 0.00}$	$96.98_{\pm 0.04}$	/	/
LTH†		90%	$90.27_{\pm 0.10}$	$96.53_{\pm 0.01}$	/	/
SMD† (ours)		90%	$91.01_{\pm 0.04}$	$97.12_{\pm 0.02}$		
CDMD		0	/	/	$47.23_{\pm 1.62}$	$89.66_{\pm 0.59}$
MP		75%	/	/	$45.62_{\pm 0.38}$	$89.17_{\pm 0.14}$
SMD (ours)	Eve.	75%	/	/	$47.72_{\pm 0.38}$	$90.04_{\pm 0.18}$
LTH†		75%	/	/	$48.84_{\pm 0.63}$	$90.47_{\pm 0.28}$
SMD† (ours)		75%	/	/	$49.02_{\pm 0.33}$	$90.96_{\pm 0.20}$
CDMD		0	$87.47_{\pm 1.02}$	$94.62_{\pm 0.21}$	$46.85_{\pm 0.98}$	$89.70_{\pm 0.30}$
MP		75%	$87.13_{\pm 0.76}$	$94.15_{\pm 0.29}$	$46.51_{\pm 0.46}$	$89.50_{\pm 0.09}$
SMD (ours)	Both	75%	$89.59_{\pm 0.09}$	$95.78_{\pm 0.19}$	$48.32_{\pm 0.53}$	$90.26_{\pm 0.25}$
LTH†		75%	$88.30_{\pm 0.67}$	$94.84_{\pm 0.10}$	$47.69_{\pm 0.36}$	$89.97_{\pm 0.14}$
SMD† (ours)		75%	$90.02_{\pm 0.10}$	$96.03_{\pm 0.13}$	$48.56_{\pm 0.31}$	$90.64_{\pm 0.19}$

only holds to exit #2 – when it goes to exit #1, the event-level performance turns downward. This implies the event task still benefits from the representation learned by the document loss, to a certain degree. Balancing the results in Tab. 1, we finally select exit #2 as the default setting in our method.

Ablation of the document-event weight. In order to achieve the document-and event-level misinformation detection with one model, the document coefficient α (event coefficient is $1-\alpha$) should be set in range $(0,1)$. Tab. 2 shows the results on the IED dataset with different α's. A larger alpha implies the loss considers the document-level performance more, so we expect the document-level/event-level metrics would be higher/lower. This is generally confirmed in Tab. 2 for the event-level performance; for the document level, the results are pretty robust to the change of α – this again shows that the document-level task is pretty insensitive (due to the severely over-parameterized model). As seen from the table, the best α does not coincide for document and event levels, while the second best $\alpha = 0.2$ does, so we choose it as the default setting in our method.

Ablation of 1's position in base sparsity pattern. In our CircuSparsity (see Fig. 2), we set the mask 1 at the first entry of the base sparsity pattern. It is of interest if changing the location of the 1 affects the performance seriously. Tab. 3 shows the test results on the IED dataset with different locations of the 1 in the base sparsity pattern. As expected, the impact is insignificant. Since every weight in the network is randomly initialized, every weight can be discarded with equal probability.

Table 5: Evaluation on the **TL17** test set. HDSF (Karimi & Tang, 2019), GROVER (Zellers et al., 2019), CDMD (Wu et al., 2022), MP (Han et al., 2015; 2016), LTH (Frankle & Carbin, 2019). MP and LTH are the most prevailing sparse training methods now. The mark † in LTH† indicates that LTH requires a *pretrained* model (others do not). We thereby include a case (SMD†) that uses a pretrained model too for a fair comparison to LTH. The mark * indicates the results are directly cited from Wu et al. (2022) (others are obtained through our experiments); NA means not available; the mark / means no result for this entry.

Method	Level	Sparsity	F-1 (doc.)	AUC (doc.)	F-1 (eve.)	AUC (eve.)
HDSF		0	80.62*	NA	/	/
GROVER-mega		0	90.00*	NA	/	/
CDMD		0	$90.32_{\pm 0.54}$	$96.69_{\pm 0.24}$	/	/
MP	Doc.	90%	$90.48_{\pm 0.24}$	$96.87_{\pm 0.07}$	/	/
SMD (ours)		90%	$92.95_{\pm 0.47}$	$98.04_{\pm 0.11}$	/	/
LTH†		90%	$93.33_{\pm 0.23}$	$98.04_{\pm 0.04}$	/	/
SMD† (ours)		90%	$93.67_{\pm 0.18}$	$98.43_{\pm 0.10}$	/	/
CDMD		0	/	/	$45.09_{\pm 1.09}$	$84.91_{\pm 0.27}$
MP		75%	/	/	$44.36_{\pm 0.44}$	$84.70_{\pm 0.30}$
SMD (ours)	Eve.	75%	/	/	$44.85_{\pm 0.47}$	$85.24_{\pm 0.33}$
LTH†		75%	/	/	$44.79_{\pm 0.55}$	$85.64_{\pm 0.21}$
SMD† (ours)		75%	/	/	$45.11_{\pm 0.42}$	$86.07_{\pm 0.30}$
CDMD		0	$88.46_{\pm 0.57}$	$95.34_{\pm 0.22}$	$45.90_{\pm 0.71}$	$85.23_{\pm 0.61}$
MP		75%	$89.83_{\pm 0.10}$	$95.86_{\pm 0.34}$	$43.46_{\pm 0.81}$	$84.37_{\pm 0.20}$
SMD (ours)	Both	75%	$91.33_{\pm 0.31}$	$97.04_{\pm 0.15}$	$46.26_{\pm 0.80}$	$86.04_{\pm 0.21}$
LTH†		75%	$89.51_{\pm 0.78}$	$96.66_{\pm 0.07}$	$47.23_{\pm 0.30}$	$86.82_{\pm 0.09}$
SMD† (ours)		75%	$91.89_{\pm 0.21}$	$97.42_{\pm 0.30}$	$47.69_{\pm 0.15}$	$87.36_{\pm 0.38}$

Table 6: Evaluation on the **Crisis** test set. HDSF (Karimi & Tang, 2019), GROVER (Zellers et al., 2019), CDMD (Wu et al., 2022), MP (Han et al., 2015; 2016), LTH (Frankle & Carbin, 2019). MP and LTH are the most prevailing sparse training methods now. The mark † in LTH† indicates that LTH requires a *pretrained* model (others do not). We thereby include a case (SMD†) that uses a pretrained model too for a fair comparison to LTH. The mark * indicates the results are directly cited from Wu et al. (2022) (others are obtained through our experiments); NA means not available; the mark / means no result for this entry.

Method	Level	Sparsity	F-1 (doc.)	AUC (doc.)	F-1 (eve.)	AUC (eve.)
HDSF		0	82.14*	NA	/	/
GROVER-mega		0	87.13*	NA	/	/
CDMD		0	$95.47_{\pm 0.29}$	$99.37_{\pm 0.03}$	/	/
MP	Doc.	90%	$95.74_{\pm 0.09}$	$99.35_{\pm 0.04}$	/	/
SMD (ours)		90%	$96.47_{\pm 0.12}$	$99.47_{\pm 0.04}$	/	/
LTH†		90%	$96.76_{\pm 0.09}$	$99.60_{\pm 0.02}$	/	/
SMD† (ours)		90%	$96.47_{\pm 0.34}$	$99.59_{\pm 0.04}$	/	/
CDMD		0	/	/	$54.08_{\pm 0.42}$	$88.84_{\pm 0.11}$
MP		50%	/	/	$53.93_{\pm 0.39}$	$88.98_{\pm 0.23}$
SMD (ours)	Eve.	50%	/	/	$53.82_{\pm 0.11}$	$88.94_{\pm 0.07}$
LTH†		50%	/	/	$55.72_{\pm 0.37}$	$89.90_{\pm 0.15}$
SMD† (ours)		50%	/	/	$56.52_{\pm 0.60}$	$90.23_{\pm 0.22}$
CDMD		0	$94.74_{\pm 0.77}$	$99.15_{\pm 0.17}$	$53.73_{\pm 0.51}$	$88.83_{\pm 0.17}$
MP		50%	$95.69_{\pm 0.28}$	$99.26_{\pm 0.05}$	$52.82_{\pm 0.24}$	$88.47_{\pm 0.16}$
SMD (ours)	Both	50%	$95.82_{\pm 0.13}$	$99.33_{\pm 0.10}$	$54.44_{\pm 0.11}$	$89.26_{\pm 0.03}$
LTH†		50%	$96.19_{\pm 0.16}$	$99.53_{\pm 0.04}$	$55.69_{\pm 0.21}$	$89.93_{\pm 0.07}$
SMD† (ours)		50%	$96.46_{\pm 0.24}$	$99.51_{\pm 0.03}$	$56.56_{\pm 0.23}$	$90.30_{\pm 0.09}$

4.2 Benchmark on IED/TL17/Crisis Datasets

The benchmark results are show in Tab. 4 (IED), Tab. 5 (TL17), and Tab. 6 (Crisis). We consider three tracks here: document (*i.e.*, the model can only conduct document-level detection), event, and

Table 7: Model complexity (representation bits, FLOPs, and wall-clock speedup) of different methods to achieve misinformation detection at *both the document and event* levels. CDMD-fuse is a scheme of Wu et al. (2022) that they feed the results of event-level results to document-level detector for improved performance. †We consider the training and testing FLOPs of *one* CDMD model as *one unit*. The FLOPs and wall-clock speedup (relative to the baseline cost of CDMD) are estimated based on the sparse-dense matrix multiplication in fully-connected layers. *Environment: Ubuntu 2004, GPU NVIDIA RTX 3090, CMake 3.18.4.

Method	Sparsity	Repre. bits (M)↓	FLOPs (G, train)↓	FLOPs (G, test)↓	Wall-clock speedup* ↓
CDMD	0	6105.6	2 units†	2 units†	1
CDMD-fuse	0	6105.6 (1×)	2.5 units (1.25×)	2 units (1×)	1.0×
MP	75%	763.2 (0.125×)	1 unit (0.5×)	1 unit (0.5×)	1.1×
LTH	75%	763.2 (0.125×)	2 units (1×)	1 unit (0.5×)	1.1×
SMD (ours)	75%	763.2 (0.125×)	0.25 unit (0.125×)	0.25 unit (0.125×)	7.4×

both (*i.e.*, the model can only conduct misinformation detection at both levels). The last track is of our most interest since we target one model to solve the two-level misinformation detection

As seen, our method consistently *outperforms* the other methods (HDSF, GROVER, CDMD, and MP) *by an obvious margin* in most cases, especially at the "Level=Both" rows. One minor exception is on the Crisis dataset, at event level, our SMD is slightly worse (*e.g.*, F-1 53.82 of ours *vs.* 54.08 of CDMD), yet the gap is not significant as the stddev implies. Meanwhile, note that our approach can achieve comparable or even better performance than all the dense CDMD models (no matter it is trained for document/event level alone or for both levels), while *with 50% to 75% sparsity*.

The LTH method sometimes surpasses our SMD because it uses a pretrained model to obtain masks (which can be considered as a kind of extra training (Zhou et al., 2019)). For a fair comparison, we also include the case of using pretrained model for our SMD method (SMD†). As seen, with a pretrained model, our SMD† also outperforms LTH consistently.

4.3 Model Complexity Comparison

In addition to the performance comparison in Sec. 4.2, here we also report the model complexity comparison, in two axes: presentation bits (*i.e.*, space complexity) and FLOPs (*i.e.*, time complexity). FLOPs of training considers all the computation needed to get the final model (if a pretrained model is needed, such as in LTH (Frankle & Carbin, 2019), then the pretraining cost is also considered for a fair comparison). As shown in Tab. 7, our method achieves the best performance while enjoying the smallest model size and least computation.

Wall-clock speedup comparison. To show the proposed CircuSparsity is useful in *practical* applications, we further conduct wall-clock comparison with a large matrix-multiplication operation[2], which is the element operation in the GNN model (the GNN model is made up with GAT layer, which essentially consists of multiple fully-connected layers). The results are also included in Tab. 7. The actual speedup of the sparsity by MP and LTH is very marginal because their unstructured sparsity is irregular and hard to be used for acceleration on the general-purpose GPUs. In contrast, since the sparsity pattern in our method is carefully pre-defined, with a regular structure, it can harvest considerable acceleration very easily by using the off-the-shelf libraries.

5 Conclusion

A new Sparse Misinformation Detector (SMD) is presented in this paper for efficient misinformation detection. SMD has two critical designs: CircuSparsity and early exit. CircuSparsity is a newly proposed *predefined* fine-grained sparsity pattern that can save model footprint and also gain practical acceleration. Early exit of the document-level classifier is proposed based on dedicated sensitivity analyses of the document-and-event misinformation detection – the document-level detection is much more *insensitive* to compact model size than the event-level detection. Empirically, our SMD can achieve 50% to 75% sparsity while still *significantly outperforming* the dense counterpart network and other counterpart sparse training approaches.

[2]We exploit the NVIDIA `cuSPARSELt` library using Sparse Tensor Cores, referring to the library samples at https://github.com/NVIDIA/CUDALibrarySamples/tree/master/cuSPARSELt/spmma

ETHICS STATEMENT

We strictly follow the ICLR General Ethical Principles. All datasets we employed are publicly available, and all related publications and source codes are cited appropriately. We see *no potential negative societal impacts* from our paper.

REPRODUCIBILITY STATEMENT

Our code cannot be released at the review stage because of company-owned IP. Yet *we promise to release all the code, logs, and trained models upon paper acceptance*.

We strictly adhere to ICLR reproducibility standards and do our best to ensure the reproducibility of our work in multiple ways, including but not limited to:

- We strictly follow the standard implementation of previous methods using their officially released code, with no change of hyper-parameters.
- All the new results presented in our paper are obtained with the same code and same software environment.
- To mitigate the influence of random variations, the benchmark and ablation study results are obtained by 3 random times, mean and stddev reported.
- Detailed experimental settings are clearly documented in the paper (Sec. 3.3 and Sec. 4).

REFERENCES

Shivangi Aneja, Chris Bregler, and Matthias Nießner. Cosmos: Catching out-of-context misinformation with self-supervised learning. *arXiv preprint arXiv:2101.06278*, 2021. 1, 2

Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020. 3

Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *SIGKDD*, 2006. 3

Guobin Chen, Wongun Choi, Xiang Yu, Tony Han, and Manmohan Chandraker. Learning efficient object detection models with knowledge distillation. In *NeurIPS*, 2017. 3

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019. 3

M. Courbariaux and Y. Bengio. BinaryNet: Training deep neural networks with weights and activations constrained to $+1$ or -1. *arXiv preprint arXiv:1602.02830*, 2016. 3

Matthieu Courbariaux, Itay Hubara, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks: Training deep neural networks with weights and activations constrained to +1 or -1. *arXiv preprint arXiv:1602.02830*, 2016. 3

Emily L Denton, Wojciech Zaremba, Joan Bruna, Yann LeCun, and Rob Fergus. Exploiting linear structure within convolutional networks for efficient evaluation. In *NeurIPS*, 2014. 3

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 3

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *ICLR*, 2019. 2, 6, 7, 8, 9

Yi Fung, Christopher Thomas, Revanth Gangi Reddy, Sandeep Polisetty, Heng Ji, Shih-Fu Chang, Kathleen McKeown, Mohit Bansal, and Avirup Sil. Infosurgeon: Cross-media fine-grained information consistency checking for fake news detection. In *ACL-IJCNLP*, 2021. 1, 2

Trevor Gale, Erich Elsen, and Sara Hooker. The state of sparsity in deep neural networks. *arXiv preprint arXiv:1902.09574*, 2019. 5

Robert M Gray et al. Toeplitz and circulant matrices: A review. *Foundations and Trends® in Communications and Information Theory*, 2(3):155–239, 2006. 4

Qipeng Guo, Xipeng Qiu, Pengfei Liu, Yunfan Shao, Xiangyang Xue, and Zheng Zhang. Star-transformer. In *NAACL*, 2019. 3

Song Han, Jeff Pool, John Tran, and William J Dally. Learning both weights and connections for efficient neural network. In *NeurIPS*, 2015. 5, 6, 7, 8

Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. In *ICLR*, 2016. 5, 6, 7, 8

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *NeurIPS Workshop*, 2014. 3

Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. *arXiv preprint arXiv:1905.02244*, 2019. 3

Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 3

Linmei Hu, Tianchi Yang, Luhao Zhang, Wanjun Zhong, Duyu Tang, Chuan Shi, Nan Duan, and Ming Zhou. Compare to the knowledge: Graph neural fake news detection with external knowledge. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 754–763, 2021. 1, 2

Kung-Hsiang Huang, Mu Yang, and Nanyun Peng. Biomedical event extraction with hierarchical knowledge graphs. In *EMNLP*, 2020. 4

Kung-Hsiang Huang, ChengXiang Zhai, and Heng Ji. Concrete: Improving cross-lingual fact-checking with cross-lingual retrieval. *arXiv preprint arXiv:2209.02071*, 2022. 1, 2

Max Jaderberg, Andrea Vedaldi, and Andrew Zisserman. Speeding up convolutional neural networks with low rank expansions. In *BMVC*, 2014. 3

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. Tinybert: Distilling bert for natural language understanding. *arXiv preprint arXiv:1909.10351*, 2019. 3

Yiqiao Jin, Xiting Wang, Ruichao Yang, Yizhou Sun, Wei Wang, Hao Liao, and Xing Xie. Towards fine-grained reasoning for fake news detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 5746–5754, 2022. 1, 2

Hamid Karimi and Jiliang Tang. Learning hierarchical discourse-level structure for fake news detection. In *NAACL*, 2019. 6, 7, 8

Hamid Karimi, Proteek Roy, Sari Saba-Sadiya, and Jiliang Tang. Multi-source multi-class fake news detection. In *Proceedings of the 27th international conference on computational linguistics*, pp. 1546–1557, 2018. 1, 2

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 6

Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. In *ICLR*, 2020. 3

Vadim Lebedev, Yaroslav Ganin, Maksim Rakhuba, Ivan Oseledets, and Victor Lempitsky. Speeding-up convolutional neural networks using fine-tuned cp-decomposition. *arXiv preprint arXiv:1412.6553*, 2014. 3

Namhoon Lee, Thalaiyasingam Ajanthan, and Philip Torr. Snip: Single-shot network pruning based on connection sensitivity. In *ICLR*, 2019. 2

Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. In *ICLR*, 2017. 5

Manling Li, Sha Li, Zhenhailong Wang, Lifu Huang, Kyunghyun Cho, Heng Ji, Jiawei Han, and Clare Voss. The future is not one-dimensional: Complex event schema induction by graph modeling for event prediction. In *EMNLP*, 2021. 5

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. A joint neural model for information extraction with global features. In *ACL*, 2020. 3

Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *ECCV*, 2018. 3

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019. 6

Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. In *ECCV*, 2016. 3

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018. 3

Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, 2019. 3

Reuben Tan, Bryan Plummer, and Kate Saenko. Detecting cross-modal inconsistency to defend against neural fake news. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2081–2106, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.163. 1, 2

Hidenori Tanaka, Daniel Kunin, Daniel L Yamins, and Surya Ganguli. Pruning neural networks without any data by iteratively conserving synaptic flow. In *NeurIPS*, 2020. 2

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 3, 4

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *ICLR*, 2018. 3

Chaoqi Wang, Guodong Zhang, and Roger Grosse. Picking winning tickets before training by preserving gradient flow. In *ICLR*, 2020a. 2, 5

Huan Wang, Yijun Li, Yuehai Wang, Haoji Hu, and Ming-Hsuan Yang. Collaborative distillation for ultra-resolution universal style transfer. In *CVPR*, 2020b. 3

Huan Wang, Can Qin, Yulun Zhang, and Yun Fu. Neural pruning via growing regularization. In *ICLR*, 2021. 5

Huan Wang, Can Qin, Yulun Zhang, and Yun Fu. Recent advances on neural network pruning at initialization. In *IJCAI*, 2022. 2

Lin Wang and Kuk-Jin Yoon. Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks. *TPAMI*, 2021. 3

William Yang Wang. " liar, liar pants on fire": A new benchmark dataset for fake news detection. *arXiv preprint arXiv:1705.00648*, 2017. 1, 2

Wei Wen, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. Learning structured sparsity in deep neural networks. In *NeurIPS*, 2016. 3

Xueqing Wu, Kung-Hsiang Huang, Yi Fung, and Heng Ji. Cross-document misinformation detection based on event graph reasoning. In *NAACL*, 2022. 1, 2, 3, 4, 5, 6, 7, 8, 9

Ruichao Yang, Xiting Wang, Yiqiao Jin, Chaozhuo Li, Jianxun Lian, and Xing Xie. Reinforcement subgraph reasoning for fake news detection. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 2253–2262, 2022. 1, 2

Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. Qa-gnn: Reasoning with language models and knowledge graphs for question answering. In *NAACL*, 2021. 4

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. In *NeurIPS*, 2020. 3

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. Defending against neural fake news. In *NeurIPS*, 2019. 1, 2, 6, 7, 8

Xiangyu Zhang, Jianhua Zou, Xiang Ming, Kaiming He, and Jian Sun. Efficient and accurate approximations of nonlinear convolutional networks. In *CVPR*, 2015. 3

Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *CVPR*, 2018. 3

Hattie Zhou, Janice Lan, Rosanne Liu, and Jason Yosinski. Deconstructing lottery tickets: Zeros, signs, and the supermask. In *NeurIPS*, 2019. 9