ET-BERT with Adapter Fusion: Time-Efficient Continual Learning Framework for Encrypted Traffic Classification*

Minsu Kim[1], Daeho Choi[1], Younghyo Cho[1], Yeog Kim[2], Jungsuk Song[3], Changhoon Lee[1], and Kiwook Sohn[1]†

[1] Seoul National University of Science and Technology, Seoul, Republic of Korea
[2] Research Center of Electrical and Information Technology, Seoul, Republic of Korea
[3] Korea Institute of Science and Technology Information(General Director)
lamoda13@seoultech.ac.kr

Abstract

Encrypted traffic classification is essential in modern network defense but remains a challenging task due to the opacity of payloads and the continuously evolving attack surface. Recently, ET-BERT [1], a Transformer-based model, achieved state-of-the-art performance by treating traffic sequences as contextualized tokens. However, the approach of fine-tuning the entire model for each new class is prohibitively costly and entails the risk of catastrophic forgetting. In this study, we introduce an adapter-based parameter-efficient continual learning framework for encrypted traffic classification. The framework consists of (i) the baseline full fine-tuning (FT), (ii) adapter tuning for base classes (0–9), (iii) incremental adapter training with only a small number of parameters for the novel class (10), and (iv) Adapter Fusion, which non-destructively combines base and incremental knowledge. Since this design freezes existing parameters and learns only new modules, it structurally suppresses the risk of catastrophic forgetting while enabling real-time updates without full retraining. On the USTC-TFC2016 dataset, the framework achieved an accuracy of 0.9970 with FT on all 11 classes, 0.9941 with base adapter tuning, and 0.9862 when applying Fusion. The incremental adapter immediately adapted to the new class, and the Fusion step reliably recovered overall classification performance. In terms of training time, while full FT required about 50 minutes, incremental adapter training took 3.5 minutes, and even with Fusion it was completed within about 13 minutes, demonstrating more than 4x faster training. This study empirically demonstrates that it is possible to simultaneously achieve performance retention and rapid deployment updates while minimizing the retraining burden.

Keywords: Encrypted Traffic Classification, Incremental Learning, Adapter Fusion, ET-BERT

1 Introduction

Today, a significant portion of network traffic is transmitted based on encrypted protocols, which have become core technologies for strengthening security and privacy. However, such encryption also acts as a factor that increases the difficulty of security threat detection. Traditional packet analysis techniques or signature-based detection methods face limitations in effectively classifying normal and malicious traffic in encrypted environments, as they cannot

*Proceedings of the 9th International Conference on Mobile Internet Security (MobiSec'25), Article No. 67, December 16-18, 2025, Sapporo , Japan.

†Corresponding author

access meaningful payload information. Consequently, recent research has evolved toward developing machine learning and deep learning classifiers that leverage the metadata and sequence patterns of encrypted traffic itself [2].

In particular, the success of Transformer-based language models has opened new possibilities in the security domain. State-of-the-art encrypted traffic classification models such as ET-BERT process traffic sequences as natural language tokens, achieving high classification accuracy [1]. However, since existing models have statically trained structures, when new types of attack traffic emerge, the entire model must be retrained, leading to inefficiency. This approach entails high training costs and storage resource consumption, and poses the risk of catastrophic forgetting that degrades existing performance [3]. Especially in situations where new traffic types appear in real time, retraining the entire model is impractical in terms of training speed, making rapid response impossible.

Incremental learning has been gaining attention as a method to address this issue. Incremental learning is designed to adapt to new tasks or classes while retaining previously learned models [4]. However, when applied to the security domain, the key challenge is to minimize forgetting of prior knowledge while quickly and efficiently learning new attacks [5].

In this study, we propose an adapter-based incremental learning framework built on the ET-BERT model. The adapter technique inserts small modules into each Transformer layer and greatly improves parameter efficiency by training only the modules instead of the entire parameters [6]. Furthermore, through the Adapter Fusion technique, multiple adapters can be merged to integrate prior and new knowledge [7]. This approach provides the potential to alleviate both resource inefficiency and forgetting problems faced by encrypted traffic classification models, allowing adaptation to new traffic much faster and more efficiently.

The main contributions of this study are as follows.

1. Design of an ET-BERT-based incremental learning framework: We propose a structure that allows comparison among various training strategies, including baseline fine-tuning, adapter tuning, incremental experts, and adapter fusion.

2. Derivation of practical implications: We experimentally verify that when new attack traffic is discovered in real security environments, the proposed framework enables rapid response without full retraining of the model.

3. Mitigation of forgetting phenomenon: Based on the USTC-TFC dataset, we analyze that forgetting does not occur when adding new classes and experimentally verify that Adapter Fusion can alleviate it.

2 Related Work

Encrypted traffic classification and incremental learning have each been actively studied in their own right, but attempts to integrate the two research streams remain at an early stage. Prior work has primarily focused on designing static models to achieve high accuracy, relying on retraining the entire model whenever new attacks or service types emerge [2]. In contrast, in the field of continual learning, various algorithmic and architectural techniques have been proposed across domains to mitigate catastrophic forgetting; however, there are few cases that directly apply these techniques to the special environment of security traffic (high-dimensionality, sequence-based features, etc.). [3] [4].

2.1 Encrypted Traffic Classification with Transformers

Encrypted traffic classification must compensate for the limitations of traditional DPI (signature) methods due to restricted payload visibility. Recently, approaches that regard traffic patterns as sequences and leverage large-scale pre-trained representations have gained traction, with ET-BERT being a representative example [1]. Lin et al. demonstrated that ET-BERT, pre-trained on large-scale unlabeled traffic, achieved substantial performance gains over existing methods on various downstream tasks (ISCX-Tor [8], ISCX-VPN-Service [9], etc.) (e.g., ISCX-Tor F1 99.2%). This study follows that line of work, but explores how to simultaneously achieve knowledge retention and rapid adaptation of ET-BERT in an incremental learning setting.

Meanwhile, publicly available datasets such as USTC-TFC2016 [10] are widely used as benchmarks for encrypted (or mixed) traffic classification and continue to serve as baselines for subsequent studies. For example, BSTFNet proposed in 2024 reported accuracy and F1 of about 99.4% on USTC-TFC2016, improving the separability of malicious traffic by integrating global semantic features with spatiotemporal features [11].

2.2 Catastrophic Forgetting & Class-Incremental Learning

In incremental/continual learning, when additional training is performed using only new-class data, the central challenge is the problem of catastrophic forgetting, where past classes are rapidly forgotten. A 2023 comprehensive survey summarized CIL methods and emphasized the need for aligning memory budgets and memory-agnostic metrics for fair comparison [4]. This perspective is especially important in situations where rehearsal is difficult due to sensitive data, as in the security domain. Moreover, a 2024 study pointed out that the use of rehearsal memory can entail security/privacy risks and raised the need for methods that reduce forgetting without data reuse [5]. The reason this study examines Adapter Fusion also stems from a practical requirement to combine knowledge at the module level without re-storing the original data.

In security domains, the long-term storage or reuse of sensitive traffic data can be strictly limited by privacy regulations (e.g., data minimization under GDPR). Motivated by such constraints, this study focuses on rehearsal-free, parameter-efficient approaches (adapters and fusion) that do not require re-storing past data, and excludes rehearsal-based CL methods from direct comparison.

2.3 Parameter-Efficient Transfer Learning (Adapters)

When transferring large-scale pre-trained models to downstream tasks, fine-tuning all parameters is inefficient because a full replica is required for each task. Houlsby et al. (2019) inserted small bottleneck modules (adapters) between transformer layers and showed that learning only a very small number of parameters per task can achieve performance comparable to full fine-tuning [6]. Subsequently (2020), in the context of LLMs, families of adapters (series/parallel, prompt-based, re-parameterized, etc.) were systematized, and reflections comparing and organizing PEFT as a whole were presented [12]. These advances offer a practical alternative for environments like security traffic classification, where tasks/classes are frequently added, by reducing storage and deployment burdens.

Furthermore, He et al. (2022) proposed LoRA (Low-Rank Adaptation), which implements efficient parameter learning using low-rank matrices instead of adapters [13]. LoRA achieved dramatic reductions in training memory and parameter costs even for very large models such as GPT-3, while minimizing performance degradation. Recently, LoRA has also been applied in the security text and log analysis domain, where it has significantly improved storage efficiency and reusability compared to conventional fine-tuning.

2.4 Knowledge Composition via Adapter Fusion

Adapter Fusion was proposed as an approach to non-destructively combine adapters trained on multiple tasks. Pfeiffer et al. empirically demonstrated over 16 NLU tasks that by decoupling knowledge extraction (training adapters per task) from knowledge composition (training fusion layers), one can mitigate the forgetting and balance issues of sequential fine-tuning/multi-task learning [7]. This study extends that idea to class-incremental security classification by integrating a base-task adapter and a new-class adapter via Fusion, thereby exploring a lightweight solution that preserves existing knowledge while rapidly injecting new knowledge.

In addition, Karimi Mahabadi et al. (2021) proposed Compacter, a new adapter architecture that leverages parameter sharing and high-order tensor factorization techniques to remain even more compact than Adapter Fusion while maintaining high performance [14]. Compacter experimentally showed that, when combined with adapter fusion techniques in multi-task environments, it can maximize storage efficiency as well as mitigate forgetting. This suggests high applicability even in environments such as security traffic, where new tasks continue to be added [15].

In this study, we combine the research axes of encrypted traffic classification and incremental learning and focus on applying incremental learning techniques to Transformer-based encrypted traffic classification models—particularly parameter-efficient transfer learning based on adapters and knowledge integration through Adapter Fusion.

3 Proposed Method

In this study, to address the problem of failing to respond to new traffic types during encrypted traffic classification, we apply parameter-efficient transfer learning (PEFT) based on a Transformer pre-trained model (ET-BERT) in an incremental learning environment and mitigate catastrophic forgetting through the Adapter Fusion technique.

3.1 Problem Definition

The encrypted traffic classification problem is defined as a multi-class classification problem that maps input data x to one of the pre-defined class sets C. In this study, the input data are encrypted traffic at the packet sequence and flow level, so the data are represented as integer sequences in bytes. Each sample x is normalized into a sequence of length L ($x = (x_1, x_2, \ldots, x_L)$), where insufficient length is padded with tokens 0. This input value is fed into the Transformer encoder, and the model outputs a class probability distribution $\hat{y} \in \mathbb{R}^C (\hat{y} = f_\Theta(x), \sum_{c=1}^{C} \hat{y}_c = 1)$. In the incremental learning environment, the class set expands over time step t (Equation 1).

$$C_t = C_{t-1} \cup C_{\text{new}}. \tag{1}$$

That is, initially only C_{base} classes exist, and whenever a new traffic type occurs, C_{new} is added. Therefore, the goal of this study is to enable adaptation to these new classes while maintaining classification performance on existing classes.

3.2 Baseline: Full Fine-Tuning with Transformers

The starting point of this study is the ET-BERT [1] model based on a Transformer encoder architecture. Input sequences are converted into high-dimensional representations through an embedding layer, and then into contextualized hidden vectors through multi-layer self-attention blocks. During this process, padding tokens are ignored via an attention mask, and only actual data positions are included in meaningful computation.

Finally, the extracted [CLS] token is regarded as the vector representing the entire sequence, which is then used to construct the classifier. The classifier consists of a linear transformation and a softmax layer to predict the input sequence as one of the C classes. During training, cross-entropy loss is generally used, which is the most widely used metric for multi-class classification problems (Equation 2).

$$\mathcal{L}_{\mathrm{CE}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c}\log \hat{y}_{i,c}. \tag{2}$$

where N is the batch size, C is the number of classes, and $y_{i,c}$ is the one-hot label. This approach has the advantage of high classification performance but suffers from the problem that all parameters must be re-trained whenever new classes are added. In other words, as the number of tasks increases, model storage cost and training time increase linearly, which is a fundamental limitation.

3.3 Parameter-Efficient Transfer Learning with Adapters

Instead of training all parameters, in this study we secure parameter efficiency by inserting adapter modules into each Transformer block. An adapter has a bottleneck structure that temporarily projects the input vector into a low-dimensional space and then expands it back to the original dimension. For a hidden vector $h \in \mathbb{R}^d$, the adapter transformation is defined as follows (Equation 3).

$$h' = h + W_{\mathrm{up}}\ \sigma(W_{\mathrm{down}}\, h) \tag{3}$$

where $W_{\mathrm{down}} \in \mathbb{R}^{r\times d}, W_{\mathrm{up}} \in \mathbb{R}^{d\times r},\ r \ll d$, and σ is a nonlinear activation function.

This structure provides three advantages. First, since only 1–5% of the entire model parameters are trained, training costs and storage requirements are greatly reduced. Second, only independent adapters per task need to be stored, making model management easier when new tasks are added. Third, the existing Transformer parameters are frozen, making it possible to preserve existing performance. Therefore, adapters can serve as particularly useful modules in incremental learning environments.

3.4 Incremental Learning with New Class Adapters

When new classes emerge in the incremental learning environment, the existing model weights are left unchanged, and only new adapters are trained. That is, the pre-trained ET-BERT model and the base adapter are frozen, and an incremental adapter specialized for new classes is added. This incremental adapter is trained solely on new class samples, without reusing existing class data.

This approach does not use rehearsal techniques, so it fundamentally avoids the security and privacy issues of re-storing data [5]. It also has the advantage of quickly adapting to new classes with only a small number of training epochs.

However, such an independent training process causes a rapid drop in performance on existing classes [3] [4]. In other words, while new classes are well classified, performance on existing classes is sacrificed. Therefore, additional knowledge integration methods are necessary.

3.5 Knowledge Integration via Adapter Fusion

To compensate for the limitations of the incremental adapter method, this study introduces Adapter Fusion [7] [14]. Adapter Fusion is a method of dynamically combining the representations output by multiple adapters through weighted summation to generate the final representation.

If there are K adapters, the fusion representation is defined as follows (Equation 4).

$$h^{\text{fusion}} = \sum_{k=1}^{K} \alpha_k h^{(k)}, \qquad \sum_{k=1}^{K} \alpha_k = 1. \tag{4}$$

where $h^(k))$ is the output of the k-th adapter, and α_k is a learnable weight.

The core of this approach is that by simultaneously activating the base adapter (which maintains existing class performance) and the incremental adapter (which includes new class knowledge), both types of knowledge can be balanced. The weights α_k are optimized during training, and a regularization term is added so that they are not skewed toward specific tasks. As a result, Fusion alleviates catastrophic forgetting while securing adaptability to new classes.

3.6 Optimization Strategy

The training process is stably carried out using the AdamW optimizer and cosine learning rate scheduling. The loss function is basically based on cross-entropy loss, but in adapter training, a regularization term is added to suppress unnecessary parameter explosion. The overall loss is defined as follows (Equation 5).

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{CE} + \lambda \|\theta_{\text{adapter}}\|_2^2 \tag{5}$$

where θ_{adapter} is the parameter set of the adapter, and λ is the regularization strength.

In the Adapter Fusion stage, additional constraints are imposed to prevent the weights from being biased toward specific adapters. In other words, a regularization term for α_k is included to maintain balance in the weight distribution.

4 Experiments

In this section, we experimentally verify the classification performance of the proposed adapter-based incremental learning framework. The objectives of the experiments are threefold. First, to confirm the efficiency of the proposed method compared to existing full fine-tuning and adapter tuning approaches. Second, to analyze how the incremental adapter method and the adapter fusion method affect catastrophic forgetting in incremental learning scenarios where new classes appear. Third, to discuss the applicability of the proposed framework to real-world environments based on the experimental results.

4.1 Experimental Setup

In this study, we used the publicly available USTC-TFC2016 [10] dataset for experiments. The dataset consists of 10 normal traffic classes (e.g., Web, Email, Chat, etc.) and 10 malicious traffic classes (e.g., Zeus, Virut, Neris, etc.), each containing thousands of network flows. Each flow is composed of multiple packets, and in this study, payload byte sequences were normalized to a fixed length of 512 for input into the Transformer model.

The dataset was split as follows (Table 1). Classes 0–9 are the base classes used in the initial training stage, and class 10 is the class added during the incremental learning stage. For each class, the data was split into an 80:10:10 ratio for training, validation, and testing. The split (Base 0–9, Incremental 10) serves as a proxy scenario for sudden zero-day or variant outbreaks in a security operations center, rather than a mere random partition. Using the standard USTC-TFC2016 benchmark in this way allows us to assess the relative responsiveness to newly emerging classes without additional data collection, while acknowledging that multi-dataset and multi-protocol evaluations are left for future work.

Class ID	Class Type	Train (80%)	Valid (10%)	Test (10%)
0–9	Base	4000 samples	500 samples	500 samples
10	Incremental	4000 samples	500 samples	500 samples

Table 1: Dataset split per class (Train/Valid/Test).

The experiments were carried out in an identical hardware and software environment (Table 2). For reproducibility, the initial parameters used in the model training were fixed to constant values.

Category	Spec / Version
GPU	NVIDIA RTX 3090 (24GB) * 1
CPU	Xeon Gold 6230R (2.1GHz 26Core) * 2
PyTorch	2.4.1 + cu124
Reproducibility	Seed = 7

Table 2: Hardware and software environment

For performance evaluation, two metrics were used.

- **Accuracy**: Ratio of correctly classified samples among all samples.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

- **Macro-F1**: Arithmetic mean of F1 Scores per class, used to assess robustness against class imbalance.

$$\text{Macro-F1} = \frac{1}{C}\sum_{c=1}^{C}\frac{2P_cR_c}{P_c + R_c}, \quad P_c = \frac{TP}{TP + FP}, \quad R_c = \frac{TP}{TP + FN}$$

Experiment	Data	Epoch	Training Time	Test Acc	Macro-F1
Baseline FT	class 0–10	3	2996 s	0.9970	0.9970
Base Adapter	class 0–9	3	934 s	0.9941	0.9941
Incremental Expert	class 10	3	212 s	1.000	-
Incremental Expert (mixed test)	-	-	-	0.0909	0.0151
Fusion	class 0–10	3	579 s	0.9862	0.9862

Table 3: Results and analysis on USTC-TFC2016

4.2 Results and Analysis

- **Baseline model vs. Base Adapter Tuning**: Full fine-tuning achieved optimal performance because all parameters were updated, but it incurred high parameter costs. In contrast, Base Adapter Tuning updated only about 1.3% of the parameters, but showed a performance difference less than 1%.

- **Incremental Adapter**: For the new class, it achieved extremely high accuracy, demonstrating excellent adaptability to new classes. However, since it was trained as a single-class expert, knowledge of existing classes was not retained. On the mixed test, the average accuracy dropped to 0.0909 ($\approx 1/11$), i.e., random-guess level. This quantitatively indicates catastrophic forgetting when a new-class expert is used in isolation without access to prior knowledge, which is why the subsequent Fusion step is essential.

- **Adapter Fusion**: This method effectively balanced performance across existing and new classes. Adapter Fusion learns input-dependent weights α_k that dynamically compose base and incremental adapters. As a result, it recovers base-class performance to ≈99% while maintaining high detection for the new class (see Table 3), in stark contrast to the 9% obtained by the standalone expert.

4.3 Discussion

The experimental results provide the following important implications.

1. **Efficiency of Adapters**: Adapters can achieve nearly the same performance as full fine-tuning with far fewer parameters. This is a significant advantage in security systems where parameter storage and deployment costs are high.

2. **Limitations of Incremental Learning**: The incremental adapter method allows rapid adaptation to new attack classes but struggles to preserve existing knowledge. This indicates that simple incremental learning alone is insufficient to solve the problem in real security environments.

3. **Effectiveness of Adapter Fusion**: Adapter Fusion demonstrated high classification performance even for new classes while substantially alleviating catastrophic forgetting.

4. **Practical Applicability**: The proposed framework enables rapid response to new threats in real security environments without retraining the entire model, showing promise for real-world deployment.

5 Conclusion

In this study, we proposed a novel framework that combines adapter-based parameter-efficient transfer learning and the Adapter Fusion technique to address the incremental learning problem arising in encrypted network traffic environments. Although Transformer-based classification models have demonstrated high performance, they face inefficiency and catastrophic forgetting issues because the entire model must be retrained whenever new attack classes appear. To overcome these limitations, this study applied adapters to reduce the scale of trainable parameters, incrementally added adapters specialized for new classes, and finally integrated existing and new knowledge through Fusion.

This study proposed an adapter-based continual learning framework for ET-BERT-based encrypted traffic classification. By freezing the core/base knowledge, rapidly training only small-parameter adapters for new classes, and ultimately integrating them non-destructively with Adapter Fusion, the framework preserved most of the performance without full retraining. Particularly in operational environments requiring real-time updates, this approach provides a practical alternative that significantly reduces retraining costs and deployment burdens while maintaining high accuracy.

However, this study also has limitations. The experiments were confined to a single dataset (USTC-TFC2016) and did not sufficiently reflect diverse network conditions in real operational environments (e.g., real-time traffic, multi-protocol, large-scale class expansion). In addition, although weight optimization in the Fusion process was effective, when the number of adapters increases under large-scale incremental learning, computational cost may again increase, which remains a potential issue.

Future work will extend to scenarios involving continuous emergence of multiple new classes, investigate efficiency when the number of additional adapters grows, and explore online adaptation/scheduling strategies, thereby suggesting directions for achieving sustainable high-speed updates even in large-scale real-world environments.

6 Acknowledgments

- This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (RS-2023-00235509, Development of security monitoring technology based network behavior against encrypted cyber threats in ICT convergence environment).

References

[1] Y. Lin, J. Li, J. Wu, Y. Sun, and Z. Wu. Et-bert: A contextualized datagram representation with pre-training transformers for encrypted traffic classification. In *Proceedings of ACM CCS*, pages 1975–1989, 2022.

[2] W. Wang, M. Zhu, X. Zeng, X. Ye, and Y. Sheng. Malware traffic classification using convolutional neural network for representation learning. In *Proc. IEEE ICOIN*, pages 712–717, 2017.

[3] M. De Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3366–3385, 2021.

[4] M. Masana, X. Liu, B. Twardowski, M. Menta, A. D. Bagdanov, and J. Van De Weijer. Class-incremental learning: survey and performance evaluation on image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):5513–5533, 2023.

[5] T. Verma, L. Jin, J. Zhou, J. Huang, M. Tan, B. C. M. Choong, and Y. Liu. Privacy-preserving continual learning methods for medical image classification: a comparative analysis. *Frontiers in Medicine*, 10:1227515, 2023.

[6] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. de Laroussilhe, A. Gesmundo, S. Gelly, et al. Parameter-efficient transfer learning for nlp. In *Proceedings of ICML*, pages 2790–2799, 2019.

[7] J. Pfeiffer, A. Kamath, A. Rückle, K. Cho, and I. Gurevych. Adapterfusion: Non-destructive task composition for transfer learning. *arXiv preprint arXiv:2005.00247*, 2020.

[8] A. H. Lashkari, G. Draper-Gil, M. S. I. Mamun, and A. A. Ghorbani. Characterization of tor traffic using time-based features. In *Proc. ICISSP*, pages 253–262, 2017.

[9] G. Draper-Gil, A. H. Lashkari, M. S. I. Mamun, and A. A. Ghorbani. Characterization of encrypted and vpn traffic using time-related features. In *Proc. ICISSP*, pages 407–414, 2016.

[10] Ustc-tfc2016 dataset. https://www.scidb.cn/en/detail?dataSetId=58b9d3c2. University of Science and Technology of China.

[11] H. Huang, X. Zhang, Y. Lu, Z. Li, and S. Zhou. Bstfnet: An encrypted malicious traffic classification method integrating global semantic and spatiotemporal features. *Computers, Materials & Continua*, 78(3), 2024.

[12] J. Pfeiffer, I. Vulić, I. Gurevych, and S. Ruder. Mad-x: An adapter-based framework for multi-task cross-lingual transfer. *arXiv preprint arXiv:2005.00052*, 2020.

[13] E. Hu, Y. Shen, P. Wallis, et al. Lora: Low-rank adaptation of large language models. In *Proceedings of ICLR*, 2022.

[14] R. Karimi Mahabadi, J. Henderson, and S. Ruder. Compacter: Efficient low-rank hypercomplex adapter layers. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1022–1035, 2021.

[15] A. Rücklé, G. Geigle, M. Glockner, T. Beck, J. Pfeiffer, N. Reimers, and I. Gurevych. Adapterdrop: On the efficiency of adapters in transformers. *arXiv preprint arXiv:2010.11918*, 2020.