Domain-Agnostic Adapter Architecture for Deception Detection: Extensive Evaluations with the DIFrauD Benchmark

Dainis Boumber, Fatima Zahra Qachfar, Rakesh M. Verma
University of Houston
{dboumber, fqachfar, rverma}@uh.edu

Abstract

Despite significant strides in training expansive transformer models, their deployment for niche tasks remains intricate. This paper delves into deception detection, assessing domain adaptation methodologies from a cross-domain lens using transformer Large Language Models (LLMs). We roll out a new corpus with roughly 100,000 honest and misleading statements in seven domains, designed to serve as a benchmark for multidomain deception detection. As a primary contribution, we present a novel parameter-efficient finetuning adapter, *PreXIA*, which was proposed and implemented as part of this work. The design is model-, domain- and task-agnostic, with broad applications that are not limited by the confines of deception or classification tasks. We comprehensively analyze and rigorously evaluate LLM tuning methods and our original design using the new benchmark, highlighting their strengths, pointing out weaknesses, and suggesting potential areas for improvement. The proposed adapter consistently outperforms all competition on the DIFrauD benchmark used in this study. To the best of our knowledge, it improves on the state-of-the-art in its class for the deception task. In addition, the evaluation process leads to unexpected findings that, at the very least, cast doubt on the conclusions made in some of the recently published research regarding reasoning ability's unequivocal dominance over representations quality with respect to the relative contribution of each one to a model's performance and predictions.

Keywords: Deception Detection, Parameter Efficient Learning, Domain Adaptation, Deep Learning

1. Introduction

Deception in linguistic communication signifies the intentional act of inducing false beliefs. Deception detection (DD) employs computational techniques to distinguish between truthful and deceptive statements. Although it is generally considered a binary classification task, DD can classify messages into various levels of deception. Its relevance has surged, exacerbated by the escalating necessity for datasets and detectors optimized for the ever-growing domains where deceptive language is prevalent. Existing research focuses predominantly on specific fields, which results in the need for clarity on linguistic markers associated with deception. We hypothesize that universal features underlie many deception tasks in distinct domains.

One of the significant obstacles in this research field is the lack of large, quality multidomain datasets. To address this issue and further advance the study of deception detection, we introduce the Domain-Independent Fraud Detection Benchmark (DIFrauD). This carefully curated and expansive multidomain corpus contains deceptive texts, statements, and claims. DIFrauD is publicly available through Huggingface datasets.[1] DIFrauD offers a finite set of domains and tasks; in the real world, these elements evolve. Recognizing this dynamic nature, our goal is to devise a strategy to create a model that can universally detect deception, regardless of the domain or task.

This study also addresses the broader issue of the research gap around knowledge transfer when multiple labeled source datasets are involved. This problem first became evident in sentiment analysis (Ruder and Plank, 2017), but it became especially noticeable as adapters (Pfeiffer et al., 2020a), transformers (Devlin et al., 2019; Peters et al., 2019), and LLMs grew in popularity. Early attempts to address this issue relied on multitask learning (MTL), which combines datasets during training, driving the model to find a shared optimal space for all tasks (Arumae et al., 2020). Another approach involves training a language model (LM) with in-domain data (Howard and Ruder, 2018). Although this yields a flexible model without domain alignment issues, its implementation is complex, often cost-ineffective (Tay et al., 2021), and rarely scalable. In-context learners (ICL) (Akyürek et al., 2022) and TART (Bhatia et al., 2023) are promising alternatives because they are task-agnostic and do not require training. However, task-agnostic models are rarely deployed in practice compared to parameter-efficient tuning methods (Ding et al., 2023; Liu et al., 2022). Given the pros and cons of each method, the optimal choice remains problem-dependent. Our work seeks to bridge the aforementioned gap by using the DIFrauD benchmark to revisit existing methodologies and evaluate the feasibility of solving the problem of seamlessly adapting a learner to multiple domains using newer methods

[1] `https://huggingface.co/datasets/redasers/difraud`

LREC-COLING 2024, pages 5260–5274
20-25 May, 2024.

such as TART and adapters. The contributions of this paper include the following:

1. A novel multidomain language resource for public use and to serve as a deception detection benchmark. To our knowledge, no similar resource of comparable magnitude and coverage is currently readily available.
2. An original PEFT (Parameter-Efficient Finetuning) adapter design[2] that consistently outperforms existing adapters and other methods when evaluated on DIFrauD.
3. Benchmark of a comprehensive collection of transformer training strategies and the proposed adapter on DIFrauD to (a) produce an established performance baseline for deception detection and (b) gain further insight into each method's respective advantages and disadvantages and when each is an appropriate choice.

2. Related Work

Deception detection (DD) has historically been explored within individual tasks and domains. The primary reason for such a focused approach was the absence of comprehensive datasets or models tailored for deception detection (DD).

2.1. Approaches to Deception Detection

Initial efforts of (Jindal and Liu, 2008) employed logistic regression that incorporated features centered on products, reviews, and reviewers. (Ott et al., 2011) relied on n-gram features, Naïve Bayes (Rish et al., 2001) and SVM (Boser et al., 1992) classifiers. Other early methods include the use of part-of-speech tags, context-free grammar parse trees, and spatial-temporal attributes (Feng et al., 2012; Mukherjee et al., 2013; Li et al., 2015). The hand-crafted features have retained their relevance. In multimodal DD, trial video and audio have been used in addition to transcripts (Pérez-Rosas et al., 2015); meanwhile, visual, thermal, and physiological characteristics were explored in (Abouelenien et al., 2015).

2.2. The Deep Learning Revolution

In contemporary times, deep learning has eclipsed traditional machine learning techniques across domains, particularly in deception detection. (Ceron et al., 2020) distinguished fake news using topic models, marking a departure from earlier supervised ML methods, which were then the norm for phishing detection. The emergence of deep learning models, such as RCNN (Fang et al., 2019) and those that harness the embeddings of BERT (Devlin et al., 2019) and Sentence-BERT (Reimers and Gurevych, 2019; Shahriar et al., 2022b), have established new benchmarks. Furthermore, linguistic transfer and representation learning have enabled groundbreaking advances (Ren and Ji, 2017; Zhang et al., 2018; Hamid et al., 2020).

2.3. Cross-Domain Deception Detection

Researchers like (Rill-García et al., 2018; Sánchez-Junquera et al., 2020; Hernández-Castañeda et al., 2017) have been at the forefront of exploring cross-domain deception detection. Although the allure of a universally applicable solution to deception challenges is undeniable, its feasibility hinges on the existence and transferability of domain-independent linguistic markers of deception. However, this notion is still debated, with some studies, like (Gröndahl and Asokan, 2019), positing the absence of universal stylistic deception markers.

However, recent evaluations contradict this belief. For example, when adequately finetuned, (Zeng et al., 2022) demonstrated that BERT could efficiently detect deception across multiple domains. Concurrent studies also revealed the utility of psychological attributes in phishing detection (Shahriar et al., 2022b) and showed that specific sources related to deception significantly improve performance (Shahriar et al., 2022a). Such findings suggest the existence of some transferable domain-independent traits.

2.4. Data Selection and Transfer Learning

The intricacies of choosing the right source domain for deception detection aren't unique to the field. Effective data selection for adaptation is crucial in multitask learning, as documented by (Ruder and Plank, 2017). The emergence of large language models (LLMs) such as (Peters et al., 2018; Radford et al., 2018) enabled remarkable results with minimal domain-specific labeled data. Furthermore, domain adaptation (DA) strategies, such as in-domain pretraining, have shown promise (Howard and Ruder, 2018). When applied across multiple domains, it is known as continued pretraining (Ring, 1997). Unfortunately, this promising approach can lead to catastrophic forgetting (McCloskey and Cohen, 1989; Ratcliff, 1990).

Adapting large transformer models to domain-specific vocabularies has been successful (Yao et al., 2021; Zhang et al., 2020). Recent work (Bhatia et al., 2023) has shifted focus from improving representations to fortifying the reasoning capabilities of transformers. Despite initial skepticism about the adaptability of transformers (Wright and

[2] To the best of our knowledge, this design has not been proposed in the literature.

Augenstein, 2020), adapters have emerged as cost-effective solutions, strengthening the adaptability of these models (Pfeiffer et al., 2020a; Houlsby et al., 2019). Recent work has used adapters in domain, task, and language transfer (Pfeiffer et al., 2020b).

2.5. Training LLMs

This subsection provides a comprehensive overview of the foundational methodologies employed in language model training and inference processes, all of which will be evaluated for applicability in multidomain deception detection tasks. These methodologies can be grouped into six distinct techniques:

Linear Probing (LP): Linear Probing trains linear classification models without dependency on their foundational embeddings being finetuned: a base LLM encodes text into embeddings that serve as input for the linear model. (Conneau and Kiela, 2018; Yu et al., 2022).

Finetuning: This technique adapts the layers of the pretrained language model to a specific task using labeled data from the target domain. Although commonly associated with BERT (Devlin et al., 2019), the essence of this method precedes it. ULMFit finetuned an LSTM classifier over a pretrained LSTM dedicated to language modeling a few years earlier, for example, as did ELMo. (Radford et al., 2018; Howard and Ruder, 2018; Peters et al., 2019). The key difference is that with BERT and most later models, finetuning typically involves all or most layers and is often called *Full Finetuning*. In contrast, *Partial finetuning* is done on some of the last layers of an otherwise frozen model. This technique was popular in low-data scenarios that precluded the model from learning to generalize well when fully finetuned. Still, it is rarely used by virtue of having been superseded by PEFT Adapters. Consequently, this method is often referred to simply as "finetuning".

Continued Pretraining (with Full Finetuning): The model initially undergoes self-supervised training on available unlabeled data to familiarize itself with the domain, followed by full finetuning on labeled data (Howard and Ruder, 2018). There is an inconsistency in the nomenclature used by different researchers, most likely caused by the recent practice of using "finetuning" to describe "full finetuning". Consequently, the term often means "continued pretraining followed by full finetuning". For the sake of clarity, we will use "finetuning" to describe "full finetuning", while "pretraining" will refer to "continued pretraining with full finetuning".

Adapter: Also termed Parameter-Efficient Language model Tuning (PELT) or PEFT, adapters are compact learned layers seamlessly integrated into a pretrained model's architecture. Their initial design added a low-rank matrix to the native matrix to expand its capabilities (Houlsby et al., 2019; Pfeiffer et al., 2020a). Modern adapters offer a wide variety of enhancements (He et al., 2022), such as improved efficiency (Rücklé et al., 2021; Hu et al., 2022) predictive prowess (Chen et al., 2023; Li and Liang, 2021), efficient knowledge transfer (Pfeiffer et al., 2021), linguistic adaptation (Pfeiffer et al., 2020b), and dynamic architectural modifications suitable for various tasks (Mao et al., 2022).

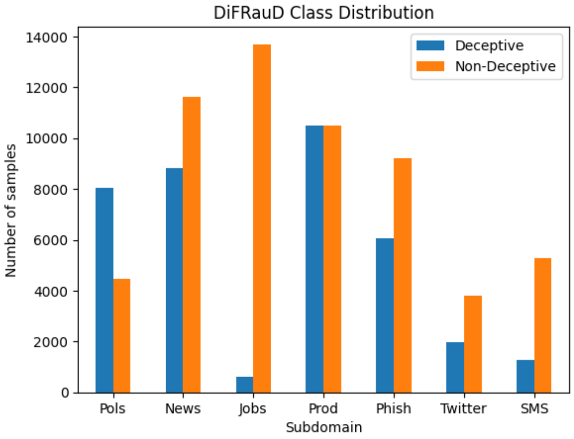

Figure 1: Label distribution across domains

In-Context Learning (ICL): ICL introduces multiple contextual examples into the model using transformer adaptability before making predictions. Choosing learned prompts over standardized ones results in prompt-tuning (Li and Liang, 2021). ICL has been found to depend on linear models that are implicitly learned (Akyürek et al., 2022). The primary distinction between finetuning and prompt-tuning is the objective: while finetuning enhances the model for a task, prompt-tuning uses additional contextual information to influence output generation.

Task-Agnostic Reasoning and Transfer (TART): TART functions independently of specific tasks, domains, or LLMs. Boasting superior accuracy and scalability over in-context learning, TART can be used with any generative LM (Bhatia et al., 2023). The aim is to maximize the potential of LLM by introducing a task-agnostic reasoning module trained on synthetic Gaussians that were used as inputs for binary logistic regression challenges. The LM processes the data and context into the reasoning module for the final predictions.

3. Datasets

DIFrauD: The Domain-Independent Fraud Detection corpus is the first significant contribution from this work. It is a labeled collection of $95,854$ text documents that contain harmless and misleading communications from seven domains. The DIFrauD corpus builds on the Generalized Deception Dataset (GDD) (Zeng et al., 2022) by adding two datasets from distinct disciplines, including more than $1,000$ additional internally labeled phishing email and SMS, while correcting more than $20,000$ mislabeled examples. The corpus encompasses $95,854$ entries in seven domains: "Fake News," "Political Statements," "Job Scams," "Product Reviews," "Emails," "SMS Spam," and "Twitter Rumors," with $37,282$ instances being deceptive and $58,572$ genuine. Figure 1 and Table 4 shows the size and dataset label distribution of each domain.

3.1. Data Preparation

After initially gathering a vast collection of corpora, we focused only on those with potential for cross-domain transfer. For our selection process, we used a set of transferability metrics[3] proven effective for the selection of transfer learning samples (Ruder and Plank, 2017). The metrics are detailed in Table 1.

Similarity	**Divergence**
Jensen-Shannon	Term-Type Count
Renyi Divergence	Token-Type Ratio
Cosine Similarity	Entropy
Euclidean Distance	Simpsons Index
Variational Distance	Renyi Entropy

Table 1: Transferability estimators

Instead of individual sequence selection, we compared entire datasets. We computed pairwise distance matrices between datasets using simple bag-of-words (BOW) embeddings and similarity metrics. The results were standardized and averaged to produce a single number $s_{src,tgt} \in [-1.0, 1.0]$ that represents the similarity between two datasets. Divergence metrics illustrated in Figure 4 are inter-domain and measure how diverse a dataset is in and out of itself.

Cleaning, Preprocessing, and Labeling We identified and rectified $20,000$ problems with existing data using the *Cleanlab* tool (Kuan and Mueller, 2022). This involved labeling unlabeled data, adding the data we collected, finding and correcting datasets with flipped (*Job Scams*) and partially flipped (*Political Statements*) labels, manually correcting mislabeled examples, resolving contradictory labeled duplicates, purging noisy entries caused by crawler/parser errors and removing non-English, null, or poorly encoded content. Two annotators manually verified all modifications; the third one stepped in when they disagreed. Datasets were shuffled and split $80/10/10$ into three sets for training, validation, and testing, stratified according to the target label y.

3.2. Domains

Fake News Incorporating $72,134$ news articles from four datasets (Kaggle, McIntire, Reuters, and BuzzFeed Political), the dataset (Verma et al., 2021) was purged of data leaks such as "[claim] (Reuters)." Duplicate and outlier detection identified many irrelevant samples. The refined Fake News dataset includes $20,456$ articles, with $8,832$ deceptive and $11,624$ genuine.

Political Statements Constructed from the *Liar* dataset (Wang, 2017), political statements made by US speakers received a truthfulness grade from PolitiFact. Following (Upadhayay and Behzadan, 2020; Shahriar et al., 2022a), the categories "pants on fire," "false," "barely true," and "half true" were labeled deceptive or 1, while "mostly true" and "true" were labeled non-deceptive or 0. Outlier detection revealed numerous anomalous statements reminiscent of the headers of political articles without context, for example, "*on the[sic] Iran nuclear deal*" or "*on sequestration*." Lacking sufficient context, these were discarded. This subset now has $12,497$ statements. Of these, $8,042$ are deceptive, and $4,455$ are not.

Job Scams The *Employment Scam Aegean* (Vidros et al., 2017) dataset, hereafter termed the *Job Scams* dataset, originally featured $17,880$ human-annotated job listings. HTML tags, empty content, and duplicates were removed during cleaning. The dataset is imbalanced because it consists of $14,295$ entries, of which only 599 are deceptive.

Product Reviews Originating from the *English Amazon Reviews* [4], entries labeled real or fake were labeled as non-deceptive and deceptive, respectively. Despite the initial non-English filtration, outlier detection revealed lingering non-English reviews. Problematic label detection suggested $6,713$ of them were potentially mislabeled. When examining the top 1% (67 entries, most appeared incorrectly labeled, prompting a lengthy evaluation process. The resulting subset is balanced, with $10,492$ deceptive and $10,479$ non-deceptive entries,

[3] https://github.com/sebastianruder/learn-to-select-data

[4] https://www.kaggle.com/datasets/lievgarcia/amazon-reviews

totaling $20,971$.

Phishing The *Email Benchmarking* dataset (Zeng et al., 2020) encompasses $21,000$ emails: $10,500$ phishing and $10,500$ genuine. It contains additional data compared to GDD. Using only the email body, phishing emails were marked as deceptive, and genuine emails as non-deceptive. Most originate from existing datasets, but just over $1,000$ have been collected and labeled internally. We removed entries with more than $1,000,000$ tokens in the email body due to tokenization issues, and entries with HTML tags, metadata, duplicates, and non-English entries. Ultimately, $6,074$ deceptive and $9,198$ non-deceptive emails remained.

Twitter Rumors The *Twitter Rumors* dataset was formed using the PHEME dataset (Kochkina et al., 2018), spanned several years and six topics. Only origin tweets (new tweets instead of replies to existing ones) were used and labeled. Selecting only posts with verifiable claims yielded $1,969$ deceptive tweets and $3,820$ non-deceptive ones.

SMS Sourced from the *SMS Spam Collection v.1* (Almeida et al., 2011) and the *SMS Phishing Dataset for Machine Learning and Pattern Recognition* (Mishra and Soni, 2023), both datasets overlap and contain contradictory and missing labels. After deduplication, the resulting dataset consists of $6,574$ SMS messages. Of these, 199 SMS messages were incorrectly labeled or lacked any label. With these issues addressed, the collection has $1,274$ deceptive SMS messages and $5,300$ genuine ones.

3.3. Limitations and Biases

A few limitations and biases are inherent in our dataset and may be addressed in future work:

1. *Deceptive Political Statements:* Most of these contain deceptive labels. This bias can unintentionally cause F_1 score to plateau, which may not accurately represent the actual performance of the model.

2. *Job Scams Imbalance:* This data subset has a notable class imbalance. We opted not to modify the data primarily because we believe that the decision to adjust should be at the discretion of individual researchers working with the dataset.

3. *Lack of Certain Data Types:* DIFrauD does not include explicitly unlabeled training data or labeled out-of-domain test data.

4. Methodology

4.1. Metrics

Each subtask is a binary classification problem. For all but one subtask, the label of interest is in a significant minority. Following established DD practices, we use the *binary* F_1 score as our primary metric to gauge the performance of the model. The F_1 score is a harmonic mean of precision and recall, expressed as:

$$F_1 = 2 * \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2 * TP}{2 * TP + FP + FN}$$

When performing binary classification tasks, it is necessary to properly designate a *positive* class to calculate the binary F_1 score. For our scenario, the *positive* class is "deceptive," and the *negative* class is "not deceptive." This is a critical specification with far-reaching consequences: the binary F_1 score does not account for *True Negatives* (TN); instead, it emphasizes the model's ability to detect and accurately identify members of the *positive* class as the most important for the task. Also of note is the fact that an incorrect designation could lead to the exclusion of true positives from the computations. Although this metric is optimal for detecting harmful samples within a large population, it may exhibit instability when the data distribution strongly favors the positive class, mainly because the $loss$ function remains oblivious to the binary F_1 score. It is essential to remember that the binary version of the F_1 score has different limitations and serves a different purpose than the *macro* F_1 score, which computes the combined F_1 score of all classes taking their unweighted average, thus treating them equally regardless of their practical importance to the task at hand.

In line with (Ceron et al., 2020; Bethu et al., 2019), we adopt robust statistical methodologies. We ensure significance with an independent test sample where $N >> 100$. Furthermore, non-deterministic operations are consistently initialized to guarantee reproducibility with a random seed value of 42.

4.2. PreXIA

Inspired by the flexibility of adapter modules, we introduce *PreXIA*: domain and task adapter. Our design is not specific to deception, binary classification, or BERT alone, even though we use them for evaluation for consistency's sake. *PreXIA* is compatible with any Transformer architecture that uses typical self-attention via being incorporated into any layer with a Transformer block.

PreXIA's adaptability is distinctively design-driven. This choice comes from the occasional absence of certain task parameters and domain specifics essential for optimal design decisions,

Method	Data	Pols	News	Jobs	Prod	Phish	Twitter	SMS	Mean
lp+pooled	tgt	0.7658	0.9043	0.1230	0.6809	0.9527	0.7018	0.9922	0.7315
lp+cls	tgt	0.7611	0.9265	0.4000	0.6792	0.9560	0.7050	**0.9961**	0.7748
finetune	tgt	0.7640	0.9773	0.6729	0.6968	0.9859	0.7944	0.9883	0.8399
finetune	all	0.7315	0.9731	0.5636	0.6975	0.9705	0.8030	0.9732	0.8161
finetune	bal	0.7516	0.9744	0.7350	**0.7215**	0.9810	0.7719	0.9845	0.8457
pt+ft	all+tgt	0.7649	0.9774	0.6867	0.6955	0.9891	0.8130	0.9807	0.8439
pt+ft	all+all	0.7287	0.9743	0.5741	0.6883	0.9685	0.8000	0.9730	0.8153
pt+ft	all+bal	0.7719	0.9795	**0.7667**	0.7159	0.9867	0.7942	0.9883	0.8576
Pfeiffer	tgt	0.7773	0.9853	0.6458	0.6882	0.9734	0.7469	0.9677	0.8264
UniPELT	tgt	0.7794	0.9819	0.7368	0.7137	0.9843	0.8206	0.9766	0.8562
MAM	tgt	0.7841	0.9789	0.7455	0.7110	0.9867	0.8333	0.9760	0.8594
PreXIA	tgt	**0.7881**	**0.9881**	0.7350	0.7206	**0.9896**	**0.8511**	0.9723	**0.8635**
pt+PreXIA	all+tgt	0.7756	0.9759	0.6981	0.7095	0.9814	0.8358	0.9594	0.8480

Table 2: Binary $F_1-scores$ achieved by $bert-base-uncased$ using various adaptation strategies with a single target and six sources. Methods: lp=linear probing, ft=finetune, pt=pretrain; Data trained on: all=combined training data from all domains; tgt=target domain itself; bal=balanced sample drawn from each domain. *PreXIA* is the novel adapter proposed in this paper. The best result for each domain is depicted in the **bold** font.

Method	Base	Embed	Pols	News	Jobs	Prod	Phish	Twitter	SMS	Mean
ICL	GPT-125M	base	0.5792	0.6094	**0.5263**	0.4681	0.5555	0.6038	0.5714	0.5591
TART	GPT-125M	base	0.5002	0.6749	0.1246	0.5638	0.8194	0.5350	0.7727	0.5701
TART	GPT-125M	stream	0.6542	0.8876	0.1879	0.5835	0.9333	**0.7160**	**0.9904**	**0.7076**
TART	GPT-125M	loo	0.6272	**0.8937**	0.1441	**0.5846**	**0.9377**	0.6608	0.9828	0.6901
TART	Pythia-14M	stream	**0.6549**	0.7195	0.1218	0.5818	0.8131	0.6112	0.8066	0.6156

Table 3: Performance metrics showcasing binary $F_1-scores$ achieved by TART and ICL methods with various models and embedding options. The best result for each domain is depicted in the **bold** font.

because they emerge only during training. A dynamic and reconfigurable design offered a natural remedy to this problem. We considered several strategies: i) combining multiple components for increased versatility over a singular static adapter in a Mix-And-Match (MAM) adapter style (He et al., 2022); ii) manipulating data flow through stacked components, a method shown to excel in cross-lingual transfer (Pfeiffer et al., 2020b); iii) dynamically updating modules by toggling components during training (Mao et al., 2022); iv) merging components across tasks or domains (Pfeiffer et al., 2021).

PreXIA's design features three main components that run parallel to the attention layer and give the adapter its name:

- *Prefix-tuning (**Pre**):* A re-parametrized bottleneck MLP for prefix tuning (Li and Liang, 2021).
- *Parallel (**X**):* A parallel-scaled bottleneck adapter, as detailed in (He et al., 2022).
- (**IA**)^3: An adapter that inhibits and amplifies inner activations (Liu et al., 2022).

Figure 2 shows the high-level design of the proposed adapter. Within this diagram, the symbols G_p, G_i, and G_x represent the XOR gate G, which chooses between the *Parallel Adapter*, $(IA)^3$, and the *Prefix-tuner*. $(IA)^3$ encompasses the learned layers L_k, L_v, and L_{ff}, while the *Prefix-tuner's* MLP produces the reparametrized outputs P_k and P_v. The module incorporates "Pre-X-IA" with a gate, as shown in Equation 1.

$$G = (IA)^3 | Prefix - tuned | Parallel \qquad (1)$$

This gate is implemented through a two-layer perception (MPN) (Singhal and Wu, 1988) that learns to activate the blocks most appropriate for the current data or task. As the adapter produces beneficial output, the MPN learns to route the gate input to the output, amplifying the influence of the appropriate components.

5. Experiments

5.1. Baselines

LP (Linear Probing): For BERT, we derived the embeddings in two distinct ways: by extracting the

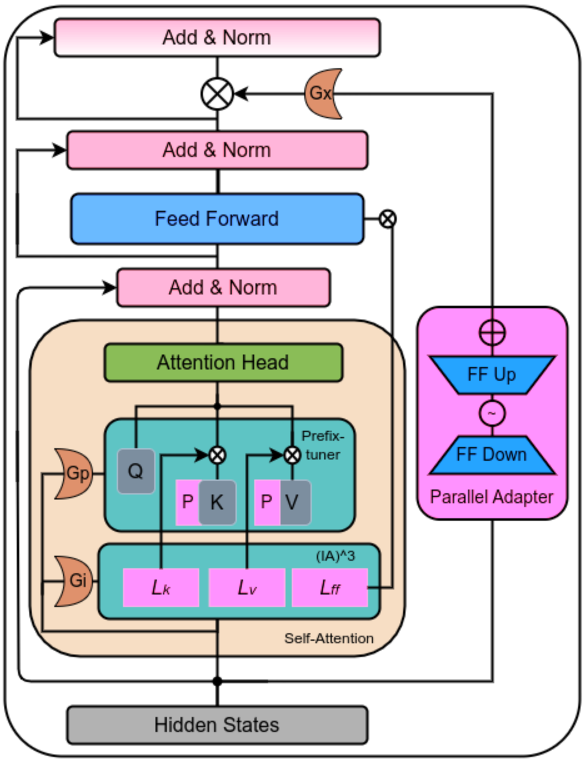

Figure 2: Illustration of the *PreXIA* adapter with trained components highlighted in magenta.

`[CLS]` token from the last layer and by channeling the final hidden representations through a dense $tanh$ layer. Using a logistic regression classifier, linear probing can be used to benchmark the base LLM's capacity to generate quality representations.

ICL (In-Context Learning): We used the *Base* embedding scheme with the GPT-Neo-125M (Black et al., 2022) model and made ICL identical to TART in all but the reasoning head to make the comparison relevant.

5.2. Training Setup

Our approach mainly relied on established hyperparameter recommendations. We adhered to the literature guidelines or default model settings, only deviating when our preliminary hyperparameter search revealed notable performance discrepancies on a given dataset.

Finetuning: As our foundational transformer, we apply BERT-base-uncased (Devlin et al., 2019) universally, except for ICL scenarios. Abiding by the original publication's hyperparameters, we conducted a concise hyperparameter search on the dataset to affirm the efficacy of the default values. The parameters included $AdamW$ optimizer, learning rate $lr = 5e - 2$, dropout rate $dropout = 0.1$, $warmup\_steps = 10$, $weight\_decay = True$, and batch size of $bs = 64$. The maximum sequence length was capped at 512 tokens. We fine-tuned for 5 epochs, saving the best model. This method denoted as ft in Table 2.

Pretraining: Our pretraining approach denoted as pt in Table 2 largely mirrored the finetuning setup, except for a lower learning rate ($lr = 5e - 6$). Instead of truncating the texts, they were merged and divided into chunks of $max\_sequence\_len = 512$, where only the last chunk required padding. This strategy prevented the loss of potentially useful data and provided the model with an improved context source. The sole training objective was MLM (Masked Language Modeling). After eight training epochs, the model that exhibited the lowest *perplexity* was retained.

TART (Task-Agnostic Reasoning and Transfer): For most of our experiments, we used *GPT-Neo-125M* (Black et al., 2022) as our foundational model, adhering to the TART hyperparameters and training instructions provided in (Bhatia et al., 2023). The synthetic reasoning module is based on *GPT-2* (Radford et al., 2019), which underwent training on various artificial logistic regression tasks. In specific experiments, we substitute *GPT-Neo-125M* with *Pythia-14M* (Biderman et al., 2023) to assess the impact of the reasoning component and gauge the impact of the lower capacity representations.

Adapters: We combined each adapter with BERT-base-uncased for comparability. Guided by (Pfeiffer et al., 2020a), we found that adapters produced optimal performance at a learning rate of $lr = 1e-4$. Our investigations led us to set the optimal sequence length at $max\_sequence\_len = 128$, contrasting the complete model, which performed best with a maximum sequence length of 512. After 15 training epochs, the performance of each adapter on the validation set was used to select the model to be evaluated on the test data.

5.3. Training Set Considerations

We performed the experiments with the established $train$, $test$, and $validation$ sets described in Table 5 in the supplementary material.

5.3.1. Learning Sequence Classification

The intuitive strategy is to train a model directly on the target dataset's T_{train} set. This approach is denoted as tgt in the $data$ column of Table 2.

Another straightforward strategy is to train a model on all available data without domain-specific adjustments, effectively treating it as originating from a singular domain. Thanks to *LLM's* representational capacity, this approach will work for some

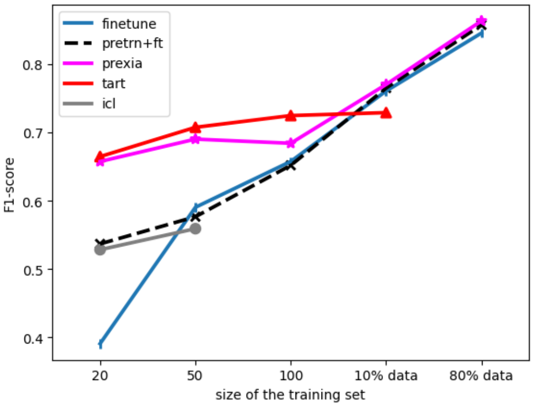

Figure 3: Effect of the number of training examples on different models. Binary $F_1-scores$ are averaged across domains.

time, but naive retraining as new domains appear is not a cost-effective solution, and bound to suffer from catastrophic forgetting sooner rather than later. This strategy is marked all in Table 2.

A balanced approach that draws training data from all sources equally can be instrumental in crafting a versatile model that does not exhibit bias toward any particular domain. The results of it are tagged bal in the $data$ column of Table 2.

5.3.2. Low Availability Resources

We studied the impact of the size of the training set on the best-performing models by systematically varying the composition of the training set for each domain in five steps: three balanced sets of 20, 50 and 100 samples, followed by two splits that contain 10% and 80% of the domain. Splits were randomly sampled and stratified according to the target label y. The average F_1 score calculated in all domains for five sizes of training sets is shown in Figure 3.

5.3.3. In-Context Training Data

TART and ICL must be task-agnostic to be used in practice, so we did not finetune their base models. The inputs were converted to prompts and paired into training pairs (x, y) as determined by the embedding scheme specified in the $embedding$ column of Table 3. For TART and *ICL*, the embedding schemes used are as follows:

- **LOO (Leave-one-out):** Aims to maximize data variation by balancing the resampling of the $train$ dataset. Data is condensed to 16 dimensions using *PCA* and added to the input training sample.
- **Stream (One-by-one):** A strategy where k samples are drawn sequentially from the training set and integrated into the embedding. These embeddings are fed into the reasoning head. The values of $k = [18, 32, 48, 64, 96, 128]$ are used in (Bhatia et al., 2023).
- **Base (Vanilla):** A straightforward approach in which the foundational model is directly used to derive the embedding.

A limitation driven by the width of the reasoning head caps the maximum number of context examples for all embedding schemes except *stream* at 64 to 256 examples, depending on context length (assuming the default maximum sample length of 100 characters). This is expected as in-context learners are constrained by context length to some degree. TART adopts a subsampling approach from the training set to collect training data, ensuring a balanced representation of classes.

6. Results and Discussion

Across the board, *PreXIA* emerges as the top performer with an average binary F_1 score of 0.8635 in seven domains. It remains a top-tier contender when the data available for training is limited. The other PELT models follow it closely. Tables 2 and 3 provide a comprehensive empirical assessment of baseline evaluations, while the few-shot performance of the model is shown in Figure 3. Interestingly, models with a binary classification layer trained on prior data generally outshine encoders in many domains, except for SMS. Here, LP, TART, and ICL take the lead. Potential overfitting by other models might explain this discrepancy.

Finetuning and In-domain pretraining with finetuning achieve binary F_1 scores of 0.8457 and 0.8576, respectively. However, each method uses approximately *100 times more* parameters for inference, and pretraining in each domain took an average of 58 minutes and 21 seconds using a $T4$ GPU with $16GB$ of RAM. Meanwhile, adapter training from scratch took *less than a minute* for each domain. Our results show that adapters may be better suited for most practical tasks.

6.1. Handling Imbalanced Datasets

TART and LP exhibit vulnerability to imbalanced datasets, where ICL significantly outperforms both with the same base model as TART. LP and TART use static embeddings with probabilistic logistic regression, and we hypothesize that this combination has a flaw in the ability to reason about imbalanced data that is not present in ICL. The consequence of using less expressive representations is further underscored by the noticeably lower scores of *Pythia-14M* compared to *GPT-Neo-125M* (Table 3). Consequently, TART may require a larger LM or finetuning to perform on imbalanced data, negating its primary advantage.

6.2. Low-Resource Learning

As shown in Figure 3, ICL and especially TART stand out in such scenarios. It faces stiff competition from *PreXIA*, which reaffirms its resilience by significantly outclassing fully finetuned models when training data comprise fewer than 100 labeled examples. With only 20 labeled examples, TART achieves an average binary F_1 score of 0.6647, with *PreXIA* delivering an impressive 0.6575. Finetuning with this sample size results in a score of 0.3906. The consistent superior binary F_1 scores of TART and its task-agnostic nature make it the natural choice in this context; however, while other models see a progressive increase in their binary F_1 scores with more training data, TART peaks at approximately $1,000$ training samples for the *stream* embeddings and 256 samples for the *LOO* and *Base* schemes. Because we kept the base models, the embedding scheme and the initial setup for TART and ICL identical for some of the experiments, it is possible to use Table 3 to approximate the impact of the reasoning head. We simply subtract the mean binary F_1 score achieved by ICL from that of TART *with base embeddings* and observe that the binary F_1 score of TART, 0.5701, is only marginally higher than the 0.5591 achieved by ICL. It is not unreasonable to hypothesize that the reasoning module is responsible only for 0.0110 increase in the F_1 score, which is underwhelming compared to 0.1375 obtained by switching the embedding scheme. More importantly, this finding contradicts the conclusions made in (Bhatia et al., 2023) and warrants further investigation. At the moment, we can only hypothesize that the answer in the debate regarding the importance of representations' quality vs. the LLM's ability to reason about the information embedded in the representations it is given may not be in favor of either, but depend on factors like data distribution, task, domain, model used, and a host of other possible factors.

6.3. Error Analysis

Product Review As shown in Table 2, the lowest F_1 score (0.7215) was attained on the Amazon product review dataset (He and McAuley, 2016) making it the hardest and most challenging deception domain in the DIFrauD deception dataset. The difficulty in this particular domain stems from the fact that the product review dataset was generated by Mechanical Turkers, who combined authentic data with fake generated reviews. We suspect that the model may struggle to identify reviews authored by Mechanical Turkers as deceptive, as these reviews often appear genuinely written. For instance, statements like "*Ive been an xbox fan for a long time, and I love new tech. I love this console and I hope Microsoft keeps on banging out great hardware for decades. WOO*!", labeled as "deceptive" in the dataset due to their synthetic origin from MTurkers, despite their authentic tone. Another factor contributing to the complexity of this dataset is exemplified by a fake product review where the review was deemed fake because it discussed a TV product when the actual item was not a TV. Without metadata indicating the type of product or its manufacturer, it becomes challenging to discern the authenticity of a review solely based on its content.

SMS Spams As shown in Table 2 and Table 3, SMS domain exhibited the highest performance meaning the easiest to detect in terms of deception. We investigate further and find that shorter SPAM messages are easy to spot compared to genuine SMS due to their generic content filled with unsolicited offers and promotions. In contrast, non-deceptive SMS messages reflect a personal relationship or conversation with the recipient.

7. Conclusion

This study significantly advances the field of deception detection by introducing a benchmark data set, introducing a novel PEFT adapter, facilitating comparative analysis of deception detection algorithms, and exploring universal linguistic deception indicators. In this work, we thoroughly assess domain adaptation and classification methods that utilize transformer-based LLMs for deception detection. The proposed adapter surpasses similar approaches and adaptation strategies, increasing the significance of our contributions. Furthermore, the evaluation showed that our design is not only model-agnostic but also task-adaptive and, to some extent, task-agnostic, extending its applicability and value across the field. Most importantly, our findings contradict research concluding that a model's reasoning ability generally is of far greater significance than the quality and richness of representations (Bhatia et al., 2023), leading us to hypothesize that the answer to this question may be quite complex and depend on many factors.

Our future endeavors include assessing *PreXIA*'s efficacy on generic benchmarks, such as SuperGLUE (Wang et al., 2019), using DIFrauD to study linguistic cues for deception across domains and as a standard benchmark for deception detection in text data. We are also interested in gaining further insight into the relative contribution of representation quality and reasoning ability to the model prediction and understanding what role, if any, various factors such as data distribution and domain have to play in this equation. Finally, we will provide other researchers and the general public with any necessary updates and support for the adapter and the dataset introduced in this paper upon request.

8. Ethical Considerations

We perform due diligence to remove private and financial information from the data, including links, emails, social security numbers, bank account numbers, full names, and other uniquely identifiable information. Although this process does not guarantee anonymity or the preservation of confidential information, it removes and obscures much of it. We also do our best to ensure the lack of bias and neutrality of the trained and published models. To the best of our knowledge, this work follows all applicable patent laws, respects the copyrights of any resources used, and does not violate any prior licensing present in tools and resources used in its creation.

9. Data and Code Availability

Language resources and contributions to the corpora are available on the ReDAS (*Reasoning and Data Analytics for Security*) Laboratory Huggingface account `ReDASers` [5].

10. Acknowledgements

Research partly supported by NSF grants 2210198 and 2244279, and ARO grants W911NF-20-1-0254 and W911NF-23-1-0191. Verma is the founder of Everest Cyber Security and Analytics, Inc.

11. Bibliographical References

Mohamed Abouelenien, Rada Mihalcea, and Mihai Burzo. 2015. Trimodal analysis of deceptive behavior. In *Proceedings of the 2015 ACM on Workshop on Multimodal Deception Detection*, WMDD '15, page 9–13, New York, NY, USA. Association for Computing Machinery.

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. 2022. What learning algorithm is in-context learning? investigations with linear models. In *The Eleventh International Conference on Learning Representations*.

Tiago A. Almeida, Jose Maria Gomez Hidalgo, and Akebo Yamakami. 2011. Contributions to the study of sms spam filtering: New collection and results. In *Proceedings of the 2011 ACM Symposium on Document Engineering (DOCENG'11)*.

Kristjan Arumae, Qing Sun, and Parminder Bhatia. 2020. An empirical investigation towards efficient multi-domain language model pre-training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4854–4864, Online. Association for Computational Linguistics.

Srikanth Bethu, B. Sankara Babu, K. Madhavi, and P. Gopala Krishna. 2019. Algorithm selection and model evaluation in application design using machine learning. In *Machine Learning for Networking - Second IFIP TC 6 International Conference, MLN 2019, Paris, France, December 3-5, 2019, Revised Selected Papers*, volume 12081 of *Lecture Notes in Computer Science*, pages 175–195. Springer.

Kush Bhatia, Avanika Narayan, Christopher De Sa, and Christopher Ré. 2023. TART: A plug-and-play transformer module for task-agnostic reasoning. *CoRR*, abs/2306.07536.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR.

Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Pieler, USVSN Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang, and Samuel Weinbach. 2022. Gpt-neox-20b: An open-source autoregressive language model. *CoRR*, abs/2204.06745.

Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. 1992. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152.

Wilson Ceron, Mathias-Felipe de Lima-Santos, and Marcos Gonçalves Quiles. 2020. Fake news agenda in the era of covid-19: Identifying trends through fact-checking content. *Online Social Networks Media*, 21:100–116.

Yukang Chen, Shengju Qian, Haotian Tang, Xin Lai, Zhijian Liu, Song Han, and Jiaya Jia. 2023. Longlora: Efficient fine-tuning of long-context large language models. *CoRR*, abs/2309.12307.

Alexis Conneau and Douwe Kiela. 2018. SentEval: An evaluation toolkit for universal sentence representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

[5] `https://huggingface.co/redasers`

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, et al. 2023. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence*, 5(3):220–235.

Yong Fang, Cheng Zhang, Cheng Huang, Liang Liu, and Yue Yang. 2019. Phishing email detection using improved rcnn model with multilevel vectors and attention mechanism. *IEEE Access*, 7:56329–56340.

Song Feng, Ritwik Banerjee, and Yejin Choi. 2012. Syntactic stylometry for deception detection. In *The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, July 8-14, 2012, Jeju Island, Korea - Volume 2: Short Papers*, pages 171–175. The Association for Computer Linguistics.

Tommi Gröndahl and N. Asokan. 2019. Text analysis in adversarial settings: Does deception leave a stylistic trace? *ACM Comput. Surv.*, 52(3).

Abdullah Hamid, Nasrullah Sheikh, Naina Said, Kashif Ahmad, Asma Gul, Laiq Hasan, and Ala I. Al-Fuqaha. 2020. Fake news detection in social media using graph neural networks and NLP techniques: A COVID-19 use-case. In *Working Notes Proceedings of the MediaEval 2020 Workshop, Online, 14-15 December 2020*, volume 2882 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2022. Towards a unified view of parameter-efficient transfer learning. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*, pages 507–517.

Ángel Hernández-Castañeda, Hiram Calvo, Alexander Gelbukh, and Jorge Flores. 2017. Cross-domain deception detection using support vector networks. *Soft Computing*, 21:585–595.

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.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *ICLR 2022*.

Otto Jespersen. 1922. *Language: Its Nature, Development, and Origin*. Allen and Unwin.

Nitin Jindal and Bing Liu. 2008. Opinion spam and analysis. In *Proceedings of the International Conference on Web Search and Data Mining, WSDM 2008, Palo Alto, California, USA, February 11-12, 2008*, pages 219–230. ACM.

Elena Kochkina, Maria Liakata, and Arkaitz Zubiaga. 2018. PHEME dataset for Rumour Detection and Veracity Classification.

Johnson Kuan and Jonas Mueller. 2022. Back to the basics: Revisiting out-of-distribution detection baselines. *CoRR*, abs/2207.03061.

Huayi Li, Zhiyuan Chen, Arjun Mukherjee, Bing Liu, and Jidong Shao. 2015. Analyzing and detecting opinion spam on a large-scale dataset via temporal and spatial patterns. In *International Conference on Web and Social Media*, volume 9. AAAI.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. 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)*, pages 4582–4597.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin A Raffel. 2022. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems*, 35:1950–1965.

Yuning Mao, Lambert Mathias, Rui Hou, Amjad Almahairi, Hao Ma, Jiawei Han, Wen-tau Yih, and Madian Khabsa. 2022. Unipelt: A unified framework for parameter-efficient language model tuning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*.

Michael McCloskey and Neal J. Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In Gordon H. Bower, editor, *Psychology of Learning and Motivation*, volume 24, pages 109–165. Academic Press.

Sandhya Mishra and Devpriya Soni. 2023. Sms phishing dataset for machine learning and pattern recognition. In *Proceedings of the 14th International Conference on Soft Computing and Pattern Recognition (SoCPaR 2022)*, pages 597–604, Cham. Springer Nature Switzerland.

Arjun Mukherjee, Vivek Venkataraman, Bing Liu, and Natalie S. Glance. 2013. What yelp fake review filter might be doing? In *Proceedings of the Seventh International Conference on Weblogs and Social Media, ICWSM 2013, Cambridge, Massachusetts, USA, July 8-11, 2013*. The AAAI Press.

Myle Ott, Yejin Choi, Claire Cardie, and Jeffrey T. Hancock. 2011. Finding deceptive opinion spam by any stretch of the imagination. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, pages 309–319. The Association for Computer Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Verónica Pérez-Rosas, Mohamed Abouelenien, Rada Mihalcea, Yao Xiao, CJ Linton, and Mihai Burzo. 2015. Verbal and nonverbal clues for real-life deception detection. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2336–2346, Lisbon, Portugal. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Matthew E. Peters, Sebastian Ruder, and Noah A. Smith. 2019. To tune or not to tune? adapting pretrained representations to diverse tasks. In *Proceedings of the 4th Workshop on Representation Learning for NLP, RepL4NLP@ACL 2019, Florence, Italy, August 2, 2019*, pages 7–14. Association for Computational Linguistics.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2021. AdapterFusion: Non-destructive task composition for transfer learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 487–503, Online. Association for Computational Linguistics.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020a. Adapterhub: A framework for adapting transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020): Systems Demonstrations*, pages 46–54, Online. Association for Computational Linguistics.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020b. Mad-x: An adapter-based framework for multi-task cross-lingual transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7654–7673.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/research-covers/languageunsupervised/language understanding paper. pdf*.

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

Roger Ratcliff. 1990. Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. *Psychological review*, 97(2):285.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *Proceedings of the 2019 Conference* *on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 3973–3983.

Yafeng Ren and Donghong Ji. 2017. Neural networks for deceptive opinion spam detection: An empirical study. *Information Sciences*, 385-386:213–224.

Rodrigo Rill-García, Luis Villasenor-Pineda, Verónica Reyes-Meza, and Hugo Jair Escalante. 2018. From text to speech: A multimodal cross-domain approach for deception detection. In *Pattern Recognition and Information Forensics: ICPR 2018 International Workshops, CVAUI, IWCF, and MIPPSNA, Beijing, China, August 20-24, 2018, Revised Selected Papers 24*, pages 164–177. Springer.

Mark B. Ring. 1997. Child: A first step towards continual learning. *Machine Learning*, 28(1):77–104.

Irina Rish et al. 2001. An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46.

Andreas Rücklé, Gregor Geigle, Max Glockner, Tilman Beck, Jonas Pfeiffer, Nils Reimers, and Iryna Gurevych. 2021. AdapterDrop: On the efficiency of adapters in transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7930–7946, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sebastian Ruder and Barbara Plank. 2017. Learning to select data for transfer learning with bayesian optimization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 372–382. Association for Computational Linguistics.

Javier Sánchez-Junquera, Luis Villasenor-Pineda, Manuel Montes-y Gómez, Paolo Rosso, and Efstathios Stamatatos. 2020. Masking domain-specific information for cross-domain deception detection. *Pattern Recognition Letters*, 135:122–130.

Sadat Shahriar, Arjun Mukherjee, and Omprakash Gnawali. 2022a. Deception detection with feature-augmentation by soft domain transfer. In *International Conference on Social Informatics*, pages 373–380. Springer.

Sadat Shahriar, Arjun Mukherjee, and Omprakash Gnawali. 2022b. Improving phishing detection via psychological trait scoring. *CoRR*, abs/2208.06792.

Sharad Singhal and Lance Wu. 1988. Training multilayer perceptrons with the extende kalman algorithm. In *Advances in Neural Information Processing Systems 1, [NIPS Conference, Denver, Colorado, USA, 1988]*, pages 133–140. Morgan Kaufmann.

Yi Tay, Mostafa Dehghani, Jinfeng Rao, William Fedus, Samira Abnar, Hyung Won Chung, Sharan Narang, Dani Yogatama, Ashish Vaswani, and Donald Metzler. 2021. Scale efficiently: Insights from pretraining and finetuning transformers. In *International Conference on Learning Representations*.

Bibek Upadhayay and Vahid Behzadan. 2020. Sentimental liar: Extended corpus and deep learning models for fake claim classification. In *2020 IEEE International Conference on Intelligence and Security Informatics (ISI)*, pages 1–6. IEEE.

Pawan Kumar Verma, Prateek Agrawal, Ivone Amorim, and Radu Prodan. 2021. Welfake: Word embedding over linguistic features for fake news detection. *IEEE Transactions on Computational Social Systems*, 8(4):881–893.

Sokratis Vidros, Constantinos Kolias, Georgios Kambourakis, and Leman Akoglu. 2017. Automatic detection of online recruitment frauds: Characteristics, methods, and a public dataset. *Future Internet*, 9(1).

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32.

William Yang Wang. 2017. "liar, liar pants on fire": A new benchmark dataset for fake news detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 2: Short Papers*, pages 422–426. Association for Computational Linguistics.

Dustin Wright and Isabelle Augenstein. 2020. Transformer based multi-source domain adaptation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7963–7974, Online. Association for Computational Linguistics.

Yunzhi Yao, Shaohan Huang, Wenhui Wang, Li Dong, and Furu Wei. 2021. Adapt-and-distill:

Developing small, fast and effective pretrained language models for domains. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 460–470, Online. Association for Computational Linguistics.

Yue Yu, Chenyan Xiong, Si Sun, Chao Zhang, and Arnold Overwijk. 2022. Coco-dr: Combating distribution shifts in zero-shot dense retrieval with contrastive and distributionally robust learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1462–1479.

Victor Zeng, Shahryar Baki, Ayman El Aassal, Rakesh Verma, Luis Felipe Teixeira De Moraes, and Avisha Das. 2020. Diverse datasets and a customizable benchmarking framework for phishing. In *Proceedings of the Sixth International Workshop on Security and Privacy Analytics*, IWSPA '20, page 35–41, New York, NY, USA. Association for Computing Machinery.

Victor Zeng, Xuting Liu, and Rakesh M. Verma. 2022. Does deception leave a content independent stylistic trace? In *Proceedings of the Twelfth ACM Conference on Data and Application Security and Privacy*, CODASPY '22, page 349–351, New York, NY, USA. Association for Computing Machinery.

Rong Zhang, Revanth Gangi Reddy, Md Arafat Sultan, Vittorio Castelli, Anthony Ferritto, Radu Florian, Efsun Sarioglu Kayi, Salim Roukos, Avi Sil, and Todd Ward. 2020. Multi-stage pre-training for low-resource domain adaptation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5461–5468, Online. Association for Computational Linguistics.

Wen Zhang, Yuhang Du, Taketoshi Yoshida, and Qing Wang. 2018. DRI-RCNN: an approach to deceptive review identification using recurrent convolutional neural network. *Inf. Process. Manag.*, 54(4):576–592.

A. Supplementary Material

A.1. DIFrauD Data Diversity

In Figure 4, we notice a strong resemblance between phishing and fake news domains, as they both employ deceptive tactics through fake offers as a means to manipulate and sway users' into taking immediate action.

On the other hand, the domains associated with job scams exhibit the lowest similarity with SMS spam domains. In the case of SMS spam, there's a tendency for shorter, deceptive messages that often prioritize rewards and payments over job scams.

Considering that DIFrauD comprises solely deceptive domains, we observe a closely correlated heatmap with a consistent color palette. This indicates a degree of similarity among the domains, albeit not excessively so, thereby DIFrauD combines a diverse array of deceptive domains into one cohesive dataset.

A.2. DIFrauD Dataset Distribution

DIFrauD Deception Dataset		
Domain	*# Deceptive*	*# Non-Deceptive*
Pols	$8,042$	$4,455$
News	$8,832$	$11,624$
Jobs	599	$13,696$
Prod	$10,492$	$10,479$
Phish	$6,074$	$9,198$
Twitter	$1,969$	$3,820$
SMS	$1,274$	$5,300$

Table 4: Class Distribution of the DIFrauD dataset

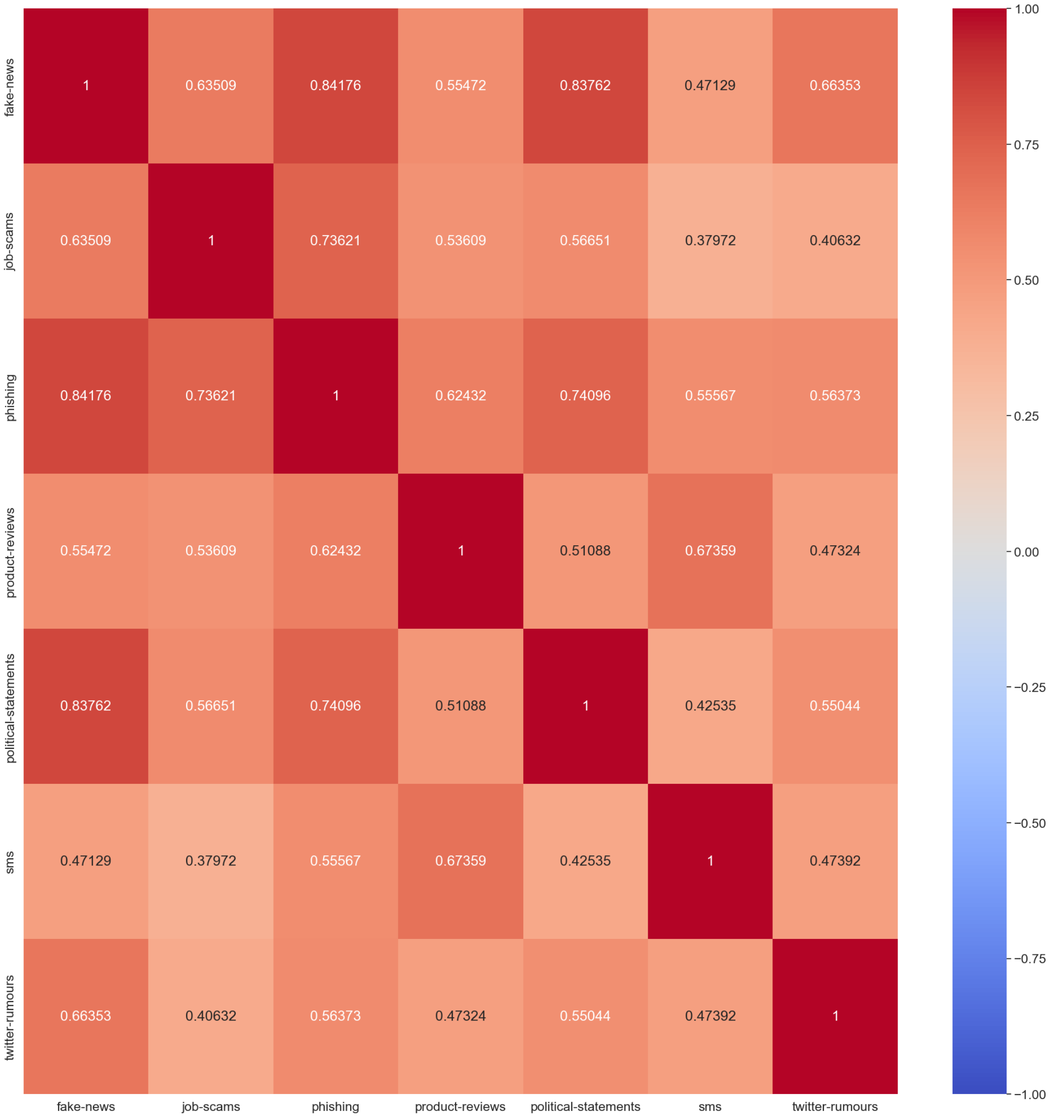

Figure 4: Diversity of DIFrauD Dataset Domains

Dataset Split Configuration						
Domain	**Training**		**Validation**		**Testing**	
	# Deceptive	*# Non-Deceptive*	# Deceptive	# Non-Deceptive	# Deceptive	# Non-Deceptive
Pols	$6,433$	$3,564$	804	446	805	445
News	$7,065$	$9,299$	884	$1,162$	883	$1,163$
Jobs	479	$10,957$	60	$1,369$	60	$1,370$
Prod	$8,393$	$8,383$	$1,049$	$1,048$	$1,050$	$1,048$
Phish	$4,859$	$7,358$	607	920	608	920
Twitter	$1,575$	$3,056$	197	382	197	382
SMS	$1,019$	$4,240$	127	530	128	530

Table 5: DIFrauD Dataset Splitting