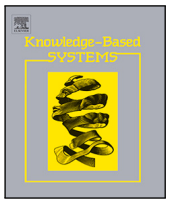
Knowledge-Based Systems 319 (2025) 113505

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys



Binary mask tuning on gradient: Towards multi-data question answering

Chuanyang Gong, Zhihua Wei *, Ping Zhu, Duoqian Miao

Department of Computer Science and Technology, Tongji University, Shanghai, China

ARTICLE INFO

Keywords:
Multi-data fine-tuning
Question answering
Binary mask tuning
Few-shot transfer learning

ABSTRACT

Pretrained language models have been widely applied in question-answering tasks. To achieve better generalization on unseen datasets, several previous studies often trained a single model on multiple datasets. However, owing to the noise and distribution differences among various datasets, the model tends to excessively adjust its weights during training on multiple datasets. This deviation from the initial pretraining state results in excellent performance on specific data but overfitting on other datasets. Ultimately, the model loses its ability to generalize to new data. In this paper, from the perspective of imposing constraints on model weights, we propose a novel fine-tuning method, binary mask tuning (BMT). We employ a carefully designed binary mask vector that is closely related to the data distribution to mask the gradient generated by backpropagation, achieving precise alignment of the subspace parameters required to fit the data from a huge parameter space. This approach aimed to enhance the adaptability of the model to the data distribution and improve parameter efficiency via more targeted fine-tuning. Our experiments demonstrate that BMT is not only effective in mitigating the tendency of the model to excessively adjust its weights but also in better capturing cross-dataset regularities and dataset-specific attributes in question-answering tasks across different datasets.

1. Introduction

In recent years, the introduction of various pretrained language models (PLMs) has brought a remarkable effect on the field of natural language processing (NLP). Many PLMs, such as BERT [1], GPT [2–4], RoBERT [5], XLNet [6], have become the powerful backbone for downstream tasks. Fine-tuning PLMs has become a new NLP paradigm and is increasingly used in machine translation, reading comprehension, and sentiment analysis. This study primarily focuses on PLM application in reading comprehension tasks. The goal of machine reading comprehension is to enable machines to read a given context and answer questions related to the content. This not only requires the model to integrate information across multiple sentences but also demands the ability to understand complex sentence structures and semantic relationships, thereby imposing extremely high requirements on its comprehension and reasoning capabilities. This task reflects the ability of artificial intelligence to acquire, understand, and extract information from text.

Many previous studies [7,8] have explored training a single network on multiple datasets with the expectation that the distribution of parameters learned by the network can be generalized better to new unseen datasets. However, despite these efforts, the model still exhibits insufficient generalization ability when confronted with new datasets. First, cross-dataset distribution shift significantly impacts the model's generalization performance. Due to potential substantial differences in feature distributions across datasets, the model is prone to overfitting to specific datasets during fine-tuning, making it difficult to learn universal patterns that can effectively transfer to new datasets. Therefore, the model fails to capture both cross-dataset regularities and dataset-specific properties well [9,10]. Second, excessive updates to model parameters can cause deviation from the pretrained weights, further undermining generalization capability. Full fine-tuning updates all parameters of the PLM, and repeated training on multiple datasets may lead to excessive deviation from the pretrained weights, resulting in the loss of general language representation abilities acquired during pretraining. Finally, catastrophic forgetting is particularly prominent in cross-dataset fine-tuning. When adapting to new datasets, the model may forget knowledge learned from the source datasets, thereby degrading its performance on new tasks. Previous studies [11,12] have demonstrated that applying constraints or regularization to the weight of the model can, to some extent, mitigate the deviation induced by fine-tuning. This approach enhances the model's generalization capabilities. For instance, Gouk et al. [13] achieved better generalization than traditional transfer learning by constraining class-specific weights in a small spherical space centred on the pretrained weights. Wu et al. [14] added different uniform noises to different parameter matrices of the pretrained weights before fine-tuning to improve the PLM effects on downstream tasks. Lee et al. [15] proposed mixout, which was inspired by the dropout concept and randomly replaced the weights of the model

* Corresponding author.
E-mail addresses: gongchuanyang@tongji.edu.cn (C. Gong), zhihua_wei@tongji.edu.cn (Z. Wei).

https://doi.org/10.1016/j.knosys.2025.113505
Received 8 December 2023; Received in revised form 24 February 2025; Accepted 4 April 2025
Available online 16 April 2025

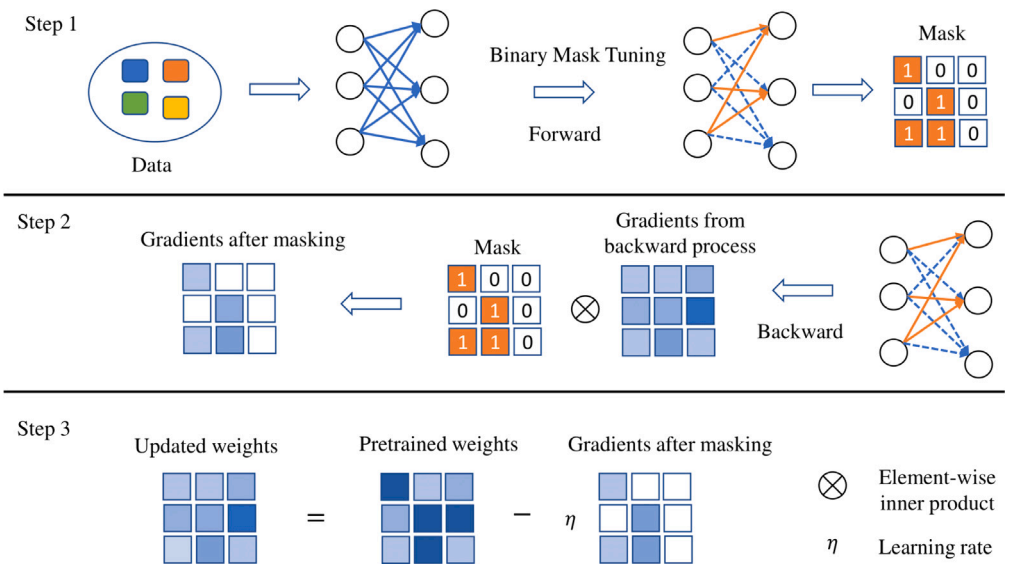


Fig. 1. Illustration of Binary Mask Tuning (BMT). Our main idea is to design a binary mask vector associated with the corresponding data distribution. We then use this well-designed binary mask vector to mask the gradients generated by backpropagation, achieving the update of model parameters".

during the fine-tuning phase. Mixout is a regularization technique designed to prevent catastrophic forgetting by modifying the fine-tuning process. Zhu et al. [16] proposed the lottery subnetwork, employing fewer critical parameters, which enhances the effectiveness in the target domain. Moreover, to adapt PLMs to various downstream tasks, Xu et al. [17] proposed a child-tuning method. This method generates a mask vector by sampling from a Bernoulli distribution during fine-tuning, which masks the gradients of non-subnetworks, resulting in the selective update of a small portion of the model parameters. This approach is a regularization technique that helps prevent overfitting on small datasets and improves generalization. Notably, the above-mentioned methods update only part of the parameters, implying that the parameters of the large-scale model are sparse. Therefore, it is not necessary to update all parameters during fine-tuning. It is not difficult to see that by restricting the degrees of freedom of model parameters or introducing uncertainty, a balance can be achieved between adapting to new tasks (fine-tuning) and retaining pretrained knowledge, thereby mitigating distributional deviation caused by task adaptation. Building on above insights, we attempt to introduce the concept of data distribution to establish connections between data distribution, gradients, and fine-tuning, thereby achieving control and constraints over model parameters and ultimately improving the model's generalization performance.

Specifically, our work focuses on leveraging data distribution as a crucial characteristic to optimize multi-dataset fine-tuning. To this end, we propose a novel fine-tuning technique, Binary Mask Tuning (BMT), to better exploit the potential of PLMs in reading comprehension tasks. BMT innovatively introduces parameters related to data distribution and transforms the process of solving these distribution parameters into an optimization problem, yielding a binary mask vector closely aligned with the data distribution. During the multi-dataset fine-tuning process, the binary mask vector is used to mask gradients generated by backpropagation, thereby guiding parameter updates in a more directional manner, rather than randomly, as shown in Fig. 1. Through the constraints imposed by the binary mask vector, BMT allows gradients to flow only to the most relevant subset of parameters, thereby reducing the risk of overfitting to any single dataset. At the same time, it ensures a more stable optimization path and alleviates excessive weight deviation caused by fine-tuning. Additionally, BMT utilizes L_0-norm regularization in the optimization objective, which encourages the sparsity of data-relevant parameters, improving model parameter efficiency and sharing. This ensures better knowledge transfer between datasets and further enhances the model's generalization ability.

We applied the BMT method to the extractive machine reading comprehension task and used the datasets from the MRQA 2019 shared task [7]. The proposed method is trained and evaluated on six in-domain datasets, and the generalization of transfer learning is studied on six out-of-domain datasets. Specifically, we conducted a series of experiments to explore the impact of BMT on multi-data fine-tuning models: BMT in multi-data fine-tuning, BMT in multi-data fine-tuning with an adapter [18,19], and BMT in few-shot transfer learning. We found that BMT can eliminate the interference of different datasets on the model to a certain extent and alleviate the preference of the model for certain datasets. Moreover, applying BMT in multi-data fine-tuning can achieve comparable results with the adapter; meanwhile, combining BMT and an adapter can further improve the generalization ability of the model on all out-of-domain datasets. Our experiments indicate that BMT can not only effectively alleviate fine-tuned models away from the pretrained weights but also better capture cross-dataset regularities and dataset-specific attributes of different datasets.

The main contributions of this study are as follows:

1. We introduce parameters associated with the data distribution and obtain a binary mask vector by solving an optimization problem. This binary mask vector helps the model filter out subspace parameters that better align with the distribution patterns of the dataset from the large parameter space.
2. We propose BMT, a simple and effective fine-tuning technique for multi-data question-answering tasks. It utilizes a carefully designed binary mask vector to mask the gradients generated by backpropagation, thereby guiding parameter updates in a more directional manner. This approach effectively mitigates the model's tendency to overfit during multi-data fine-tuning, enhancing its ability to capture cross-dataset regularities and dataset-specific attributes.
3. The BMT can also be combined with the adapter to improve further the model's performance on in-domain and out-of-domain datasets, zero-shot generalization, and few-shot transfer learning scenarios, providing an innovative solution for tackling challenges in reading comprehension tasks.

In summary, BMT leverages data-driven binary mask vectors to selectively update model parameters, guiding the fine-tuning process

in a more directed and precisely controlled manner. This not only helps the model maintain the effectiveness of its pretrained weights during multi-dataset fine-tuning (avoiding excessive deviation from pretrained weights when the model is trained on multiple datasets), but also enhances the model's balanced learning of both cross-dataset commonality and dataset-specific characteristics, thereby significantly improving the model's generalization capability.

This paper is organized as follows. Firstly, the related work is briefly introduced in Section 2, and then we will elaborate in detail on the design of the binary mask vector and the process of BMT in Section 3. Secondly, the experimental details and analysis are described in Section 4. Finally, our work is summarized in Section 5.

2. Related work

Early approaches [20–23] to solving reading comprehension tasks often relied on complex handcrafted feature engineering and sophisticated network architectures tailored for specific datasets. Although these methods could improve performance to some extent, they required an in-depth understanding of the data. Since these model architectures are being customized for specific datasets, they tend to perform poorly when applied to different tasks, exhibiting weak transfer learning capabilities. Consequently, the model has to be reconfigured and retrained for each new task. With the emergence of PLMs [1,24], this practice of designing complex model architectures for single datasets has gradually been replaced by more general methods. The common approach now involves fine-tuning PLMs on specific tasks or datasets, which simplifies the model design process. While fine-tuning PLMs on a single dataset can improve performance for specific task, it may lead to overfitting and impair the model's generalization ability [1,24,25].

Rather than focusing solely on optimizing performance for specific datasets, recent efforts have shifted towards enhancing the generalization capabilities of reading comprehension systems, moving from model-level improvements to data-level strategies. The current approach to improving the generalization performance of reading comprehension models primarily involves multi-task learning, where pretrained transformer models are fine-tuned on multiple reading comprehension datasets simultaneously [7,26]. This approach aims to enable the model to learn shared knowledge across different tasks. While the benefits of training on diverse datasets are evident, the process is not without its share of challenges. Managing noise, grappling with distribution variations, and reconciling differences in annotation styles across datasets present ongoing challenges. The model's effectiveness hinges on achieving a delicate balance between tailoring its parameters to specific datasets and preserving a broad applicability that transcends individual variations.

To address this, several studies have explored various multi-task sampling strategies. These strategies are devised to tackle issues related to data imbalance, ensuring a more equitable and effective learning process across various tasks. Xu et al. [27] propose a multi-task learning framework with sample reweighting, while Gottumukkala et al. [10] introduce a dynamic sampling method that adjusts sampling probabilities inversely based on each dataset's validation accuracy. Despite employing multi-task sampling strategies, fine-tuning on multiple datasets can still lead to overfitting to certain datasets, resulting in preference effects where the model performs well on some datasets but struggles on others. Other work has focused on training models to answer a broader range of question types. Khashabi et al. [8] develop UnifiedQA, a question-answering model based on the T5 architecture [24], which uses a unified text-to-text format and is trained on datasets containing various answer formats.

In contrast to the aforementioned studies, we propose BMT, a simple yet effective fine-tuning technique for multi-data question-answering tasks, from the perspective of data distribution. BMT innovatively introduces parameters related to data distribution and transforms the process of solving these distribution parameters into an optimization problem, yielding a binary mask vector closely aligned with the data distribution. During multi-data fine-tuning, the binary mask vector is used to mask gradients generated by backpropagation, thereby controlling the update of model parameters. This enables the selection of subspace parameters that align better with the data distribution, effectively mitigating the tendency of the model to over-adjust its own weights and achieving more precise fine-tuning control.

Our proposed fine-tuning technique is architecture-agnostic and can be easily applied to various models, enhancing their ability to capture regularities across datasets as well as dataset-specific attributes. Additionally, adapter modules [18,19] can be integrated as plugins within transformer blocks to improve the adaptability and task-specific performance of PLMs. Our proposed BMT can also be combined with adapters to further enhance the model's performance in both in-domain and out-of-domain datasets, as well as in zero-shot and few-shot learning scenarios, providing an innovative solution to the challenges in reading comprehension tasks.

3. Methodology

3.1. Overview of reading comprehension

In this paper, we primarily focus on extractive reading comprehension tasks. Given a context c and a related question q, the objective of a model is to extract a continuous sequence of tokens from the context c as the correct answer a to the question q by maximizing the conditional probability $p(a|c,q)$. Consequently, the task can be divided into two subtasks: determining the start and end positions of the answer. For a specific context c and question q, the position of the answer can be represented by two integers: the start position i and the end position j. The model needs to estimate the probability distribution of these two positions as follows:

- **Probability of the Start Position** i
  First, the model needs to identify the most likely start position i of the answer. This can be accomplished by calculating the conditional probability $p(start = i \mid c,q)$ and finding the i that maximizes this probability:

$$i' = \arg\max_i p(start = i \mid c,q), \tag{1}$$

  where i' represents the estimated optimal start position.
- **Probability of the End Position** j
  Second, once the start position i' is determined, the model then needs to determine the most likely end position j of the answer, which can be done by calculating the conditional probability $p(end = j \mid start = i', c,q)$ and finding the j that maximizes the probability:

$$j' = \arg\max_j p(end = j \mid start = i', c,q), \tag{2}$$

  where j' represents the estimated optimal end position.

By following these steps, we obtain a complete answer span $[i', j']$ denoting the position of the answer a. Specifically, the question q and the context c related to the question are concatenated into the "[CLS] + question + [SEP] + context + [SEP]" format, and then they are fed into the PLM. Subsequently, we feed the start token and the end token outputs from the model into a linear classifier, as shown in Fig. 2. Meanwhile, we consider datasets that are divided into source datasets $\mathcal{S}$ and target datasets $\mathcal{T}$ from the MRQA 2019 shared task. Each example D_d in every dataset D is represented in a supervised form as (q,c,a), where c, q, and a are context, question, and answer, respectively. Fig. 3 presents a typical example from a reading comprehension dataset.

3.2. Design of binary mask vector

We introduce a binary mask vector during the multi-data fine-tuning phase. The model is trained on different datasets, we should find

Table 1
Information about MRQA sub-domain datasets.

Dataset	Question (Q)	Context (C)	$\|Q\|$	$\|C\|$	Train	Dev	Test
SQuAD1.1	Crowdsourced	Wikipedia	11	137	86,588	10,507	–
NewsQA	Crowdsourced	News articles	8	599	74,160	4212	–
TriviaQA	Trivia	Web snippets	16	784	61,688	7785	–
SearchQA	Jeopardy	Web snippets	17	749	117,384	16,980	–
HotpotQA	Crowdsourced	Wikipedia	22	232	72,928	5904	–
Natural Questions	Search logs	Wikipedia	9	153	104,071	12,836	–
BioASQ	Domain experts	Science articles	11	248	–	1504	1518
DROP	Crowdsourced	Wikipedia	11	243	–	1503	1501
DuoRC	Crowdsourced	Movie plots	9	681	–	1501	1503
RACE	Domain experts	Examinations	12	349	–	674	1502
RelationExtraction	Synthetic	Wikipedia	9	30	–	2948	1500
TextbookQA	Domain experts	Textbook	11	657	–	1503	1508

Note: The information in the table is from Table 1 of Fisch et al.(2019).

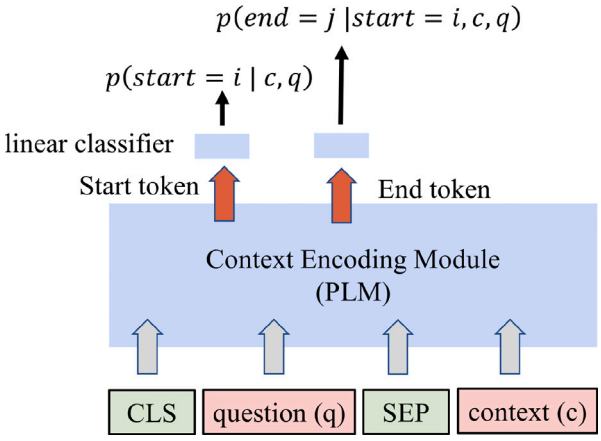


Fig. 2. Pretrained Language Model (PLM) in reading comprehension.

suitable θ and ϕ parameters to minimize the average loss of the model on the source datasets $\mathcal{S}$, and the minimization problem is calculated as follows:

$$\arg\min_{\theta,\phi} \mathbb{E}_{D_i\sim\mathcal{S}}\left[\mathbb{E}_{q,c,a\sim D_i}\left[-\log p(a \mid q,c;\theta,\phi)\right]\right]. \tag{3}$$

where θ denotes the parameters of the encoder in the pretrained model and ϕ refers to the parameters of the classification layer used to predict the start and end tokens.

Owing to the different data distributions of each batch of samples, we introduce *data-specific* parameters δ_d to capture data distributional attributes. The model parameters θ and ϕ remain fixed, and then the parameters δ_d is added to the θ and ϕ, resulting in the following structural risk minimization problem:

$$\arg\min_{\theta,\phi,\delta_d} \mathbb{E}_{D_i\sim\mathcal{S}}\left[\mathbb{E}_{q,c,a\sim D_i}\left[-\log p(a \mid q,c;\theta,\phi,\delta_d)+\lambda R\left(\theta,\phi,\delta_d\right)\right]\right]. \tag{4}$$

Note that the parameters θ and ϕ can be shared by multiple datasets, while the introduced parameters δ_d is data-specific. If the parameters δ_d can be regularized to be very sparse, this parameter-sharing approach becomes more efficient as the number of datasets increases. Therefore, we choose to regularize δ_d using the L_0-norm, i.e., $\|\delta_d\|_0 \ll \|\theta+\phi\|_0$. This approach explicitly enforces sparsity on δ_d, preventing the update of unimportant parameters, thus improving the efficiency of parameter sharing, especially when handling multiple datasets. In this way, the update process becomes more efficient without excessively deviating from the pretrained weights. Thus, we redefine the $R\left(\theta,\phi,\delta_d\right)$ as the L_0-norm of δ_d, i.e., $\|\delta_d\|_0$, the number of non-zero values in the δ_d vector, and then the structural risk minimization problem is refined as follows:

$$\arg\min_{\delta_d} \mathbb{E}_{D_i\sim\mathcal{S}}\left[\mathbb{E}_{q,c,a\sim D_i}\left[-\log p(a \mid q,c;\theta,\phi,\delta_d)+\lambda\|\delta_d\|_0\right]\right]. \tag{5}$$

Owing to the presence of a non-differentiable L_0-norm term in the optimization problem, we employ a relaxed mask vector based on the gradient method [28]. Given the L_0-norm of δ_d, we set the corresponding binary-gate variable s_d that obeys Bernoulli distribution $p(s_d;\alpha_d)$ with parameters α_d. Subsequently, δ_d can be decomposed into an element-wise product of a binary mask vector s_d and a dense vector w_d.

$$\delta_d = s_d \odot w_d, \quad s_d \in \{0,1\}^d, w_d \in \mathbb{R}^d. \tag{6}$$

Now, we can lower the bounds of the true minimization problem and optimize an expectation with respect to s_d that obeys Bernoulli distribution $p(s_d;\alpha_d)$ with parameters α_d. This process is achieved by finding suitable values for α_d and w_d as follows:

$$\arg\min_{\alpha_d,w_d} \mathbb{E}_{s_d\sim p(s_d;\alpha_d)}\left[\mathbb{E}_{D_i\sim\mathcal{S}}\left[\mathbb{E}_{q,c,a\sim D_i}\left[-\log p(a \mid q,c;\theta,\phi,\delta_d) +\lambda\|\delta_d\|_0\right]\right]\right]. \tag{7}$$

Owing to the discrete nature of s_d, this optimization problem is still complicated, e.g., it is not easy to calculate the gradient by sampling discrete s_d from Bernoulli distribution. Inspired by previous study [29,30], we map s_d to a continuous space through a stretched Hard-Concrete distribution, enabling the use of a pathwise gradient estimator to compute gradients. This design not only simplifies the optimization process but also ensures the smooth optimization of the mask vector. Furthermore, by jointly optimizing α_d and w_d, the model can dynamically adjust the weight update path, further enhancing the stability of the optimization process. Here, we first introduce the pathwise gradient estimator and the law of the unconscious statistician, which are involved in the reparameterization process.

Theorem 1. *Pathwise gradient estimator*

$$\hat{\mathbf{x}} \sim p(\mathbf{x};\theta) \equiv \hat{\mathbf{x}} = g(\hat{\epsilon},\theta), \quad \hat{\epsilon} \sim p(\epsilon) \tag{8}$$

Theorem 2. *The law of unconscious statistician*

$$\mathbb{E}_{p(\mathbf{x};\theta)}[f(\mathbf{x})] = \mathbb{E}_{p(\epsilon)}[f(g(\epsilon;\theta))] \tag{9}$$

As indicated in Theory 1, we can sample from a simpler non-parametric probability density function (pdf), i.e., $\hat{\epsilon} \sim p(\epsilon)$ and deterministically transform the samples, i.e., $\hat{\mathbf{x}} = g(\hat{\epsilon},\theta)$, rather than sampling directly from a complex pdf $p(\mathbf{x};\theta)$. Meanwhile, as indicated in Theory 2, to obtain the expectation of a function with a complex distribution, it is only necessary to introduce a simple non-parametric pdf and construct a mapping relationship from a simple distribution to a complex distribution. According to the abovementioned theories, we define s_d as a deterministic and differentiable function constructed from a variable u that obeys a uniform distribution, which can be regarded as a simple distribution mentioned above:

$$\begin{aligned} &u \sim U(\mathbf{0},\mathbf{1}) \\ &v_d = \sigma\left(\log u - \log(1-u) + \alpha_d\right) \\ &s_d = \min\left(\mathbf{1},\max\left(\mathbf{0},v_d\times(\zeta-\gamma)+\gamma\right)\right) \quad \gamma<0,\zeta>1. \end{aligned} \tag{10}$$

Context: Architecturally, the school has a Catholic character. Atop the Main Building's gold dome is a golden statue of the Virgin Mary. Immediately in front of the Main Building and facing it, is a copper statue of Christ with arms upraised with the legend "Venite Ad Me Omnes". Next to the Main Building is the Basilica of the Sacred Heart. Immediately behind the basilica is the Grotto, a Marian place of prayer and reflection. It is a replica of the grotto at Lourdes, France where the Virgin Mary reputedly appeared to Saint Bernadette Soubirous in 1858. At the end of the main drive (and in a direct line that connects through 3 statues and the Gold Dome), is a simple, modern stone statue of Mary.

1. **Question:** To whom did the Virgin Mary allegedly appear in 1858 in Lourdes France?
   Answer text: Saint Bernadette Soubirous **Answer span:** [515, 540]
2. **Question:** What is in front of the Notre Dame Main Building?
   Answer text: a copper statue of Christ **Answer span:** [188, 212]
3. **Question:** The Basilica of the Sacred heart at Notre Dame is beside to which structure?
   Answer text: the Main Building **Answer span:** [279, 295]
4. **Question:** What is the Grotto at Notre Dame?
   Answer text: a Marian place of prayer and reflection **Answer span:** [381, 419]

Fig. 3. Typical extractive reading comprehension example.

We use two constants γ and ζ to stretch $\boldsymbol{v}_d$ into the interval $[\gamma, \zeta]^d$ before it is clamped to $[0,1]^d$ through the $min(1, max(0, .))$ operation. Consequently, the expected L_0-norm of $\boldsymbol{\delta}_d$ can be written as a differentiable closed-form expression:

$$\mathbb{E}\left[\|\boldsymbol{\delta}_d\|_0\right] = \sum_{i=1}^{d} \sigma\left(\boldsymbol{\alpha}_{d,i} - \log\frac{-\gamma}{\zeta}\right). \tag{11}$$

Thus, the final optimization problem is expressed as follows:

$$\begin{aligned}\arg\min_{\boldsymbol{\alpha}_d, \boldsymbol{w}_d} \mathbb{E}_{\mathbf{u}\sim U[\mathbf{0},\mathbf{1}]}[\mathbb{E}_{D_i\sim S}[\mathbb{E}_{q,c,a\sim D_i}[-\log p(a \mid q, c; \boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{s_d} \odot \boldsymbol{w_d})]]] + \\ \lambda \sum_{i=1}^{d} \sigma\left(\boldsymbol{\alpha}_{d,i} - \log\frac{-\gamma}{\zeta}\right).\end{aligned} \tag{12}$$

Now, we can use the pathwise gradient estimators to optimize the first term with respect to $\boldsymbol{\alpha}_d$ because the expectation no longer depends on it. By sampling $\boldsymbol{u}$ once to obtain a ***binary mask vector*** $\boldsymbol{s}_d$ mentioned in Eq. (6) ($\boldsymbol{s}_d$ is not necessarily a binary vector, we can utilize the clamping function to make a part of elements of $\boldsymbol{s}_d$ equal to exactly zero), then data-specific parameters $\boldsymbol{\delta}_d = \boldsymbol{s}_d \odot \boldsymbol{w}_d$ can be set.

3.3. Binary mask tuning

The standard backpropagation process is to compute the gradient of the loss $\mathcal{L}(\mathbf{w}_t)$, and apply the gradient descent algorithm to update the model parameters, as shown in Eq. (13),

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \frac{\partial \mathcal{L}(\mathbf{w}_t)}{\partial \mathbf{w}_t}. \tag{13}$$

BMT also involves computing gradients and updating model parameters during the backward process. However, the clear distinction is that it uses $\boldsymbol{s}_d$ as a binary mask vector mentioned in Section 3.2 to update model parameters, as shown in Eq. (14),

$$\begin{aligned} m, n &= \frac{\partial \mathcal{L}(\mathbf{w}_t)}{\partial \mathbf{w}_t}.shape() \\ \boldsymbol{M}_d &= \boldsymbol{s}_d.reshape(m, n) \\ \mathbf{w}_{t+1} &= \mathbf{w}_t - \eta \frac{\partial \mathcal{L}(\mathbf{w}_t)}{\partial \mathbf{w}_t} \odot \boldsymbol{M}_d. \end{aligned} \tag{14}$$

where ".shape()" is the operation that obtains the dimension information of the matrix, and ".reshape()" is the operation that reshapes the dimensions of the matrix. $\boldsymbol{M}_d$ denotes the mask vector, and "$\odot$" represents the element-wise product. Owing to the carefully designed binary mask vector, which is closely tied to the data distribution, it can pinpoint a sub-parameter space within the vast parameter space that aligns with the distribution of the training data when updating model parameters. Therefore, the direction of parameter updates becomes more explicit.

As outlined above, the core generalization strategy of BMT is to guide the updating of model parameters through a binary mask vector associated with the data distribution, thereby balancing the commonalities across datasets and the dataset-specific attributes during multi-dataset fine-tuning. After introducing data distribution-related parameters, BMT transforms the generation of the mask vector into an optimization problem, enabling the mask vector to capture distributional differences between datasets and selectively update the most relevant subset of parameters. This effectively mitigates the decline in generalization ability caused by distributional differences. In multi-dataset fine-tuning, the model needs to simultaneously learn cross-dataset common patterns and unique features of each dataset. BMT controls the parameter updates to ensure that the model retains its pretrained weights while gradually learning the commonality across datasets, thus improving its generalization ability on new datasets. Furthermore, by restricting the range of parameter updates, BMT prevents the model from overfitting to specific datasets, thereby further enhancing overall generalization performance.

BMT incorporates technical components such as sparsity constraints, per-path gradient estimators, and adapters. The selection of these components aims to enhance the model's flexibility and robustness, ensuring its effective application across different tasks and pre-trained models, particularly in natural language processing tasks such as reading comprehension and machine translation. These technical components, combined with regularization methods and data distribution-related mask vectors, help BMT not only improve the model's generalization ability but also enhance its adaptability in cross-domain tasks. In summary, the design goal of BMT is to optimize the generalization performance of multi-dataset fine-tuning, ensuring that the model operates efficiently and stably across various application scenarios.

4. Experiments

4.1. Setup

In this paper, we employ the BMT method for machine reading comprehension tasks. Our experiments utilize datasets from the MRQA 2019 Shared Task [7], comprising six large in-domain datasets for training and six small out-of-domain datasets for testing. The format of these datasets is extractive, where the answer to each question is a segment comprising consecutive tokens within the given context. Furthermore, our primary choice for the encoder is the RoBERTa-base model [5] as the encoder, which features 12 layers, 768 hidden nodes, 12 heads, and ∼125M parameters. The specific experiments conducted are single-data fine-tuning and multi-data fine-tuning on in-domain datasets, as well as few-shot transfer learning on out-of-domain datasets. To enhance the ability of the model to learn the distribution of different datasets, we also integrate an adapter module into the encoder. In addition, we configure the corresponding adapter module for each dataset. In downstream tasks, where the majority of RoBERTa-base parameters remain fixed, fine-tuning is exclusively applied to the parameters of the different adapters, facilitating knowledge transfer across various datasets.

4.2. Description of datasets

We utilize extractive reading comprehension datasets from the MRQA 2019 shared task to evaluate the effectiveness of the proposed BMT method. Table 1 presents detailed properties for each dataset.

4.2.1. Brief introduction to in-domain datasets

- **SQuAD1.1** [31]: The renowned SQuAD (Stanford Question Answering Dataset) was introduced in 2016, featuring 107,785 question-answer pairs collected from 536 Wikipedia passages. The answer in SQuAD examples is a context span related to the question. Benefiting from the development of wide knowledge bases and the crowd workers service model, SQuAD has become the first reading comprehension dataset containing large-scale natural language questions in academia.
- **NewsQA** [32]: The NewsQA dataset includes 119,633 natural language questions derived from CNN news articles through crowd worker information processing.
- **TriviaQA** [33]: Contributors initially collect question-answer pairs on a quiz-league website, then search the web and Wikipedia for evidence to construct question-answer-evidence triples. This process results in a high-quality dataset with 95.9k question-and-answer pairs and 663k triples eventually.
- **SearchQA** [34]: The authors of the SearchQA dataset first crawl questions on Jeopardy and then retrieve these questions through Google to obtain answer snippets. The dataset eventually consists of 6.9M paragraph snippets and 140K question-answer pairs.
- **HotpotQA** [35]: Crowd workers are tasked with creating questions based on Wikipedia articles, providing supporting paragraphs and evidence for multi-hop reasoning simultaneously. To increase the difficulty of reasoning, supporting paragraphs are mixed with distracting paragraphs.
- **Natural Questions (NaturalQ)** [36]: Natural Questions dataset is collected from real queries that users request from the Google search engine.

4.2.2. A brief introduction to out-of-domain datasets

- **BioASQ** [37]: A dataset of large-scale biomedical semantic indexing and question-answering created by domain experts.
- **DROP** [38]: Crowd workers create question-answer pairs based on Wikipedia paragraphs. Answers in DROP involve various types, such as numbers, dates, or text strings, and may span multiple paragraphs.
- **DuoRC** [39]: DuoRC dataset contains 186,089 unique question-answer pairs derived from 7680 movie plot pairs, each reflecting two versions of the same movie.
- **RACE** [40]: The RACE dataset is collected from the reading comprehension test for middle and high school English in China. It includes 28,000 short essays, nearly 100,000 questions, and various topics to assess students' comprehension. The proportion of questions requiring inference is higher than in other datasets, resulting in higher precision and greater difficulty.
- **RelationExtraction (RelExt)** [41]: Crowdworkers map relationships between entities in the wiki reading dataset into question-answer pairs to gather labelled slot-filling examples, which constitutes the relation extraction reading comprehension dataset. For example, the "occupation(x, y)" relationship between entities x and y appearing in a sentence can be transformed into "What did x do for a living?" with answer y. Multiple question templates for each relationship are collected.
- **TextbookQA** [42]: TextbookQA dataset is collected from life science, earth science, and physical science textbooks for secondary schools.

4.3. Evaluation metrics

In extractive question answering (QA) tasks, the word-level F1 score (F1) is commonly used to measure the overlap between the predicted and gold answers at the word level. By allowing partial matches, F1 offers a more nuanced evaluation of the model's understanding, particularly in cases where the gold answer is long or there are slight differences in phrasing or word order, thus balancing the assessment of partial overlaps. Since F1 is more lenient and better reflects the model's comprehension ability, it is crucial for evaluating how well a model generalizes to new questions and contexts. Therefore, in this paper, we primarily adopt F1 as the main metric for evaluating the generalization capability of the QA system. To further assess the robustness of the proposed method, we also report performance on exact match (EM), which only gives credit when the predicted answer exactly matches the gold answer.

4.4. BMT in multi-data fine-tuning

Multi-data fine-tuning means that the model is trained on different datasets and appropriate parameters are found to fit distributional attributes of different datasets well, as discussed in Section 3.2.

Heterogeneous and dynamic sampling. Previous studies [10,43] have demonstrated that sampling heterogeneous batches can enhance the model's generalization performance. The term "heterogeneous" implies that the samples are derived from different datasets or exhibit diversity, whereas "dynamic" suggests that the sampling approach is adaptable and changes over time or based on certain criteria. Drawing inspiration from this concept, we introduce a straightforward dynamic sampling strategy during multi-data fine-tuning. We determine the proportion of each dataset in the next sampling mixed batch based on the difference between the current validation accuracy (EM+F1) and the previous validation accuracy on that dataset, as shown in Eq. (15).

$$weight_i = \frac{|different_i|}{\sum_j |difference_j|}, \tag{15}$$

where $weight_i$ denotes the normalized proportion of the ith dataset in each mixed batch.

Training Details on the source datasets. We train the model on six in-domain datasets and compare the F1 score on single-data fine-tuning and multi-data fine-tuning. The model is implemented using the Pytorch [44] deep learning framework, and we primarily use RoBERTa-base [5] (L = 12, H = 768, A = 12) as our PLM to obtain the contextual

Table 2
Influence of mask on multi-data fine-tuning.

Model	SQuAD1.1	HotpotQA	TriviaQA	NewsQA	SearchQA	NaturalQ	Avg.
RoBERTa-base							
MT(w/o mask)	91.8	81.1	80.2	71.5	84.9	79.5	81.5
MT(w/ mask)	92.0	81.0	80.6	71.7	85.0	79.8	**81.7**(↑ 0.2)
RoBERTa-large							
MT(w/o mask)	93.4	82.5	82.9	72.2	84.6	80.7	82.7
MT(w/ mask)	93.4	83.3	83.4	73.3	85.0	81.0	**83.2**(↑ 0.5)
T5-base							
MT(w/o mask)	89.8	76.2	58.6	46.3	64.8	73.0	68.1
MT(w/ mask)	90.0	76.7	58.9	47.9	65.9	74.0	**68.9**(↑ 0.8)

Note: MT means multi-data fine-tuning. Avg means average.

embeddings of questions and contexts in our experiments. The initialization weights for RoBERTa-base are derived from the Hugging Face [45], serving as our starting point for fine-tuning. During the single-data fine-tuning phase, we randomly sample from the dataset to construct mini-batches, whereas we use dynamic sampling to construct mini-batches during the multi-data fine-tuning phase. Specifically, we sample 80,000 training and 1000 validation examples per epoch and then save checkpoints every 2048 steps. If the validation F1 score does not improve after five checkpoints, the model will be terminated to train; otherwise, the model will continue to be trained on a single dataset up to 10 epochs and multiple datasets up to 3 epochs. With respect to learning rate settings, we use a fixed learning rate of 1e−5 for transformer parameters and the AdamW optimizer [46] following the HuggingFace default parameters. We use V100 GPUs with a 32 GB memory to train all the models, and it takes a day and a half to train a multi-data model.

How to handle long text? When the length of the input tokens exceeds the RoBERTa maximum length limit of 512 tokens, we apply a sliding window of length 256 tokens to segment the input into multiple "chunks". Each chunk is then concatenated with "[CLS]", questions and special separator tokens to generate the complete input. During prediction, the output is determined by selecting the chunk with the highest scoring probability.

4.4.1. Effectiveness of randomly generated mask

To evaluate the influence of the carefully designed $\boldsymbol{M}_d$ derived from $\mathbf{s}_d$ on multi-data fine-tuning, we follow child-tuning [17] and utilize a randomly generated 0–1 mask, $\boldsymbol{m}_d$, drawn from a Bernoulli distribution with a probability p_F. This mask is employed to update model parameters, serving as a control in our study:

$$\mathbf{m}_d \sim Bernoulli\left(p_F\right). \tag{16}$$

In the experiment, we sample 0–1 masks from a Bernoulli distribution with varying parameters $p_F \in \{0.3, 0.5, 0.7\}$. The experimental results in Table A.10 of the Appendix indicate that when $p_F = 0.5$, multi-data fine-tuning yields the best performance. Therefore, we set p_F to 0.5, and the calculation process is depicted in the third line of Eq. (14). As shown in Table 2, the use of $\boldsymbol{m}_d$ generated by the Bernoulli distribution improves the average F1 score across all datasets over the corresponding RoBERTa-base without $\boldsymbol{m}_d$ by 0.2 points, RoBERTa-large by 0.5 points, and T5-base by 0.8 points respectively. The limited improvement indicates that the randomly generated mask $\boldsymbol{m}_d$ is helpful, naturally inspiring us to improve the randomly generated mask to obtain better performance. Therefore, we will introduce the carefully designed $\boldsymbol{M}_d$ (vs. without any modification $\boldsymbol{m}_d$) on multi-data fine-tuning.

4.4.2. Effectiveness of BMT in multi-data fine-tuning

In this section, we will explore the impact of the proposed BMT on multi-data fine-tuning. In a brief recap, the BMT method consists of two parts. First, $\boldsymbol{M}_d$ is obtained from the binary mask vector $\boldsymbol{s}_d$. Second, $\boldsymbol{M}_d$ is applied to update the model parameters with respect to the gradients. As shown in Table 3, the model with BMT improves the average F1 score across all datasets over the corresponding RoBERTa-base with mask by 1.2 points, RoBERTa-large by 1.5 points, and T5-base by 1.5 points, respectively. This demonstrates the effectiveness of our proposed method, implying that BMT can effectively alleviate the model's tendency to excessively adjust its own weights and capture cross-dataset regularities and dataset-specific attributes of different datasets.

Additionally, since this study is under the context of the MRQA 2019 Shared Task, which aims to evaluate the generalization ability of reading comprehension systems, our supplementary reading comprehension models primarily focus on enhancing the generalization ability of such systems, rather than solely optimizing performance on a specific dataset. We compare our proposed method, MT (w/ BMT), with other models that focus on improving generalization in reading comprehension tasks. Table 4 presents the performance (F1 scores) on the MRQA datasets, showing that MT (w/ BMT) achieves strong performance across all datasets. On average (Avg. F1), our method shows a 5 points improvement over Multi-Task $\text{BERT}_{\text{large}}$, a 4.4 points improvement over the $\text{BERT}_{\text{large}}$ model fine-tuned separately on each dataset, and a 1.5 points improvement over the multi-dataset fine-tuning method Dynamic Sampling. Moreover, it even outperforms the competitive LinkBERT (RoBERTa-base) by 0.6 points. These results indicate that the BMT method can significantly enhance the generalization ability of model in reading comprehension tasks under multi-dataset fine-tuning scenarios.

4.4.3. Difference between single-data fine-tuning and multi-data fine-tuning

To further verify the effectiveness of the BMT method in single-data fine-tuning, we compare single-data fine-tuning with multi-data fine-tuning. The difference between single-data fine-tuning and multi-data fine-tuning is that the former only uses a single dataset to fine-tune the model, whereas the latter trains the model on mixed datasets via dynamic sampling. As shown in Table 5, the multi-data fine-tuning with BMT improves the average F1 score across all datasets over the corresponding single-data fine-tuning model by 1.8 points (82.9 vs. 81.1), implying that the BMT method is more effective in multi-data fine-tuning. The model converges on different datasets at different times, and it needs to find a balance to maximize its performance on all datasets, which imposes a regularization effect on the model, thus indirectly enhancing the generalization ability of the model. Meanwhile, multi-data fine-tuning, particularly by introducing different datasets through dynamic sampling, allows the model to encode cross-dataset regularities and shared information that exists in multiple datasets, thus improving performance.

In addition, we also find that single-data fine-tuning with BMT improves the average F1 score over itself with mask by 0.2 points, and multi-data fine-tuning with BMT improves the average F1 score over itself with mask by 1.2 points, implying that BMT is more suitable for multi-data fine-tuning scenarios and weakly influences single-data fine-tuning. Therefore, based on the above experimental results, we introduce the adapter [18,19] in multi-data fine-tuning to further explore the effectiveness of BMT.

Table 3
BMT in the multi-data fine-tuning.

Model	SQuAD1.1	HotpotQA	TriviaQA	NewsQA	SearchQA	NaturalQ	Avg.
RoBERTa-base							
MT(w/ mask)	92.0	81.0	80.6	71.7	85.0	79.8	81.7
MT(w/ BMT)	92.4	83.3	81.4	73.0	85.5	81.8	**82.9**(↑ 1.2)
RoBERTa-large							
MT(w/ mask)	93.4	83.3	83.4	73.3	85.0	81.0	83.2
MT(w/ BMT)	94.2	85.2	84.7	75.2	86.3	82.8	**84.7**(↑ 1.5)
T5-base							
MT(w/ mask)	90.0	76.7	58.9	47.9	65.9	74.0	68.9
MT(w/ BMT)	90.3	78.9	60.2	49.4	67.5	76.3	**70.4**(↑ 1.5)

Note: BMT means Binary Mask Tuning.

Table 4
Performance (F1) on the six MRQA extractive question answering datasets (in-domain datasets).

Model	SQuAD1.1	HotpotQA	TriviaQA	NewsQA	SearchQA	NaturalQ	Avg.
Multi-Task $\text{BERT}_{\text{base}}$	86.7	76.6	71.6	66.8	76.7	65.4	74.0
Multi-Task $\text{BERT}_{\text{large}}$	88.4	79.0	74.7	66.3	79.0	79.8	77.9
$\text{BERT}_{\text{base}}$	88.7	76.0	70.3	65.7	74.2	76.5	75.2
$\text{BERT}_{\text{large}}$	91.1	78.1	73.7	70.9	78.3	79.0	78.5
$\text{LinkBERT}_{\text{base}}$	90.1	78.2	73.9	69.3	76.8	78.3	77.8
$\text{LinkBERT}_{\text{large}}$	92.7	80.8	78.2	72.6	80.5	81.0	81.0
LinkBERT(RoBERT-base)	92.9	82.7	80.9	72.8	82.8	81.4	82.3
Dynamic sampling (BERT-base)	88.2	78.4	76.1	67.8	79.6	78.3	78.1
Dynamic sampling (RoBERT-base)	91.4	80.9	80.5	71.4	84.6	79.8	81.4
MT(w/ BMT)	**92.4**	**83.3**	**81.4**	**73.0**	**85.5**	**81.8**	**82.9**

Table 5
Difference between single-data fine-tuning and multi-data fine-tuning.

Model	SQuAD1.1	HotpotQA	TriviaQA	NewsQA	SearchQA	NaturalQ	Avg.
RoBERTa-base							
ST(w/o mask)	90.7	78.8	78.7	70.6	85.0	79.0	80.5
ST(w/ mask)	91.3	78.4	79.6	71.6	85.2	79.2	80.9
ST(w/ BMT)	91.6	78.5	79.9	71.8	84.8	80.1	**81.1**(↑ 0.2)
MT(w/o mask)	91.8	81.1	80.2	71.5	84.9	79.5	81.5
MT(w/ mask)	92.0	81.0	80.6	71.7	85.0	79.8	81.7
MT(w/ BMT)	92.4	83.3	81.4	73.0	85.5	81.8	**82.9** (↑ 1.2)

Note: ST means single-data fine-tuning. MT means multi-data fine-tuning.

4.5. Effectiveness of BMT in multi-data fine-tuning with adapters

Although multi-data fine-tuning can improve the generalization of the model, the cost of fine-tuning will become expensive as the number of datasets increases. When new datasets are added, the model parameters need to be retrained, resulting in catastrophic forgetting of what has already been learned. Besides, the manner in which the model learns different information from different datasets is not decoupled, and new datasets will interfere with the knowledge that has been acquired by the model. Therefore, in this section, we introduce an adapter into multi-data fine-tuning to reduce the cost of model training and improve parameter efficiency.

The adapter is a module related to a specific task or knowledge or data source, often embedded as a plug-in in the transformer block, whose input comes from the hidden state of the middle layer of PLM. In this section, we still use the RoBERTa-base as the pretrained model and apply the default adapter configuration from Houlsby [18]. When faced with various downstream tasks, most of the PLM parameters are frozen, and only a small number of parameters related to specific tasks need to be adjusted, considerably improving PLM scalability and practicability. For continuous learning across tasks, it is only necessary to train the corresponding adapters in the model to avoid forgetting past knowledge when learning new tasks.

Based on the abovementioned considerations, we divide the parameters of the model into three parts, including the encoder $\boldsymbol{\theta}$ shared by all datasets, the token classifier set of dataset-specific $\boldsymbol{\phi} = \{\phi_1, \ldots, \phi_{|S|}\}$, and the adapter set $\boldsymbol{\psi} = \{\psi_1, \ldots, \psi_{|S|}\}$, as shown in Fig. 4. "w/ adapter" denotes that we train all parameters of the model $(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\psi})$ on the source datasets S. When the average F1 score of the source datasets no longer improves, we freeze the encoder's parameter $\boldsymbol{\theta}$ and continue to fine-tune each pair of (ϕ_i, ψ_i) on each dataset separately. Therefore, our goal is to find suitable $\boldsymbol{\theta}$, $\boldsymbol{\phi}$, and $\boldsymbol{\psi}$ that minimize the expectation of the model on all source datasets, and the final optimization problem is given by,

$$\arg\min_{\boldsymbol{\theta},\boldsymbol{\phi},\boldsymbol{\psi}} \mathbb{E}_{\mathcal{D}_i \sim S}\left[\mathbb{E}_{q,c,a\sim\mathcal{D}_i}\left[-\log p_{\boldsymbol{\theta},\phi_i,\psi_i}(a \mid q, c)\right]\right]. \tag{17}$$

At this point, we obtain a binary mask vector based on the method mentioned in Section 3.2 and update the model parameters using BMT as outlined in Section 3.3. As shown in Table 6, multi-data fine-tuning with the adapter but without BMT (w/ adapter, w/o BMT) improves the average F1 score over the corresponding case without both the adapter and BMT simultaneously (w/o adapter, w/o BMT) by 0.5 points (82.2 vs. 81.7). This indicates that the shared parameters can encode cross-dataset regularities while different adapters model the sub-distributions, resulting in more accurate and robust generalization ability across all datasets. We hypothesize that the adapter enhances the continuous learning ability of the model. During the knowledge injection process, the parameters of the original PLM remain unchanged, ensuring that subsequent knowledge does not impact the previously added knowledge. Moreover, we also observe that multi-data fine-tuning with both the adapter and BMT simultaneously resulted in an average F1 score improvement of 1.3 points compared with that of

Table 6
BMT in multi-data fine-tuning with adapter.

Model	SQuAD1.1		HotpotQA		TriviaQA		NewsQA		SearchQA		NaturalQ		Avg.	
	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
MT(w/o adapter,w/o BMT)	85.9	92.0	65.1	81.0	75.7	80.6	55.8	71.7	79.3	85.0	67.2	79.8	71.5	81.7
MT(w/o adapter,w/ BMT)	86.8	92.4	66.9	83.3	77.2	81.4	56.6	73.0	80.7	85.5	68.4	81.8	72.8	82.9
MT(w/ adapter,w/o BMT)	86.6	92.4	65.5	81.5	75.6	80.5	56.2	72.1	80.4	85.8	68.1	80.9	72.1	82.2
MT(w/ adapter,w/ BMT)	87.9	93.0	66.8	82.9	78.6	82.1	57.5	74.2	81.1	86.2	69.2	82.3	**73.5**	**83.5**

Table 7
Comparison of model parameter efficiency.

Model	Encoder θ	Token classifier ϕ	Adapter ψ	Dataset-specific δ_d	Total parameters	Trainable parameters	Trainable parameters Percentage (%)
MT(w/o adapter, w/o BMT)	✓	✓	✗	✗	124.6M	124.6M	100%
MT(w/o adapter, w/ BMT)	✓	✓	✗	✓	131.9M	131.9M	100%
MT(w/ adapter, w/o BMT)	✓	✓	✓	✗	135.4M	1.8M	1.3%
MT(w/ adapter, w/ BMT)	✓	✓	✓	✓	142.7M	9.1M	6.4%

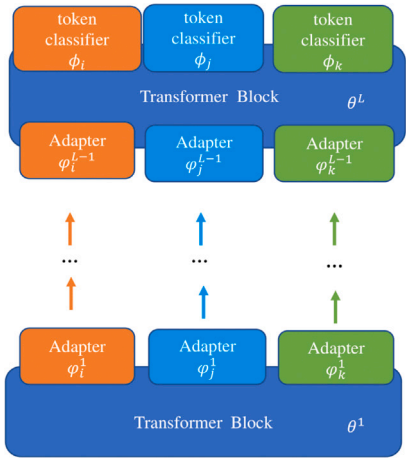


Fig. 4. Parameters of the model consist of the token classifier set of dataset-specific ϕ and the adapter set ψ with a shared transformer encoder θ.

using the adapter alone (83.5 vs. 82.2). This indicates that BMT method has a promotive effect on the adapter. Simultaneously employing BMT and the adapter can further alleviate the tendency of fine-tuned model to excessively adjust its weights during training on multiple datasets and mitigate catastrophic forgetting. Additionally, we provide a comparison of the involvement of different parameter types during training for various multi-data fine-tuning methods, as well as the total number of parameters, the number of trainable parameters, and the parameter utilization efficiency for each method, as shown in Table 7. When adapters are introduced, the number of trainable parameters significantly decreases, leading to improved parameter utilization efficiency and, consequently, lower training costs.

4.6. Effectiveness of BMT in transfer learning

In this paper, we also study the influence of BMT on transfer learning, exploring a way to build a more generalized model. Specifically, we consider two transfer learning settings: zero-shot generalization and few-shot transfer learning. In both settings, we utilize RoBERTa-base as the pretrained model and apply the default adapter configuration from Houlsby [18]. The model parameters are still divided into three parts, including the encoder θ shared by all datasets, the token classifier set of dataset-specific $\phi = \{\phi_1, \ldots, \phi_{|S|}\}$, and the adapter set $\psi = \{\psi_1, \ldots, \psi_{|S|}\}$. As discussed in Section 4.4, we have applied the BMT method on in-domain datasets using multi-data fine-tuning with adapters, obtaining the fine-tuned model parameters (θ, ϕ, ψ).

4.6.1. Zero-shot generalization

For zero-shot generalization, we use a simple strategy to extend the BMT method to out-of-domain datasets: we initialize a new adapter and classifier by averaging the parameters of the adapters $\{\psi_1, \ldots, \psi_{|S|}\}$ and the token classifiers $\{\phi_1, \ldots, \phi_{|S|}\}$ fine-tuned on the in-domain datasets. This allowed us to easily transfer the cross-dataset regularities and dataset-specific attributes learned from in-domain datasets to out-of-domain (unseen) datasets. We then compute the answer with the highest probability using this new model on the out-of-domain datasets.

The experimental results of zero-shot generalization are shown in Table 8. MT (w/o adapter, w/o BMT) serves as the multi-dataset fine-tuning baseline, while UnifiedQA, trained on different architectures and datasets, represents the current state-of-the-art multi-dataset QA transfer model. We chose to compare against UnifiedQA-base, as its number of encoder parameters is similar to that of RoBERTa-base. Compared to the baseline MT (w/o adapter, w/o BMT) and UnifiedQA, our proposed MT (w/o adapter, w/ BMT) shows better zero-shot generalization performance. Moreover, averaging the parameters of different adapters yields favourable results, with MT (w/ adapter, w/ BMT) outperforming MT (w/o adapter, w/ BMT) in terms of overall generalization performance, further improving the model's zero-shot generalization (Avg. F1 from 63.3 to 63.8). This indicates that averaging the adapter and classifier parameters provides a better initialization, offering a solid starting point for zero-shot generalization. Additionally, the adapter enhances the model's continual learning ability, enabling it to effectively capture cross-dataset regularities in question-answering tasks and exhibit strong performance in zero-shot transfer learning scenarios.

We also observe that the improvement in zero-shot generalization performance from the BMT method is significantly greater than that from using the adapter alone. We hypothesize that this is because the BMT method, during multi-dataset fine-tuning, is better able to capture parameters aligned with the data distribution. Furthermore, the combination of the adapter and BMT can further enhance the model's generalization capability.

4.6.2. Few-shot transfer learning

We also consider a few-shot transfer learning setting. Specifically, we first compute the few-shot loss for each adapter using a small number of out-of-domain datasets with labels. Subsequently, we assign weights to individual adapters based on their few-shot losses. While keeping the shared encoder θ fixed, we utilize (ϕ', ψ'), $\phi' = \frac{1}{|S|}\sum_i^{|S|} \alpha_i \phi_i$, and $\psi' = \frac{1}{|S|}\sum_i^{|S|} \alpha_i \psi_i$ to initialize the pretrained model with the adapter. Finally, we continuously fine-tune θ and (ϕ', ψ') on the out-of-domain datasets. The few-shot transfer learning process is conducted with different random seeds (specifically, three random seeds are generated). The detailed training procedure is outlined as follows, and the final result is the average F1 score across different

Table 8
The performance of zero-shot generalization.

Model	BioASQ		DROP		DuoRC		RACE		RelExt		TextbookQA		Avg.	
	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
UnifiedQA-base	48.0	59.5	36.5	44.9	24.6	29.8	39.5	52.1	71.6	81.6	31.0	33.4	41.9	50.2
MT(w/o adapter, w/o BMT)	53.1	66.2	41.5	51.9	51.3	63.7	36.2	46.2	75.4	86.8	48.9	58.2	51.1	62.2
MT(w/o adapter, w/ BMT)	54.7	67.3	43.4	53.4	53.2	64.6	37.7	46.9	77.3	87.6	50.7	59.7	52.8	63.3
MT(w/ adapter, w/o BMT)	54.1	66.7	42.8	52.6	52.2	64.1	37.1	46.4	76.3	87.1	49.9	59.2	52.1	62.7
MT(w/ adapter, w/ BMT)	**55.4**	**68.1**	**43.7**	**53.9**	**53.6**	**65.3**	**38.9**	**47.2**	**78.2**	**87.9**	**51.4**	**60.1**	**53.5**	**63.8**

Table 9
The performance of few-shot transfer learning.

K	Model	BioASQ		DROP		DuoRC		RACE		RelExt		TextbookQA		Avg.	
		EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
16	UnifiedQA-base	51.9	59.7	38.5	45.2	27.9	30.2	41.5	52.9	73.8	82.0	34.2	33.8	44.6	50.6
	MT(w/ adapter,w/o BMT)	55.1	67.0	43.7	52.8	53.3	65.7	38.4	47.5	77.9	87.7	50.9	59.8	53.2	63.4
	MT(w/ adapter,w/ BMT)	56.8	68.9	44.9	54.2	52.6	64.2	40.2	48.3	76.9	87.3	52.5	60.4	54.0	63.9
64	UnifiedQA-base	54.7	65.5	42.9	47.9	31.7	32.9	45.3	54.3	75.7	83.4	35.4	35.6	47.6	53.3
	MT(w/ adapter,w/o BMT)	56.9	69.4	45.5	55.9	53.8	66.8	39.8	47.6	78.2	88.4	52.2	60.2	54.4	64.7
	MT(w/ adapter,w/ BMT)	58.3	71.2	47.6	58.0	54.9	67.2	41.8	48.4	78.9	88.7	54.1	62.1	55.9	65.9
256	UnifiedQA-base	59.4	71.7	47.7	49.8	32.7	36.5	49.4	55.2	79.2	85.6	35.9	36.8	50.7	55.9
	MT(w/ adapter,w/o BMT)	60.1	73.9	46.5	56.9	54.2	65.3	42.6	48.7	79.7	88.9	53.4	61.5	56.1	65.9
	MT(w/ adapter,w/ BMT)	**60.9**	**74.7**	**48.4**	**58.9**	**56.1**	**68.2**	**43.3**	**49.3**	**80.5**	**89.5**	**54.6**	**62.2**	**57.3**	**67.1**

random seeds: (1) We randomly sample K examples from each out-of-domain dataset as the training data. (2) The few-shot loss of each adapter is independently computed independently on the K training examples. We assign corresponding weights to each adapter based on its few-shot loss. Subsequently, the average adapter parameters are calculated and utilized as the initialization weight for the model. (3) We set aside $K/2$ examples from K training examples as the training set and the remaining $K/2$ as the validation set. The adapter learning rate is set to 1e−5, with other training hyperparameters identical to those used on the in-domain dataset. We apply BMT to tune model parameters $(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\psi})$ on the training set to 300 steps or until the training is halted when the F1 score on the validation set no longer improves after 10 epochs. (4) The model is then tested on the entire development set.

The experimental results are shown in Table 9. When the training data size is small ($K = 16$), the few-shot transfer learning performance of MT (w/ adapter, w/o BMT) shows a more significant improvement compared to the baseline UnifiedQA-base [8] (Avg. F1 increased from 50.6 to 63.4). This indicates that the weighted adapter parameters provide a better initialization, enhancing the model's continual learning capability. The model with adapters is less prone to forgetting previously learned knowledge when exposed to new data, thus providing a solid starting point for few-shot transfer learning. Moreover, as the training data size increases, the impact of BMT on the model's generalization performance becomes more pronounced. From $K = 16$ to $K = 256$, the gap in Avg. F1 between MT (w/ adapter, w/ BMT) and MT (w/ adapter, w/o BMT) increases from 0.5 points to 1.2 points. This implies that with more fine-tuning samples, BMT enables the multi-dataset fine-tuning model to learn parameters that better align with the data distribution. As a result, the few-shot transfer learning performance continues to improve.

Another interesting experimental phenomenon is that, in the few-shot transfer learning setting with $K = 16$, the multi-data fine-tuning model without BMT outperforms the one with BMT on the DuoRC and RelEx datasets. Compared with other reading comprehension datasets, DuoRC and RelEx exhibit more complex characteristics: The questions in DuoRC require integrating information from multiple sentences or paragraphs to answer, thus necessitating the model to have the capability to handle long-range dependencies. On the other hand, RelEx requires a cross-paragraph understanding of semantic relations between entities. This inherent complexity in both datasets poses challenges for fine-tuning. We hypothesize that this phenomenon may be attributed to the limited number of samples used for few-shot transfer learning and the inherent complexity of the datasets, resulting in inaccurate parameter initialization of the multi-dataset fine-tuning model with BMT. As fine-tuning continues with a restricted number of samples, this inaccuracy is further exacerbated, ultimately resulting in inferior performance compared to fine-tuning without BMT. However, as the number of samples used for few-shot transfer learning gradually increases (e.g., when K is increased to 64 or 256), the parameter initialization of the multi-data fine-tuning model with BMT becomes more reliable and accurate. With an increase in the amount of fine-tuning data, the adjustment of dataset-specific token classifier parameters and adapter parameters becomes more precise. Consequently, the model that employs the BMT method gradually exhibits superior performance on the DuoRC and RelExt datasets.

5. Conclusion

In this paper, we propose a novel fine-tuning method, BMT. By carefully designing a binary mask vector closely related to the data distribution to mask gradients, the adaptability of the model to new data distribution is enhanced, improving the efficiency of the model parameters. Detailed experiments demonstrate the effectiveness of our proposed method that BMT is not only effective in mitigating the model's tendency to excessively adjust its own weights but also in better capturing cross-dataset regularities and dataset-specific attributes of different datasets in question-answering tasks, improving the generalization ability of the model on out-of-domain datasets and few-shot transfer learning scenarios. In the future, we will explore better-performing fine-tuning methods for question-answering tasks.

CRediT authorship contribution statement

Chuanyang Gong: Writing – review & editing, Writing – original draft, Validation, Software, Methodology, Investigation, Conceptualization. **Zhihua Wei:** Supervision. **Ping Zhu:** Investigation. **Duoqian Miao:** Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The work is partially supported by the National Nature Science Foundation of China (No. 61976160, 61906137, 61976158, 62076184, 62076182) and Shanghai Science and Technology Plan Project, China (No. 21DZ1204800) and Technology research plan project of the Ministry of Public and Security, China (Grant No. 2020JSYJD01).

Appendix A. Hyperparameter experiments

We conduct additional experiments on hyperparameter tuning for the multi-data mask fine-tuning method MT (w/ mask). MT (w/ mask) follows the child-tuning work, randomly sampling 0–1 masks from a Bernoulli distribution to mask gradients and controlling the model parameter updates. The purpose of this experiment is to investigate the impact of masking on multi-data fine-tuning, providing insights for designing more effective masks in future work. We sampled 0–1 masks from a Bernoulli distribution with varying parameters $p_F \in \{0.3, 0.5, 0.7\}$. The results in Table A.10 indicate that when $p_F = 0.5$, the multi-data fine-tuning yields the best performance.

Table A.10
Performance comparison of different p_F on multi-data mask fine-tuning (MT(w/mask)).

p_F	SQuAD1.1	HotpotQA	TriviaQA	NewsQA	SearchQA	NaturalQ	Avg.
RoBERTa-base							
0.3	92.2	81.1	79.7	72.0	84.3	79.2	81.4
0.5	92.0	81.0	80.6	71.7	85.0	79.8	**81.7**
0.7	91.7	80.8	80.2	71.6	84.5	79.1	81.3
RoBERTa-large							
0.3	93.5	83.3	82.8	73.4	85.6	80.2	83.1
0.5	93.4	83.3	83.4	73.3	85.0	81.0	**83.2**
0.7	92.9	82.7	83.1	73.0	85.5	80.3	82.9
T5-base							
0.3	89.5	76.1	57.7	46.3	64.5	72.9	67.8
0.5	90.0	76.7	58.9	47.9	65.9	74.0	**68.9**
0.7	89.4	75.8	57.2	46.3	64.1	72.6	67.6

Appendix B. Notations and explanations

See Table B.11.

Table B.11
Notations and Explanations.

Notation	Explanation
$\boldsymbol{\theta}$	Encoder parameters
$\boldsymbol{\phi}$	Token classifier parameters
$\boldsymbol{\psi}$	Adapter parameters
$\boldsymbol{\delta}_d$	Data-specific parameters
$\mathcal{S}$	Source datasets
$\mathcal{T}$	Target datasets
D_i	The ith mini-batch sampled from dataset $\mathcal{S}$
(q, c, a)	QA instance
$\mathbf{m}_d \sim Bernoulli(p_F)$	A random mask vector $\mathbf{m}_d$ following a Bernoulli distribution with parameter p_F.
$\mathbf{s}_d \sim p(\mathbf{s}_d; \boldsymbol{\alpha}_d)$	$\mathbf{s}_d$ that obeys Bernoulli distribution $p(\boldsymbol{s}_d; \boldsymbol{\alpha}_d)$ with parameters $\boldsymbol{\alpha}_d$
$\boldsymbol{u} \sim U(0, 1)$	A variable $\boldsymbol{u}$ that obeys a uniform distribution
$\boldsymbol{s}_d$	Binary mask vector
$\boldsymbol{M}_d$	Mask vector

Appendix C. Additional examples for error analysis

We randomly select some test examples from six in-domain datasets, and the answer F1 score to these samples is 0. Then, we provide the model's prediction results (Four settings, e.g., P1, P2, P3, and P4). Q, A, and P represent the question, answer, and prediction, respectively. Through a detailed analysis of each example, we find that these errors can be primarily categorized into three types: (1) Annotation: Errors in the annotations provided by the dataset itself; (2) Commonsense & external knowledge: Answering certain questions requires a combination of common sense and the introduction of external knowledge; (3) Discrete reasoning: Certain questions necessitate comparing two entities or multiple relationships, and correct answers can only be obtained through discrete reasoning.

Category: **Annotation**

ID: b09d661a89274e0eb49e22879a3e7180

Dataset: HotpotQA
Q: Which TV series, written by the creators of Robocalypse, is set in Bikini Bottom?
A: SpongeBob SquarePants
P1: SpongeBob SquarePant (MT w/o mask)
P2: SpongeBob SquarePant (MT w/ mask)
P3: SpongeBob SquarePant (MT w/ BMT)
P4: SpongeBob SquarePant (MT w/ BMT,w/ adapter)

ID: 270e1adfc99940aaa1e0b6299d9560d2
Dataset: NaturalQuestion
Q: on which river did the exploration of the LouisiaPurchasease begin?
A: The Missouri River
P1: River (MT w/o mask)
P2: Missouri River (MT w/ mask)
P3: Missouri River (MT w/ BMT)
P4: Missouri River's (MT w/ BMT,w/ adapter)

ID: b4f35804350540b987525eeb1722bb7e
Dataset: HotpotQA
Q: Moss Side railway station is on Blackpool South railway station which is how far from Waterloo Road tram stop?
A: about 500 m
P1: 500 (MT w/o mask)
P2: 500 m (MT w/ mask)
P3: 500 m (MT w/ BMT)
P4: 500 m (MT w/ BMT,w/ adapter)

ID: 342cb6f69f6449318f4c4e5385f954fd
Dataset: NewsQA
Q: How do you send in your video?
A: Use the iReport form
P1: iReport form (MT w/o mask)
P2: iReport form (MT w/ mask)
P3: iReport form (MT w/ BMT)
P4: iReport form (MT w/ BMT,w/ adapter)

Category: **Commonsense & External Knowledge**

ID: e2b115ca7b1b4a418f5cf884e202de21
Dataset: NaturalQuestions
Q: Which word means separation of church and state?
A: separationism
P1: accommodationism (MT w/o mask)
P2: accommodationism (MT w/ mask)
P3: accommodationism (MT w/ BMT)
P4: accommodationism (MT w/ BMT,w/ adapter)

ID: 7d393a39916342fba58b4ba8d675dcf5
Dataset: HotpotQA
Q: What star in the film Up at the Villa has earned the title of knighthood?
A: Sir Derek George Jacobi
P1: Derek Jacobi (MT w/o mask)
P2: Derek Jacobi (MT w/ mask)
P3: Derek Jacobi (MT w/ BMT)
P4: Derek Jacobi (MT w/ BMT,w/ adapter)

ID: 9ec6f9f9af044b988a7d21f9d2b88a20
Dataset: SearchQA
Q: Which organization was formally born on Oct. 24, 1945, with the Soviet ratification of the charter?
A: the United Nations
P1: American (MT w/o mask)
P2: American (MT w/ mask)
P3: American (MT w/ BMT)
P4: American (MT w/ BMT,w/ adapter)

ID: 6811e66b5ab54ede8a1299043e3a3210
Dataset: HotpotQA
Q: In the multi-sport games usually held every four years between nations around the Mediterranean Sea, Fatma Lanouar is best known for winning gold in which event?
A: 1500 metres
P1: women 1500 metres (MT w/o mask)
P2: women 1500 m (MT w/ mask)
P3: women 1500 m (MT w/ BMT)
P4: women's 1500 m (MT w/ BMT,w/ adapter)
Category: **Discrete Reasoning**

ID: 86d8d9a1eabd499ebfb270616e2885ea
Dataset: HotpotQA
Q: Which genus has more species, Quesnelia or Honeysuckle?
A: Honeysuckle
P1: Honeysuckles (MT w/o mask)
P2: Honeysuckles (MT w/ mask)
P3: Honeysuckles (MT w/ BMT)
P4: Honeysuckles (MT w/ BMT,w/ adapter)

ID: b6f46f53c113423db4f7282a84208cfd
Dataset: SearchQA
Q: Were both Life magazine and Strictly Slots magazine published monthly in 1998?
A: yes
P1: no (MT w/o mask)
P2: no (MT w/ mask)
P3: no (MT w/ BMT)
P4: no (MT w/ BMT,w/ adapter)

ID: 5192ae0bd23340e9b7f523b005bf7e39
Dataset: NaturalQuestions
Q: Who was born earlier, Johnny Lujack or Jim Kelly?
A: Jim Kelly
P1: Christopher Lujack (MT w/o mask)
P2: Jim Kelly's (MT w/ mask)
P3: Jim Kelly's (MT w/ BMT)
P4: Jim Kelly's (MT w/ BMT,w/ adapter)

ID: 832e5c8b6780443195d48ee21694d242
Dataset: TriviaQA
Q: Were both Life magazine and Strictly Slots magazine published monthly in 1998?
A: yes
P1: no (MT w/o mask)
P2: no (MT w/ mask)
P3: no (MT w/ BMT)
P4: no (MT w/ BMT,w/ adapter)

Data availability

Data will be made available on request.

References

[1] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, 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, 2019, pp. 4171–4186.

[2] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, et al., Improving language understanding by generative pre-training, 2018.

[3] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al., Language models are unsupervised multitask learners, OpenAI Blog 1 (2019) 9.

[4] T. Brown, B. Mann, N. Ryder, M. Subbiah, J.D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, in: Advances in Neural Information Processing Systems, Vol. 33, 2020, pp. 1877–1901.

[5] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, 2019, arXiv preprint arXiv:1907.11692.

[6] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R.R. Salakhutdinov, Q.V. Le, Xlnet: Generalized autoregressive pretraining for language understanding, Adv. Neural Inf. Process. Syst. 32 (2019).

[7] A. Fisch, A. Talmor, R. Jia, M. Seo, E. Choi, D. Chen, MRQA 2019 shared task: Evaluating generalization in reading comprehension, in: Proceedings of the 2nd Workshop on Machine Reading for Question Answering, 2019, pp. 1–13.

[8] D. Khashabi, S. Min, T. Khot, A. Sabharwal, O. Tafjord, P. Clark, H. Hajishirzi, Unifiedqa: Crossing format boundaries with a single qa system, in: Findings of the Association for Computational Linguistics: EMNLP 2020, 2020, pp. 1896–1907.

[9] D. Jin, S. Gao, J.-Y. Kao, T. Chung, D. Hakkani-tur, Mmm: Multi-stage multi-task learning for multi-choice reading comprehension, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, 2020, pp. 8010–8017.

[10] A. Gottumukkala, D. Dua, S. Singh, M. Gardner, Dynamic sampling strategies for multi-task reading comprehension, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 920–924.

[11] Y. Fan, W. Yang, A backpropagation learning algorithm with graph regularization for feedforward neural networks, Inform. Sci. 607 (2022) 263–277.

[12] D. Li, H. Zhang, Improved regularization and robustness for fine-tuning in neural networks, Adv. Neural Inf. Process. Syst. 34 (2021) 27249–27262.

[13] H. Gouk, T.M. Hospedales, M. Pontil, Distance-based regularisation of deep networks for fine-tuning, in: 9th International Conference on Learning Representations, ICLR, 2021.

[14] C. Wu, F. Wu, T. Qi, Y. Huang, X. Xie, NoisyTune: A little noise can help you finetune pretrained language models better, in: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), 2022, pp. 680–685.

[15] C. Lee, K. Cho, W. Kang, Mixout: Effective regularization to finetune large-scale pretrained language models, 2019, arXiv preprint arXiv:1909.11299.

[16] H. Zhu, Z. Wang, H. Zhang, M. Liu, S. Zhao, B. Qin, Less is more: Domain adaptation with lottery ticket for reading comprehension, in: M.-F. Moens, X. Huang, L. Specia, S.W.-t. Yih (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2021, Association for Computational Linguistics, Punta Cana, Dominican Republic, 2021, pp. 1102–1113, http://dx.doi.org/10.18653/v1/2021.findings-emnlp.95, URL https://aclanthology.org/2021.findings-emnlp.95.

[17] R. Xu, F. Luo, Z. Zhang, C. Tan, B. Chang, S. Huang, F. Huang, Raise a child in large language model: Towards effective and generalizable fine-tuning, in: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, 2021, pp. 9514–9528.

[18] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, S. Gelly, Parameter-efficient transfer learning for NLP, in: International Conference on Machine Learning, 2019, pp. 2790–2799.

[19] S.-A. Rebuffi, H. Bilen, A. Vedaldi, Efficient parametrization of multi-domain deep neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 8119–8127.

[20] M.J. Seo, A. Kembhavi, A. Farhadi, H. Hajishirzi, Bidirectional attention flow for machine comprehension, in: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings, OpenReview.net, 2017, URL https://openreview.net/forum?id=HJ0UKP9ge.

[21] A.W. Yu, D. Dohan, M. Luong, R. Zhao, K. Chen, M. Norouzi, Q.V. Le, QANet: Combining local convolution with global self-attention for reading comprehension, in: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings, OpenReview.net, 2018, URL https://openreview.net/forum?id=B14TlG-RW.

[22] N.L.C. Group, R-NET: Machine reading comprehension with self-matching networks, 2017.

[23] S. Wang, J. Jiang, Machine comprehension using match-LSTM and answer pointer, in: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings, OpenReview.net, 2017, URL https://openreview.net/forum?id=B1-q5Pqxl.

[24] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P.J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, J. Mach. Learn. Res. 21 (2020) 1–67.

[25] M. Yasunaga, J. Leskovec, P. Liang, LinkBERT: Pretraining language models with document links, in: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, 2022, pp. 8003–8016, http://dx.doi.org/10.18653/v1/2022.acl-long.551, URL https://aclanthology.org/2022.acl-long.551.

[26] A. Talmor, J. Berant, MultiQA: An empirical investigation of generalization and transfer in reading comprehension, in: A. Korhonen, D. Traum, L. Màrquez (Eds.), Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 4911–4921, http://dx.doi.org/10.18653/v1/P19-1485, URL https://aclanthology.org/P19-1485.

[27] Y. Xu, X. Liu, Y. Shen, J. Liu, J. Gao, Multi-task learning with sample re-weighting for machine reading comprehension, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, 2019, pp. 2644–2655, http://dx.doi.org/10.18653/v1/N19-1271, URL https://aclanthology.org/N19-1271.

[28] C. Louizos, M. Welling, D.P. Kingma, Learning sparse neural networks through L_0 regularization, in: 6th International Conference on Learning Representations, ICLR, 2018.

[29] E. Jang, S. Gu, B. Poole, Categorical reparameterization with gumbel-softmax, 2016, arXiv preprint arXiv:1611.01144.

[30] C.J. Maddison, A. Mnih, Y.W. Teh, The concrete distribution: A continuous relaxation of discrete random variables, in: 5th International Conference on Learning Representations, ICLR, 2017.

[31] P. Rajpurkar, J. Zhang, K. Lopyrev, P. Liang, SQuAD: 100,000+ questions for machine comprehension of text, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, 2016, pp. 2383–2392.

[32] A. Trischler, T. Wang, X. Yuan, J. Harris, A. Sordoni, P. Bachman, K. Suleman, NewsQA: A machine comprehension dataset, in: Proceedings of the 2nd Workshop on Representation Learning for *NLP*, 2017, pp. 191–200.

[33] M. Joshi, E. Choi, D.S. Weld, L. Zettlemoyer, TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, 2017, pp. 1601–1611.

[34] M. Dunn, L. Sagun, M. Higgins, V.U. Guney, V. Cirik, K. Cho, SearchQA: A new Q&A dataset augmented with context from a search engine, 2017, arXiv preprint arXiv:1704.05179.

[35] Z. Yang, P. Qi, S. Zhang, Y. Bengio, W.W. Cohen, R. Salakhutdinov, C.D. Manning, HotpotQA: A dataset for diverse, explainable multi-hop question answering, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, pp. 2369–2380.

[36] T. Kwiatkowski, J. Palomaki, O. Redfield, M. Collins, A. Parikh, C. Alberti, D. Epstein, I. Polosukhin, J. Devlin, K. Lee, et al., Natural questions: a benchmark for question answering research, Trans. Assoc. Comput. Linguist. 7 (2019) 453–466.

[37] G. Tsatsaronis, G. Balikas, P. Malakasiotis, I. Partalas, M. Zschunke, M.R. Alvers, D. Weissenborn, A. Krithara, S. Petridis, D. Polychronopoulos, et al., An overview of the BIOASQ large-scale biomedical semantic indexing and question answering competition, BMC Bioinformatics 16 (2015) 1–28.

[38] D. Dua, Y. Wang, P. Dasigi, G. Stanovsky, S. Singh, M. Gardner, DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics, 2019, pp. 2368–2378.

[39] A. Saha, R. Aralikatte, M.M. Khapra, K. Sankaranarayanan, DuoRC: Towards complex language understanding with paraphrased reading comprehension, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, 2018, pp. 1683–1693.

[40] G. Lai, Q. Xie, H. Liu, Y. Yang, E. Hovy, RACE: Large-scale reading comprehension dataset from examinations, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017, pp. 785–794.

[41] O. Levy, M. Seo, E. Choi, L. Zettlemoyer, Zero-shot relation extraction via reading comprehension, in: Computational Natural Language Learning, CoNLL, 2017, pp. 333–342.

[42] A. Kembhavi, M. Seo, D. Schwenk, J. Choi, A. Farhadi, H. Hajishirzi, Are you smarter than a sixth grader? textbook question answering for multimodal machine comprehension, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4999–5007.

[43] X. Tao, Q. Li, W. Guo, C. Ren, Q. He, R. Liu, J. Zou, Adaptive weighted over-sampling for imbalanced datasets based on density peaks clustering with heuristic filtering, Inform. Sci. 519 (2020) 43–73.

[44] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., Pytorch: An imperative style, high-performance deep learning library, in: Advances in Neural Information Processing Systems, Vol. 32, 2019, pp. 8026–8037.

[45] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, et al., Transformers: State-of-the-art natural language processing, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, 2020, pp. 38–45.

[46] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, in: 7th International Conference on Learning Representations, ICLR, 2019.