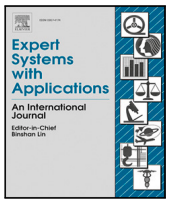
Expert Systems With Applications 281 (2025) 127582

Contents lists available at ScienceDirect

Expert Systems With Applications

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



STAF-LLM: A scalable and task-adaptive fine-tuning framework for large language models in medical domain

Tianhan Xu [a,b], Ling Chen [a,b], Zhe Hu [c], Bin Li [a,b,*]

[a] School of Information Engineering, Yangzhou University, Yangzhou, 225127, Jiangsu, China
[b] Jiangsu Province Engineering Research Center of Knowledge Management and Intelligent Service, Yangzhou, 225127, Jiangsu, China
[c] Department of Computing, The Hong Kong Polytechnic University, 999077, Hong Kong Special Administrative Region of China

ARTICLE INFO

Keywords:
Large language models
Task-adaptive fine-tuning
Knowledge transfer
Scalable
Medical applications

ABSTRACT

Recent large language models (LLMs) have demonstrated remarkable performance across various NLP tasks. However, their application in the medical domain is often limited by a lack of specialized medical knowledge, which is crucial for practical clinical tasks. In this work, we propose STAF-LLM, a scalable and task-adaptive fine-tuning framework designed to customize general-purpose LLMs for diverse downstream medical applications. STAF-LLM consists of two stages: **expert model training** and **task adaptation**. In the first stage, we design 12 core medical tasks and use AdaLoRA to train 12 expert models on these tasks with a unified instruction format, transferring the learned domain-specific knowledge to the general-purpose LLM. In the second stage, a task-guided router is trained for each downstream application to adaptively combine the expert knowledge with the LLM, dynamically selecting the most relevant knowledge for inference. Experiments on 9 medical tasks, including 3 unseen ones, show that STAF-LLM outperforms Llama 2 by 10%–30%. Notably, STAF-LLM achieves state-of-the-art performance on benchmark tasks like ICD coding.

1. Introduction

Recently, large language models (LLMs), such as ChatGPT (OpenAI, 2023), Llama 2 (Touvron, Martin et al., 2023), and PaLM 2 (Anil et al., 2023), have garnered significant attention due to their remarkable performance and strong generalization capabilities across various natural language processing (NLP) tasks, such as question answering (QA), summarization, and natural language inference (NLI) (Qin et al., 2023). LLMs have demonstrated effectiveness in domains like code generation, copywriting, and mathematical problem-solving (Laskar et al., 2023). However, their direct application to medical NLP tasks remains challenging due to insufficient domain-specific knowledge, which is critical for practical clinical applications, such as ICD coding (Mullenbach, Wiegreffe, Duke, Sun, & Eisenstein, 2018), medication recommendation (Jensen, Jensen, & Brunak, 2012), and readmission prediction (Shulan, Gao, & Moore, 2013).

To address these challenges, previous approaches have focused on adapting general-purpose LLMs to the medical domain by continuing training on medical corpora (Peng et al., 2023; Singhal, Azizi et al., 2023) or fine-tuning with medical instruction datasets (Li et al., 2023; Wang et al., 2023; Wornow et al., 2023). However, these methods have several limitations. (1) Continual pre-training or fine-tuning of general-purpose LLMs is resource-intensive and time-consuming (Ding et al., 2023). (2) Differences in task-specific input–output formats (e.g., named entity recognition vs. text classification) and dataset size often create imbalances, particularly for tasks with smaller datasets (Peters, Ruder, & Smith, 2019). (3) Existing methods struggle to extend domain knowledge to new downstream tasks, requiring retraining on the entire domain dataset, which limits their plug-and-play capability.

To overcome these limitations, we introduce a **S**calable and **T**ask-**A**daptive **F**ine-tuning Framework for LLMs in the medical domain (STAF-LLM). STAF-LLM transforms a general-purpose LLM, such as Llama 2 (Touvron, Martin et al., 2023), into a medical domain-specific model. The framework consists of two core stages: **expert model training** and **task adaptation**, as illustrated in Fig. 1.

In the first stage, we design 12 types of data. Inspired by the Mixture of Experts (MoE) framework (Jacobs, Jordan, Nowlan, & Hinton, 1991), we employ 12 expert models, each trained on a specific type of data, to learn domain-specific knowledge using a unified instruction format. The parameters learned by these experts form the foundation of our domain knowledge, which is transferred to the general-purpose LLM via AdaLoRA (Zhang et al., 2023).

In the second stage, a task-specific router is trained for each downstream application to adapt the knowledge learned by the experts. This router utilizes a task-guided routing mechanism to select the

* Corresponding author at: School of Information Engineering, Yangzhou University, Yangzhou, 225127, Jiangsu, China.
E-mail addresses: dx120210092@stu.yzu.edu.cn (T. Xu), lchen@yzu.edu.cn (L. Chen), zhe-derek.hu@connect.polyu.hk (Z. Hu), lb_kmis@yzu.edu.cn (B. Li).

https://doi.org/10.1016/j.eswa.2025.127582
Received 29 January 2024; Received in revised form 17 January 2025; Accepted 2 April 2025
Available online 11 April 2025

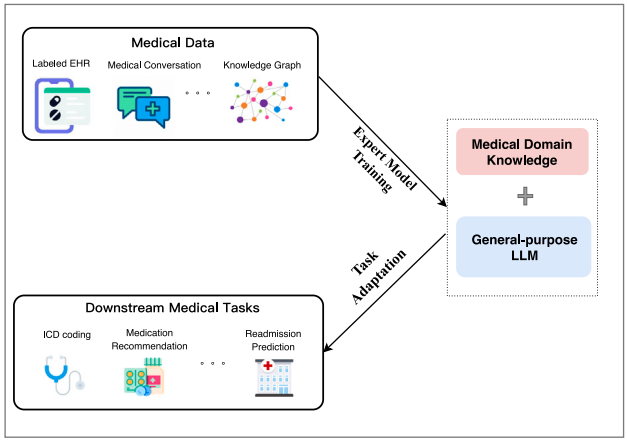


Fig. 1. Overview of STAF-LLM's expert model training and downstream task adaptation.

most relevant expert knowledge for each downstream task. The expert weights calculated by the routers enable dynamic selection of expert knowledge. The task-specific knowledge is then fused with the general-purpose LLM to facilitate efficient inference for diverse medical tasks. Optimization is performed using gradient descent or CMA-ES (Hansen & Ostermeier, 1996).

We evaluate STAF-LLM on 9 downstream medical tasks, including 3 unseen tasks (ICD coding, medication recommendation, and readmission prediction). The results demonstrate that STAF-LLM significantly outperforms Llama 2, with performance improvements ranging from 10% to 30%. STAF-LLM also achieves state-of-the-art performance in both normal and few-shot settings on benchmark tasks, such as ICD coding.

The contributions of this work are as follows:

- We present STAF-LLM, a two-stage scalable and task-adaptive fine-tuning framework that effectively addresses a variety of downstream medical tasks in both normal and few-shot settings.
- The design of 12 core medical tasks and a unified instruction format allows each task to be fine-tuned separately using expert models, enabling the transfer of domain-specific knowledge to the general-purpose LLM.
- A task-guided routing mechanism is proposed to adaptively integrate knowledge from expert models, facilitating the efficient handling of diverse downstream medical applications.
- Experimental results show that STAF-LLM outperforms general-purpose LLMs, particularly on unseen tasks, achieving substantial performance gains across various downstream medical applications.

2. Related work

2.1. BERT-based medical models

Some previous work (Arib et al., 2022; Hu, Chan, & Huang, 2022; Li, xia Liu, Su, & Zhang, 2022) pre-trained token representation on a medical corpus based on BERT (Kenton & Toutanova, 2019), and then fine-tuned downstream task data based on the representation of input tokens. BioBERT (Lee et al., 2020) is continuously trained using PubMed abstracts and PMC full-text articles on top of the generalized corpus pre-training, thus injecting biomedical knowledge at the pre-training stage. SMedBERT (Zhang et al., 2021) simultaneously introduces the medical entities in the knowledge graph, together with the structured semantic information in the entity relationships, into the pre-trained model. G-BERT (Shang, Ma, Xiao, & Sun, 2019) combines GNNs and BERT to learn medical code representations of hierarchies and further integrates the results into pre-trained Transformer-based models. MedM-PLM (Liu et al., 2023) explores the interaction of structured and unstructured data by learning enhanced electronic health records(EHR) representations through pre-training tasks that correlate these two modalities. Yang et al. proposed GatorTron (Yang et al., 2022), a PLM for EHR, and achieved accurate results on five medical NLP tasks.

However, such models have limitations. First, these models are trained on a single type of medical corpus, which affects their contextual understanding and reasoning; second, these models are limited in model scale, and since they are designed based on BERT, their ability of few-shot learning is insufficient, and they have poor performance when confronted with unseen medical tasks.

2.2. LLMs in the medical domain

Recently, generative large language models have shown strong generalization and few-shot learning capabilities in various tasks (Brown et al., 2020). Therefore, some researchers have considered training LLMs on medical corpus. Google and Deepmind introduce prompt tuning based on their multi-category medical datasets, and train the medical LLMs called Med-PaLM 2 (Singhal, Tu et al., 2023) from scratch. GatorTronGPT (Peng et al., 2023), a clinical large language generation model, can be used for biomedical natural language processing, clinical text generation and evaluation. It uses a unified P-tuning (Liu, Ji et al., 2022) base text generation architecture to address biomedical relationship extraction and question answering. PMC-LLaMA (Wu, Zhang, Zhang, Wang, & Xie, 2023) and Huatuo (Wang et al., 2023) are based on LLaMA (Touvron, Lavril et al., 2023) as the original LLM and then fine-tuned using medical papers and knowledge graphs, respectively. DoctorGLM (Xiong et al., 2023) is based on ChatGLM (Zeng et al., 2022) and fine-tuned with Chinese medical dialog data.

However, training the aforementioned medical LLMs requires a large corpus and consumes significant time and memory resources. In addition, updating and extending the corpus of these LLMs is a challenging task.

2.3. Parameter-efficient fine-tuning

Parameter-efficient fine-tuning (PEFT) (Ding et al., 2023) fine-tunes only a small subset of additional model parameters, leaving most LLM parameters fixed. PEFT significantly reduces computational and storage costs and can achieve accuracy comparable to full parameter fine-tuning.

Addition-based: Adapter-tuning (He et al., 2021) introduces small-scale neural network modules (Adapter) between Transformer sub-layers as fine-tuning parameters. Prompt-tuning (Lester, Al-Rfou, & Constant, 2021) and P-tuning (Liu, Ji et al., 2022) perform model fine-tuning with trainable, parameterized prompts.

Specification-based: BitFit (Zaken, Ravfogel, & Goldberg, 2021) achieves parameter reduction by training only the bias-terms and task-specific classification layer in the original model while freezing other parameters.

Reparameterization-based: LoRA (Hu et al., 2021) reduces the number of training parameters through a low-rank matrix representation, enabling efficient fine-tuning of LLMs with a small number of parameters. Its improved variant, QLoRA (Dettmers, Pagnoni, Holtzman, & Zettlemoyer, 2023) achieves approximate computation through a frozen 4-bit quantized PLM.

Our work is based on the reparameterization-based methods. The reason is that, compared with addition-based methods, reparameterization methods do not need to insert additional neural network modules, and have better inference performance and convergence speed; moreover, compared with specification-based methods, reparameterization methods have better performance (Ding et al., 2022).

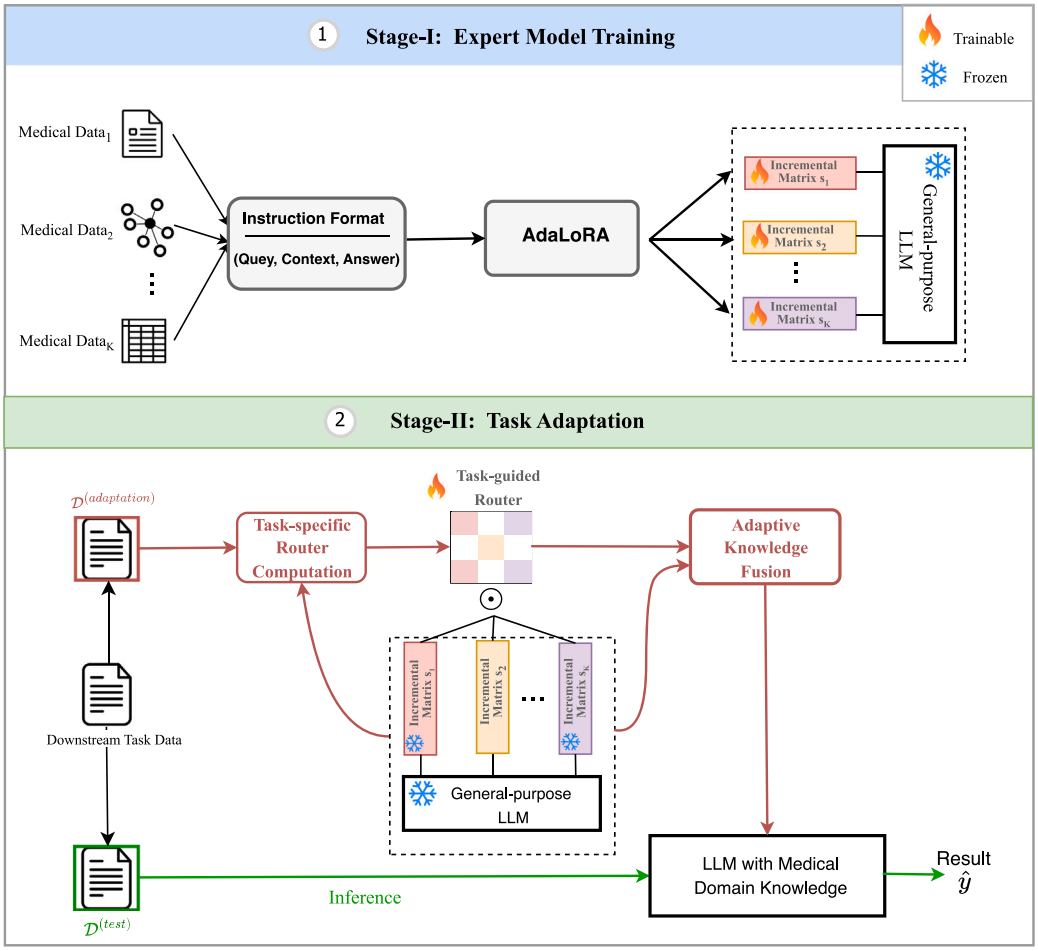


Fig. 2. The framework of STAF-LLM. The red and green lines denote the task adaptation and inference processes, respectively.

2.4. Mixture of Experts

The Mixture-of-Experts (MoE) model, first introduced by Jacobs et al. (1991), utilizes multiple independent networks (experts) to process different subsets of training data, with a gating network directing each input to the most relevant expert, thereby reducing interference and improving learning efficiency and generalization. Shazeer et al. (2017) advanced this concept by introducing sparsely-gated MoE at the token level, where only a small subset of experts is activated for each token, enabling both rapid inference and substantial model scaling. Building upon this, Fedus et al. proposed the Switch Transformer (Fedus, Zoph, & Shazeer, 2022), which efficiently scales MoE models to trillions of parameters by utilizing a simple sparsity mechanism that activates a limited number of experts per token, significantly improving computational efficiency without compromising model performance. Recently, DeepSeekMoE (Dai et al., 2024) improves expert specialization by segmenting experts into smaller subsets and introducing shared experts to capture common knowledge, resulting in significant performance gains with fewer parameters and reduced computational costs, making it a more efficient alternative to traditional MoE architectures.

Our research explores the integration of MoE and LLMs in the medical domain, focusing on leveraging expert knowledge derived from diverse medical data types. By employing an adaptive routing mechanism, we effectively combine experts' insights to address a range of downstream medical tasks. The objective of this work is to develop a domain-specific medical expert model based on a general-purpose LLM.

3. Method

3.1. Overview

The proposed STAF-LLM framework is shown in Fig. 2. It consists of two stages: (1) **Expert model training**, where 12 expert models are trained on specific medical tasks to learn domain-specific knowledge. (2) **Task adaptation**, where the acquired medical knowledge is dynamically integrated with the general-purpose LLM using a task-guided router, enabling efficient inference for downstream medical tasks. This two-stage approach leverages the general capabilities of LLMs while incorporating specialized medical knowledge learned from task-specific data. The following sections provide a detailed description of the method.

3.2. Expert model training

3.2.1. Data construction

In this stage, we design 12 types of data, each derived from different medical tasks, which are essential for the downstream medical tasks. These data types include: Question Answering (QA), Multiple Choice Question Answering (MCQA), Medical Conversation (MC), Multi-Label Document Classification (MLDC), Machine Reading Comprehension (MRC), Natural Language Inference (NLI), Text Summarization (TS), Named Entity Recognition (NER), Relation Extraction (RE), Entity Attribute (EA), Entity Synonymy (ES), and Entity–Entity Relation (ER).

These data types enable the model to effectively address a wide range of downstream tasks, such as identifying sentence similarities

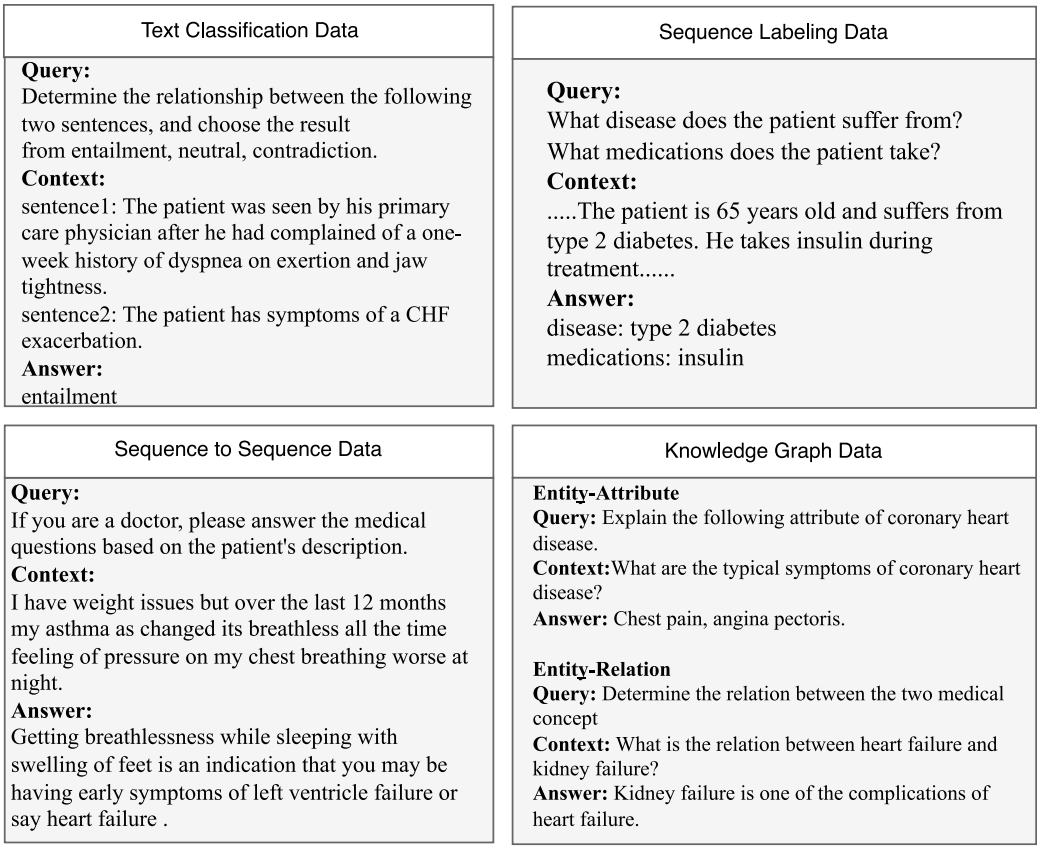


Fig. 3. Examples of converting different categories of medical data into a unified instruction format.

(NLI), assigning labels to EHRs (MLDC), and understanding medical attributes, relationships, and disease characteristics (EA, ER). Specifically, RE is designed to capture causal relationships that are crucial for disease analysis, including etiology, risk factors, and comorbidities.

We represent the entire data as $\mathcal{B} = \{s_1, s_2, \ldots, s_K\}$, where K denotes the number of data types.

3.2.2. Unified instruction format

To standardize the model input for training, we categorize the aforementioned heterogeneous data into five types: question answering, text classification, sequence labeling, sequence-to-sequence, and knowledge graph data. These types are standardized into a **unified instruction format**, which includes a ***context*** and a ***query*** as input, and the ***answer*** to the query as output. Fig. 3 illustrates how these data types are converted into the unified instruction format.

Text Classification Data: For tasks like TC, MLDC, and NLI, we use the original input text as the context and construct a query with all valid labels. The model is trained to predict the start and end positions of the relevant answer in the query.

Sequence Labeling Data: For tasks such as NER, MRC, and causal discovery, we design task-specific templates to map inputs to contexts and queries.

Sequence-to-Sequence Data: For medical conversation and text summarization tasks, the context and answers are treated as input and output sequences, with queries generated based on task-specific templates.

Knowledge Graph Data: The medical knowledge graph contains entity descriptions and relations. Separate queries are generated for each entity or relationship, and the model is trained to output entity descriptions or predict the relationships between entities.

3.2.3. Training methods

We adopt the efficient reparameterization method AdaLoRA (Zhang et al., 2023) to train each of the medical expert models s_i. During the expert model training stage, parameter updates to the LLM are modeled using low-rank decomposition, enabling efficient tuning with minimal incremental updates. Compared to LoRA (Hu et al., 2021), AdaLoRA tunes additional layers within the transformer block and dynamically adjusts the rank of each incremental matrix based on its importance, leading to improved performance.

To parameterize the incremental matrix update $\Delta \in \mathcal{R}^{d_1 \times d_2}$ of the original LLM weight matrix $W^{(0)}$, singular value decomposition (SVD) is applied as follows:

$$W = W^{(0)} + \Delta \approx W^{(0)} + U \Lambda V \tag{1}$$

where $U \in \mathcal{R}^{d_1 \times r}$ and $V \in \mathcal{R}^{r \times d_2}$ are the left and right singular vectors of Δ, and $\Lambda \in \mathcal{R}^{r \times r}$ is the diagonal matrix of singular values. Since $r \ll \min(d_1, d_2)$, this decomposition significantly reduces the number of training parameters compared to full fine-tuning.

Incremental matrix updates are applied to the query, key, and value matrices (W_q, W_k, W_v) in the self-attention block, as well as to the two linear layers in the feedforward network (W_{f_1}, W_{f_2}), and the output projection matrix (W_o) for each transformer layer.

Training is conducted in parallel for all K knowledge types across their respective datasets $\mathcal{D}_i^{(train)}$. The resulting knowledge parameter matrix set is $\Theta = \{\theta_1, \theta_2, \ldots, \theta_K\}$, where each $\theta_i = \{W_q^i, W_k^i, W_v^i, W_{f_1}^i, W_{f_2}^i, W_o^i\}$ for $i \in \{1, \ldots, K\}$. Further details on AdaLoRA can be found in Appendix, and the main training flow is presented in Algorithm 1.

Algorithm 1 Expert Model Training Stage

Input: K medical knowledge datasets $D_1^{(train)}, \ldots, D_K^{(train)}$;
weight matrix of the original LLM $W^{(0)}$;
initial fine-tuning warm-up step t_0, final fine-tuning step t_1.
for $1 \le i \le K$ **do**
Apply incremental matrix parameterization as described in (1).
for $t_0 \le t \le t_1$ **do**
Sample a mini-batch from D_i and update gradients according to (A.3).
Apply eigenvalue gradient trimming as outlined in (A.4).
end for
end for
Output: The trained medical knowledge parameters $\Theta = \{\theta_1, \theta_2, ..., \theta_K\}$, where $\theta_i = \{W_q^i, W_k^i, W_v^i, W_{f_1}^i, W_{f_2}^i, W_o^i\}$.

3.3. Task adaptation

3.3.1. Adaptation for downstream task

In the task adaptation stage, the trained experts are adaptively combined based on different downstream tasks to transfer knowledge in a task-specific manner. For each downstream task, we convert it into the instruction format. Each task $\mathcal{T}$ is represented by the context–target pairs: $(x_n, y_n)_{n=1,\ldots,N}$, where x_n is the input sequence of tokens including the query and context, and y_n is the corresponding answer sequence of tokens, with N denoting the total number of samples.

Formulating the upstream knowledge types and downstream tasks into a unified format has two advantages: (1) it bridges the gap between the two stages in our framework, and (2) the queries generated from the designed task-specific templates serve as semantically rich prompts that better stimulate the potential of the LLM, leading to improved performance.

We divide the downstream task data D into two parts: $D^{(adaptation)}$ for adaptation and $D^{(test)}$ for testing. Inspired by previous works (Du et al., 2022; Masoudnia & Ebrahimpour, 2014), we implement adaptation using the **task-guided router**, and compute the router $\mathcal{R}$ based on the following formulas:

$$E = A \cdot \Theta + \boldsymbol{b} \tag{2}$$

where $E = [\boldsymbol{e_1}, \boldsymbol{e_2}, \ldots, \boldsymbol{e_K}]$ and K is the number of experts. $A = [\boldsymbol{\alpha_1}, \boldsymbol{\alpha_2}, \ldots, \boldsymbol{\alpha_K}]$ represents the weight matrix of medical experts, and $\boldsymbol{b}$ is the bias vector. A and $\boldsymbol{b}$ are trained using $D^{(adaptation)}$.

$$\mathcal{R}(\boldsymbol{e}_i) = \frac{\exp\left(\boldsymbol{e}_i/\tau\right)}{\sum_{j=1}^{K} \exp\left(\boldsymbol{e}_j/\tau\right)} \tag{3}$$

where τ is the temperature coefficient used to smooth the output distribution. Given the parameters of all the above-trained experts $\Theta = \{\theta_1, \theta_2, \ldots, \theta_K\}$ and the frozen parameters Φ_0 of the original LLM, the objective function aims to compute the matrix A and vector $\boldsymbol{b}$ by maximizing the probability of generating the target sequence y. The loss function $\mathcal{L}_{task}$ for task $\mathcal{T}$ is as follows:

$$\Delta\Phi = \sum_{i=1}^{K} \mathcal{R}(\boldsymbol{e}_i) \cdot \theta_i \tag{4}$$

$$\mathcal{L}_{task} = -\sum_{(x,y)\in\mathcal{Z}} \sum_{t=1}^{|y|} \log\left(p_{\Phi_0+\Delta\Phi}(y_t|x, y_{<t})\right) \tag{5}$$

Matrix A and vector $\boldsymbol{b}$ are randomly initialized, and since Θ is trained in the first stage, we only need to iteratively update A and $\boldsymbol{b}$ during the adaptation stage. Our approach is parameter-efficient because $\Delta\Phi \ll \Phi_0$.

To ensure stable training, accurate results, and to avoid overfitting, we categorize each downstream task $\mathcal{T}$ into one of two settings based on its sample size: the **normal settings** and the **few-shot settings**.

For both settings, the total loss function $\mathcal{L}_{total}$ is optimized using the following objective:

$$\mathcal{L}_{total} = \mathcal{L}_{task} + \gamma_1 \mathcal{L}_r \tag{6}$$

$$\mathcal{L}_r = \sum_{i=1}^{K} \|\mathcal{R}(\boldsymbol{e}_i)\|_2^2 \tag{7}$$

where $\mathcal{L}_r$ is the regularization term, and γ_1 is a hyperparameter controlling the trade-off between the task loss and the regularization term, which prevents the model from relying excessively on any single knowledge type and overfitting. Details of task adaptation in STAF-LLM are shown in Fig. 4.

3.3.2. Normal settings

For the normal settings, gradient descent is used to minimize the total loss function $\mathcal{L}_{total}$ in Eq. (6). This approach benefits from the sufficient amount of labeled data available for training, which allows for reliable updates of the knowledge type weights A and bias vector $\boldsymbol{b}$.

3.3.3. Few-shot settings

In the few-shot settings, *Covariance Matrix Adaptive Evolution Strategies (CMA-ES)* (Hansen & Ostermeier, 1996) is used to optimize the weight matrix A and bias vector $\boldsymbol{b}$. Unlike traditional gradient-based methods, CMA-ES is a *gradient-free optimization* technique, making it especially suitable for few-shot learning scenarios where gradient information may be sparse or unreliable due to the limited availability of labeled data. In such cases, gradient-based methods often struggle to produce meaningful updates, leading to overfitting or suboptimal performance.

CMA-ES addresses this issue by evolving a population of candidate solutions, thereby avoiding the need for explicit gradient computation. The optimization process adapts the search strategy based on the population's diversity and the objective function landscape. Specifically, CMA-ES updates the parameters iteratively through the following equations:

$$\mu_{t+1} = \mu_t + \sigma_t \cdot \mathbf{v} \tag{8}$$

$$\Sigma_{t+1} = (1-\rho) \cdot \Sigma_t + \rho \cdot (\mu_{t+1} - \mu_t)(\mu_{t+1} - \mu_t)^T \tag{9}$$

where: μ_t and Σ_t represent the mean vector and covariance matrix of the population at iteration t, σ_t is the step size controlling the width of the search space, $\mathbf{v}$ is the search direction determined by the CMA-ES algorithm, ρ is the learning rate for updating the covariance matrix.

This approach enables CMA-ES to efficiently explore the parameter space, even with limited data, by adapting its search direction based on the evolving covariance matrix. By leveraging the diversity of candidate solutions and updating the search distribution, CMA-ES avoids overfitting and local minima, which are common challenges in few-shot learning. This gradient-free method is particularly effective when gradient signals are weak or unavailable, ensuring stable optimization and robust performance with minimal data.

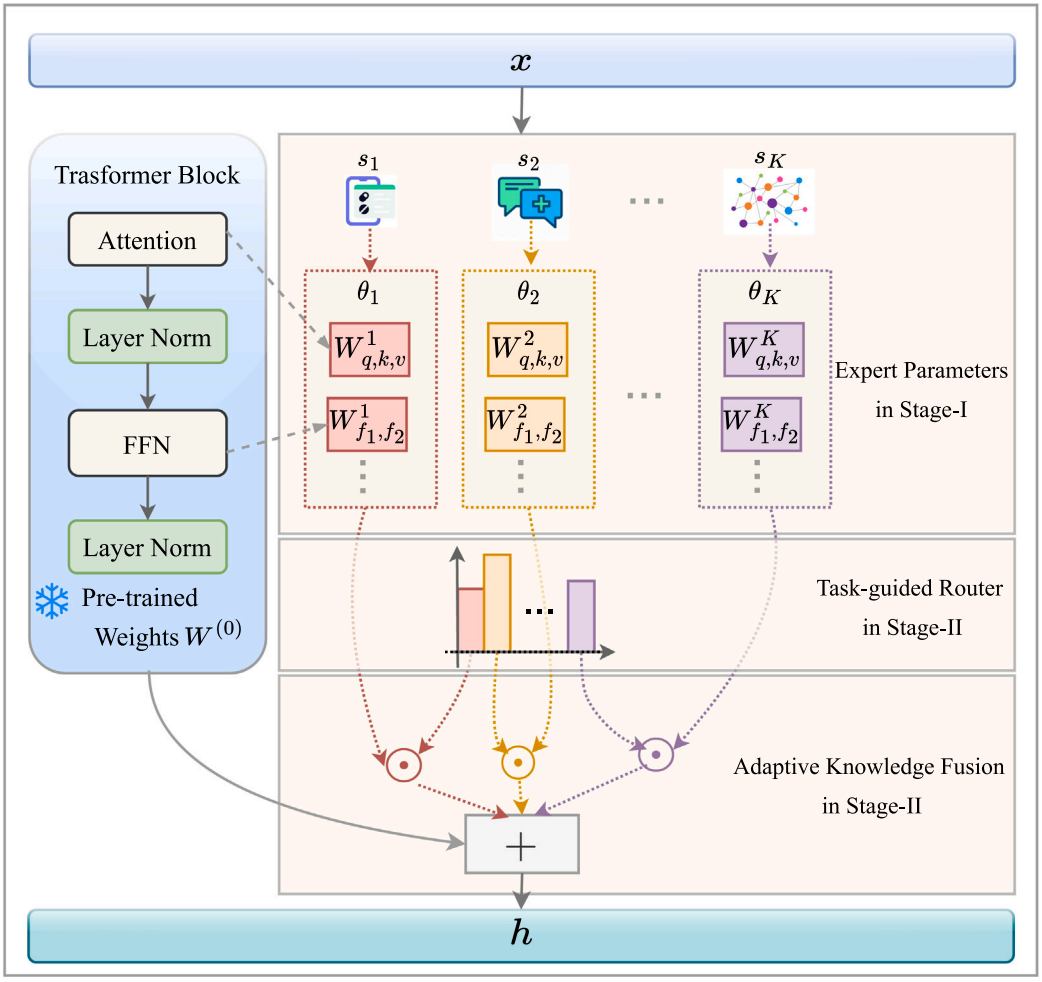


Fig. 4. Details of task adaptation in STAF-LLM.

Algorithm 2 Task Adaptation Stage

Input: Downstream task dataset $\mathcal{D}$;
weight matrix of the general-purpose LLM $W^{(0)}$;
the trained parameters of basic medical experts Θ.
Split $\mathcal{D}$ into two parts: $\mathcal{D}^{(adaptation)}$, $\mathcal{D}^{(test)}$.
Initialize matrix A and vector $\boldsymbol{b}$ in (2) . ▷ Task-specific router computation
if normal settings **then**
Optimize loss function in (6) by gradient descent on $\mathcal{D}^{(adaptation)}$;
else if few-shot settings **then**
Optimize loss function in (6) by CMA-ES on $\mathcal{D}^{(adaptation)}$.
end if
Output: The task-guided router $\mathcal{R}$ in (3).
Fuse the medical expert knowledge with the general-purpose LLM by (10).
Output: LLM with Medical Domain Knowledge.
Inference on $\mathcal{D}^{(test)}$. ▷ Inference
Output: Result $\hat{y}$.

3.3.4. Fusion and inference

Following task adaptation, the output of the task-guided router $\mathcal{R}$ is obtained. The expert knowledge, represented by the weight matrices of the medical experts, is integrated with the general-purpose LLM. The fusion process combines the adaptive expert knowledge with the original LLM weight matrix $W^{(0)}$. Assuming h represents an arbitrary hidden layer within the transformer block, the fusion step is formulated as follows:

$$h = W^{(0)}x + \Delta W x = W^{(0)}x + \sum_{i=1}^{K} \mathcal{R}(e_i) \cdot W_{s_i} x \tag{10}$$

where x represents the input layer. After fusion, the medical domain-enhanced LLM is used to generate predictions $\hat{y}$ on the test dataset $\mathcal{D}^{(test)}$. The details of the task adaptation stage are outlined in Algorithm 2.

4. Experiments

4.1. Datasets

Our proposed framework STAF-LLM, consists of two stages. The datasets used in these two stages are described below.

Table 1
Statistics of the dataset used to train basic medical experts. QA: question and answering, MCQA: multiple choice question answering, MC: medical conversation, MLDC: multi-label document classification, MRC: machine reading comprehension, RE: relation extraction, NLI: natural language inference, TS: text summarization, NER: named entity recognition, EA: entity attribute, ES: entity synonymy, ER: entity relation.

Dataset	Data category	Knowledge type	Corpus size
MedQuAD (Abacha & Demner-Fushman, 2019)	Question answering	MCQA	47,457
USMLE (Jin et al., 2021)	Question answering	MCQA	61,097
HealthCareMagic (Li et al., 2023)	Sequence to sequence	MC	112,165
UMLS (Bodenreider, 2004)	Knowledge graph	EA + ES + ER	15,479
WikiMed (Vashishth, Newman-Griffis, Joshi, Dutt, & Rosé, 2021)	Text classification	MLDC	393,618
CliCR (Šuster & Daelemans, 2018)	Sequence labeling	MRC	10,500
MIMIC-Cause (Khetan et al., 2022)	Sequence labeling	RE	2714
MeQSum (Ben Abacha & Demner-Fushman, 2019)	Sequence to sequence	TS	2333
EMRQA (Pampari, Raghavan, Liang, & Peng, 2018)	Sequence labeling	MRC	5789
PubMedQA (Jin, Dhingra, Liu, Cohen, & Lu, 2019)	Question answering	QA	23,149
MedNLI (Shivade et al., 2019)	Text classification	NLI	1422
CliNER (Text Machine Lab, 2023)	Sequence labeling	NER	1327

Table 2
Statistics of training, adaptation and test datasets in downstream tasks.

Settings	Downstream task	Type	Training	$\text{Adaptation}_{normal}$	$\text{Adaptation}_{few-shot}$	Test
Unseen Data	EMRQA (Pampari et al., 2018)	MRC	4629	580	32	580
	MedQuAD (Abacha & Demner-Fushman, 2019)	MCQA	44 911	1273	32	1273
	PubMedQA (Jin et al., 2019)	QA	21 545	802	32	802
	MIMIC-Cause (Khetan et al., 2022)	RE	1286	714	32	714
	MedNLI (Shivade et al., 2019)	NLI	1138	142	32	142
	CliNER (Text Machine Lab, 2023)	NER	911	208	32	208
Unseen Task	ICD coding (Mullenbach et al., 2018)	MLDC	41 315	1500	32	1584
	Medication recommendation (Jensen et al., 2012)	MLDC	33 225	1200	32	1282
	30-day readmission prediction (Shulan et al., 2013)	DC	5108	580	32	575

4.1.1. Expert model training data

We train 12 expert modules based on 5 categories of data: question answering data, text classification data, sequence labeling data, sequence to sequence data, and knowledge graph data. The statistics of the corpus used to train the medical experts are shown in Table 1.

4.1.2. Downstream tasks

We evaluate our STAF-LLM over 9 downstream tasks, and divide these tasks into two tracks: **unseen data** and **unseen task**, following the work of MP[2] (Sun, He, Zhu, Qiu, & Huang, 2023). The **unseen data** track contains 6 datasets that are used in the first stage to train the medical experts, and we keep a small amount of test data $D^{(test)}$ from the corpus to make sure that the downstream samples are unseen by STAF-LLM. The **unseen task** track consists of 3 new downstream medical tasks that are not used during the knowledge training stage.

As for an unseen task, we conduct test experiments on the MIMIC-III dataset (Johnson et al., 2016), a large, open-access database that represents a real-world dataset. The dataset consists of 58,976 admission records for 49,583 patients treated at Beth Israel Deaconess Medical Center between 2001 and 2012. We use its EHR text for training and testing, and compare the performance of STAF-LLM with the baseline models on three practical clinical tasks.

- **ICD coding** (World Health Organization, 2022) is a multi-label classification task based on EHR text for assigning disease labels to patients (Mullenbach et al., 2018).
- **Medication Recommendation** (Jensen et al., 2012) is a multi-label classification task based on EHR text for automatically recommending medications to patients based on their health conditions
- **30-Day Readmission Prediction** (Shulan et al., 2013) treats patients who are readmitted to the hospital within 30 days of their previous discharge date as positive samples, and it is a binary classification task. This task is of great practical importance in improving the prognosis and quality of patient survival.

Table 2 shows the statistics of data size in downstream tasks. It should be emphasized that for unseen tasks, we do not use the training set to train experts.

4.2. Experimental settings

4.2.1. Mode settings

We follow previous work to evaluate our model with the **normal settings** (Chen, Zhang, & Yang, 2021) and the **few-shot settings** (Schick & Schütze, 2021) of the downstream task data. The normal setting experiment is used to reflect the effect of multiple knowledge sharing and complementarity, while the few-shot setting experiment reflects the model's ability to generalize and transfer learned knowledge to new tasks.

In the normal settings, for each of the 12 expert datasets, the data is divided into three subsets: $D^{(train)}$ (for training the expert model), $D^{(adaptation)}$ (for training the router), and $D^{(test)}$ (for evaluation). A subset of samples is selected from the training set to match the size of the original $D^{(test)}$, and this subset is used as $D^{(adaptation)}$ for router training. This approach ensures that the router is trained on a distribution of samples that closely aligns with the test set, thereby improving task adaptation. In the few-shot settings, following the methodology of Gu, Han, Liu, and Huang (2021), 32 random samples are selected from the training set of the downstream task to construct the adaptation dataset $D^{(adaptation)}$.

4.2.2. Implementation details

We use Llama 2–7B (Touvron, Martin et al., 2023) as our general-purpose model, which was pre-trained on 2 trillion pieces of data from publicly available sources and released by Meta as an open-source LLM. Llama 2 is an auto-regressive language model that uses an optimized transformer architecture.

We set the initial rank and target average rank of the incremental matrix to 12 and 4, respectively. The orthogonal regularization coefficient is set to 0.5. The number of steps for the initial fine-tuning warm-up and the final fine-tuning are 200 and 1000, respectively. Each knowledge matrix has a dropout rate of 0.1. The whole process is trained for 10 epochs on 8 A100 GPUs.

Table 3
Comparative experimental results in normal and few-shot settings on the Unseen Data. Bold font indicates optimal score, underline indicates suboptimal score.

Unseen Data								
Settings	Methods	Tunable Params	EMRQA Acc.	MeDQuAD Acc.	PubMedQA Acc.	MedNLI Acc.	CliNER F1.	MIMIC-Cause Acc.
	Original Llama 2 (Touvron, Martin et al., 2023)	0	51.1	54.7	63.6	65.8	41.4	44.8
Normal settings	Prompt tuning (Lester et al., 2021)	3.8M	60.8	65.5	80.0	76.1	56.6	66.3
	Prefix tuning (Li & Liang, 2021)	1.4M	64.4	78.5	70.2	79.0	59.8	70.1
	P-tuning (Liu, Ji et al., 2022)	6.0M	58.1	76.2	80.4	75.3	52.0	71.0
	LoRA (Hu et al., 2021)	4.2M	65.3	70.7	75.4	79.1	57.9	77.8
	IA3 (Liu, Tam et al., 2022)	7.0M	66.7	74.0	77.5	80.9	64.2	76.9
	STAF-LLM$_{single}$	7.2M	68.3	73.6	74.9	81.4	64.7	78.1
	STAF-LLM$_{average}$	7.3M	71.9	78.0	81.7	83.0	73.4	82.3
	STAF-LLM$_{normal}$	7.3M	78.0	**84.1**	83.0	85.2	**75.2**	**84.9**
	Full fine-tuning	7169M	**78.4**	83.3	**84.2**	**86.5**	74.9	84.0
Few-shot settings	ICL (Xun, Jia, Gopalakrishnan, & Zhang, 2017)	7.5K	53.5	57.1	67.9	68.3	42.1	46.6
	STAF-LLM$_{few-shot}$	10.9K	72.2	79.0	82.0	83.3	73.5	83.2

5. Results and analysis

Tables 3 and 4 present a comparison of STAF-LLM's performance with other fine-tuning methods on 6 unseen data and 3 unseen tasks, respectively. The results we report are the average performance over 5 runs with different random seeds.

5.1. Results of normal settings

In this section, we analyze the effectiveness of our two-stage framework through comparative experiments on the baseline PEFT methods, full fine-tuning, STAF-LLM and its variants.

- **PEFT Baselines**: We use the PEFT baselines (Prompt Tuning, Prefix Tuning, P-tuning, LoRA, I3) for fine-tuning on the training datasets of each individual task, with evaluation performed on the corresponding test datasets.
- **Full Fine-tuning**: This version performs full fine-tuning by using data from all experts and updating all model parameters. In this configuration, the model is trained with the combined knowledge of all experts, allowing us to evaluate the performance when no expert weight adaptation occurs and all parameters are jointly optimized.
- **STAF-LLM**$_{normal}$: In this variant, the expert weights are adaptively learned during the task adaptation stage. The router dynamically adjusts the contribution of each expert based on the downstream task, allowing the model to specialize in task-specific knowledge while leveraging the full range of expert knowledge.
- **STAF-LLM**$_{average}$: This variant uses uniform routing weights, meaning that each expert contributes equally to the task. This setup serves as a baseline, enabling us to compare the performance of adaptive expert routing (in STAF-LLM$_{normal}$) with a uniform contribution from all experts.
- **STAF-LLM**$_{single}$: In this configuration, the expert corresponding to the task is assigned a weight of 1, while all other experts have a weight of 0. This isolates the contribution of the task-specific expert, enabling us to evaluate the performance when only a single expert is used for prediction, without any influence from the others.

Comparison with the original Llama 2. The results of all PEFT methods on the 6 datasets show significant performance improvements over the original Llama 2, demonstrating the importance of medical knowledge for general-purpose LLM.

Comparison with the other PEFT methods. Compared to the 5 main PEFT methods, our STAF-LLM$_{normal}$ method achieves the best results on most of the datasets, further demonstrating the advantage of adaptation for downstream tasks. In particular, STAF-LLM$_{single}$ represents the results of fine-tuning AdaLoRA for a single task. As shown in Table 3, AdaLoRA outperforms LoRA in most tasks. This can be attributed to AdaLoRA's use of more fine-tuned parameters, including both low-rank matrices and task-specific adapters, which allow for better adaptation to the target task. The increased number of tunable parameters enables AdaLoRA to capture more task-specific information, resulting in improved performance.

Comparison with full fine-tuning. For all 9 test datasets in Unseen Data and Unseen Task, STAF-LLM outperforms the full fine-tuning method for 4 of them, and is comparable to it for the other 5 datasets, but with only about 0.1% of the number of parameters, demonstrating the efficiency and performance of our method.

Comparison with other variants. The experimental results for the three variants of STAF-LLM on unseen data, presented in Table 3, reveal significant performance differences. STAF-LLM$_{normal}$ achieves the best results, delivering the highest accuracy across all tasks, including 84.1 on MeDQuAD and 75.2 on CliNER. This highlights the advantage of the task-guided router in enhancing task-specific performance. In contrast, STAF-LLM$_{single}$, which fine-tunes each task independently, does not reach the performance level of STAF-LLM$_{average}$, emphasizing the importance of shared knowledge among the experts.

Comparison with Cross-Domain Variants. STAF-LLM$_{cross-domain}$ is a variant designed to integrate medical data with non-medical data (e.g., NER tasks on non-medical data) in order to explore the impact of cross-domain knowledge on medical task performance. The objective is to investigate whether combining medical and non-medical data can enhance the model's ability to generalize to new medical tasks, particularly those with task characteristics similar to non-medical tasks.

In the experimental setup, we replaced the dataset of the last expert from CliNER with CoNLL-03 (CLiPS Research Group, 2003), a widely used annotated NER dataset derived from an English news corpus. The CoNLL-03 dataset contains entity categories such as people, places, organizations, and miscellaneous items, which are non-medical in nature. This setup allows us to examine how exposure to non-medical data influences the model's performance on medical tasks.

The experimental results show that, when using CliNER as a downstream task, the STAF-LLM$_{cross-domain}$ model achieved an F1 score of 0.665, outperforming Llama 2, which scored 0.414. Additionally, the normalized contribution of the CoNLL-03 expert was 0.05, indicating a small but noticeable impact. These findings demonstrate that STAF-LLM is capable of learning from both medical and non-medical domains, and the cross-domain knowledge contributes to better performance on related medical tasks. This ability to transfer and adapt knowledge across domains is referred to as *skill transfer*, highlighting the model's potential for learning rules not only from medical datasets but also from similar non-medical tasks.

5.2. Results of few-shot settings

Overall Performance. We compare STAF-LLM$_{few-shot}$ with the original Llama 2 and In-Context Learning (ICL) baselines in the few-shot

Table 4
Comparative experimental results in normal and few-shot settings on the Unseen Task. Bold font indicates optimal score, underline indicates suboptimal score.

Unseen Task					
Settings	Methods	Tunable Params	ICD coding AUC.	Medication recommendation F1.	Readmission prediction AUC.
	Original Llama 2 (Touvron, Martin et al., 2023)	0	29.6	37.0	50.4
Normal settings	Prompt tuning (Lester et al., 2021)	5.5M	40.4	58.7	59.0
	Prefix tuning (Li & Liang, 2021)	1.6M	41.5	55.8	63.3
	P-tuning (Liu, Ji et al., 2022)	6.1M	39.6	50.9	60.1
	LoRA (Hu et al., 2021)	4.8M	40.6	54.9	64.3
	IA3 (Liu, Tam et al., 2022)	7.2M	48.9	55.6	67.7
	STAF-LLM$_{average}$	7.8M	57.2	61.3	72.5
	STAF-LLM$_{normal}$	8.0M	<u>61.1</u>	<u>64.2</u>	**76.0**
	Full Fine-tuning	7238M	**62.3**	**65.0**	<u>75.3</u>
Few-shot settings	ICL (Xun et al., 2017)	24.0K	32.1	39.6	56.7
	STAF-LLM$_{few-shot}$	40.4K	58.7	62.8	74.1

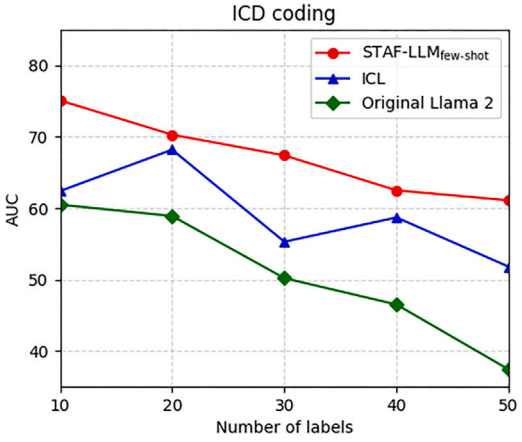


Fig. 5. AUC of the few-shot learning methods comparison on different number of labels in a multi-label classification task (ICD coding).

settings. The experimental data in Tables 3 and 4 show that STAF-LLM$_{few-shot}$ achieves much higher accuracy compared to the original Llama 2, and ICL. Furthermore, by adapting the router with only a small number of samples, STAF-LLM$_{few-shot}$ outperforms STAF-LLM$_{average}$ on 9 downstream tasks, reflecting the importance of using CMA-ES to train the router in the few-shot settings. This further proves the effectiveness of our proposed two-stage framework.

On Multi-Label Classification Tasks. Our proposed method, STAF-LLM, uses a *unified instruction format* in both expert model training stage and task adaptation stage. Thus, it can handle tasks with different numbers of labels. To test the performance of the model on different numbers of labels, we take the ICD coding task as an example. We compare the AUC values of STAF-LLM and the other two baseline models for the 10/20/30/40/50 labels with the highest frequency of occurrence. As can be seen in Fig. 5, there is a sharp drop in the AUC of the ICL when the number of labels is greater than 20. In contrast, the performance of STAF-LLM declines more slowly and steadily as the number of labels increases. This demonstrates the superiority of the *unified instruction format*.

Impact of Sample Size on Router Training Performance. Fig. 6 demonstrates the effect of varying sample sizes on the performance of STAF-LLM$_{few-shot}$ across three tasks: PubMedQA, CliNER, and MedNLI. As the sample size increases from 8 to 32, the AUC scores exhibit substantial improvements for all tasks. However, when increasing the sample size from 32 to 64, the performance gains become marginal, with MedNLI showing a slight decrease. These results suggest that the sample size used for training the router in STAF-LLM$few-shot$ has a significant impact on model performance. Based on these findings, we select 32 samples for the few-shot setting and employ CMA-ES to train the router for optimal task performance.

Token Consumption. In our approach, STAF-LLM$_{few-shot}$ uses small samples to train the router, which then combines expert parameters based on CMA-ES. This method significantly reduces token consumption compared to ICL. In ICL, all examples must be explicitly included in the prompt, which increases token consumption, particularly as the number of examples grows. For instance, if each example requires 150 tokens (input + output), 32 examples would consume 4800 tokens, which exceeds the input limit of models like Llama 2 (4096 tokens), requiring truncation or the use of fewer examples. In contrast, STAF-LLM does not require placing all examples into the prompt. Instead, it efficiently trains the router with a small subset of samples, leading to a much more efficient token usage. This approach allows STAF-LLM to scale more effectively with larger task sizes, while keeping token consumption well within model limits, offering a significant advantage in terms of efficiency and scalability.

5.3. Comparison with downstream baseline models

To ensure fairness, we use the data from MIMIC-III to train a new knowledge and plug it into Llama 2 to get a new version of STAF-LLM, dubbed STAF-LLM$_{new}$. Then, we compare the performance of STAF-LLM$_{new}$ with the baseline models on the test data using the three clinical tasks mentioned above: ICD coding, medication recommendation, and readmission prediction.

From the results demonstrated in Table 5, it can be seen that the AUC and F1 scores of STAF-LLM$_{new}$ on all three clinical tasks exceeded the corresponding scores of the Bert-based baseline. On the three downstream tasks, STAF-LLM$_{new}$ outperforms STAF-LLM$_{normal}$, demonstrating that incorporating downstream task data into expert training can significantly enhance performance. This improvement occurs because the model is able to acquire more task-specific knowledge. Additionally, compared to the strong baseline Meditron 7B (Chen et al., 2023), a LLM pretrained on various medical corpus, STAF-LLM$_{new}$ shows competitive performance, highlighting the effectiveness of our proposed STAF-LLM framework.

5.4. Analysis of expert weights

Fig. 7 visualizes the expert weights across three downstream tasks. Each bar represents the contribution of a corresponding expert, with the bar lengths normalized to 1, ensuring consistency in comparison. The distribution of expert weights varies significantly across tasks, reflecting the specialized knowledge each expert brings to different aspects of medical knowledge.

For the question-answering task PubMedQA, experts MedQuAD and PubMedQA contribute notably, as their knowledge aligns closely with the task. In contrast, for the sequence labeling task CliNER, in addition

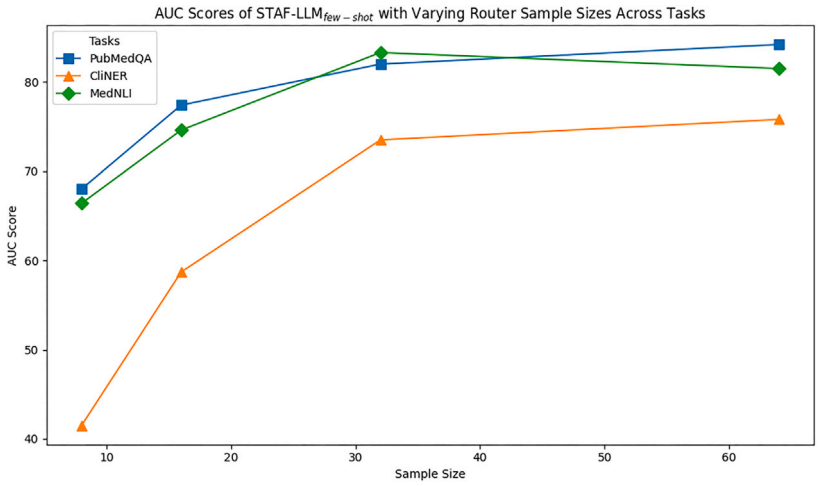


Fig. 6. The figure shows the AUC scores of the STAF-LLM$_{few-shot}$ model's router trained with different sample sizes (8, 16, 32, 64) for three tasks: PubMedQA, CliNER, and MedNLI.

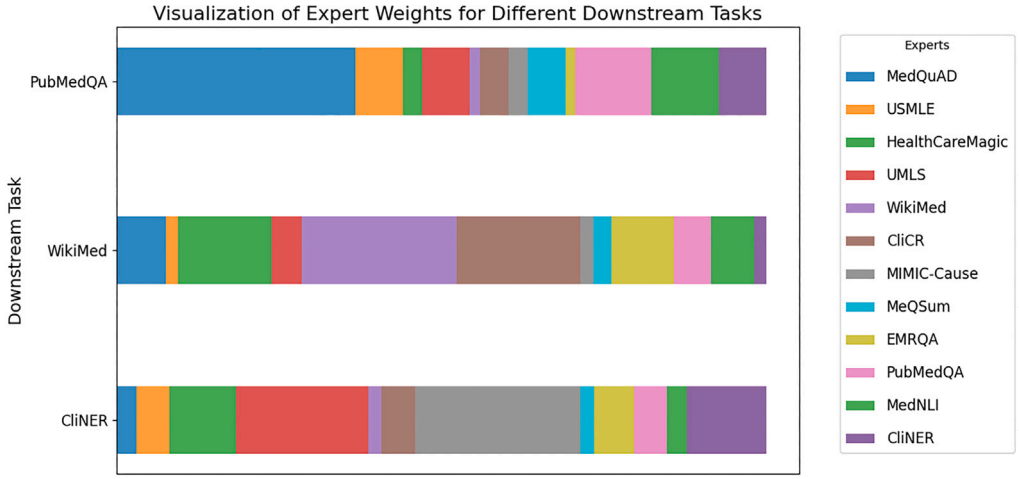


Fig. 7. Visualization of expert weights across three downstream medical tasks. In each task, the bar length, represented by different colors, corresponds to the weight assigned to each expert.

Table 5
Performance comparison of STAF-LLM and baseline models on three clinical tasks. Results are averaged over five runs.

Task	Model	Accuracy	AUC	F1
ICD coding (Mullenbach et al., 2018)	TextCNN (Kim, 2014)	67.9	63.7	66.3
	ClinicalBERT (Huang, Altosaar, & Ranganath, 2019)	75.5	74.6	75.6
	Meditron 7B (Chen et al., 2023)	80.3	**80.7**	**81.2**
	STAF-LLM$_{normal}$	72.7	62.3	68.5
	STAF-LLM$_{new}$	**82.5**	78.9	79.3
Medication recommendation (Jensen et al., 2012)	TextCNN (Kim, 2014)	69.3	61.8	58.1
	ClinicalBERT (Huang et al., 2019)	77.3	75.5	66.4
	Meditron 7B (Chen et al., 2023)	**83.4**	**80.3**	**79.1**
	STAF-LLM$_{normal}$	71.2	68.4	64.2
	STAF-LLM$_{new}$	81.0	79.6	77.8
Readmission prediction (Shulan et al., 2013)	TextCNN (Kim, 2014)	59.4	57.1	50.4
	ClinicalBERT (Huang et al., 2019)	78.5	77.2	70.8
	Meditron 7B (Chen et al., 2023)	81.7	**82.9**	70.4
	STAF-LLM$_{normal}$	73.3	76.0	64.5
	STAF-LLM$_{new}$	**84.8**	81.3	**72.5**

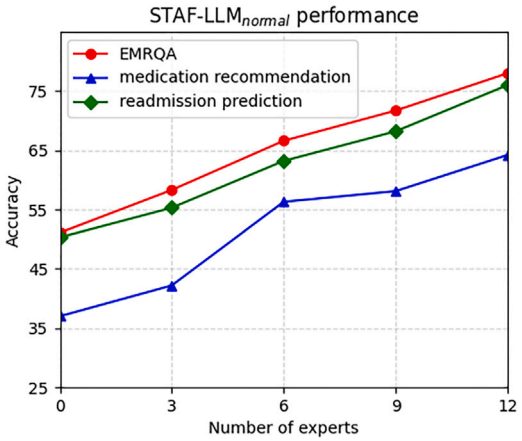


Fig. 8. Effect of expert number on STAF-LLM performance.

to the CliNER expert, the UMLS and MIMIC-Cause experts also make substantial contributions. This is due to their knowledge of entities and relationships, which closely match the requirements of the task. These observations suggest that experts whose knowledge more closely matches the downstream tasks tend to be assigned higher weights.

This analysis underscores STAF-LLM's ability to effectively allocate expert knowledge based on task-specific needs, demonstrating its adaptability in optimizing performance for diverse medical tasks.

5.5. Effect of expert number on performance

We selected three tasks, EMRQA, medication recommendation, and readmission prediction, to observe the effect of different numbers of experts on the performance of our STAF-LLM by adjusting the number of experts involved in the training.

Fig. 8 presents the experimental results, which demonstrate that STAF-LLM$_{normal}$ consistently improves its performance as the number of experts increases. Notably, even with just 3 experts, STAF-LLM$_{normal}$ significantly outperforms the original Llama 2, effectively validating the critical role of expert knowledge in medical downstream tasks.

6. Conclusion

In this work, we propose a **S**calable and **T**ask-**A**daptive **F**ine-tuning Framework for LLMs in the medical domain (STAF-LLM). The framework consists of two core stages: **expert model training** and **task adaptation**. In Stage I, we design 12 medical tasks and use AdaLoRA to train 12 experts. In Stage II, a task-guided router is trained for each downstream application to adaptively combine the expert knowledge with the general-purpose LLM, dynamically selecting the most relevant knowledge for inference.

Extensive experiments on 9 medical tasks, including 3 previously unseen tasks, demonstrate that STAF-LLM significantly outperforms Llama 2, with performance improvements ranging from 10% to 30%. STAF-LLM also achieves state-of-the-art performance on benchmark tasks like ICD coding. Furthermore, STAF-LLM exhibits strong generalization capabilities, performing well in both normal and few-shot settings. Our framework not only enhances medical NLP tasks but also has the potential for application in other domains requiring specialized knowledge integration.

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.

Acknowledgments

This paper is supported by the National Natural Science Foundation of China under Grant No. 61972335.

Appendix. Details of AdaLoRA

We use j to index the incremental matrix, i.e., $\Delta_j = U_j \Lambda_j V_j$ where $j \in \Gamma = \{q, k, v, f_1, f_2, o\}$. We further donate the parameter sets $\mathcal{U} = \{U_j\}_{j\in\Gamma}, \mathcal{E} = \{\Lambda_j\}_{j\in\Gamma}, \mathcal{V} = \{V_j\}_{j\in\Gamma}$. Then the final loss function of a specific knowledge s_i can be described as follows:

$$\mathcal{L}(\mathcal{U}, \mathcal{E}, \mathcal{V}) = C(\mathcal{U}, \mathcal{E}, \mathcal{V}) + \gamma \sum_{j\in\Gamma} R(U_j, V_j) \tag{A.1}$$

$$R(U, V) = \|U^T U - I\|_F^2 + \|V^T V - I\|_F^2 \tag{A.2}$$

where $C(\mathcal{U}, \mathcal{E}, \mathcal{V})$ denotes the loss function on the training data, $R(U, V)$ is the regularizer to enforce the orthogonality of U and V, γ is the hyper-parameter.

In order to control the budget of the fine-tunable parameters, the trainable parameters are dynamically assigned during the training process. At the tth step, where t is between the initial fine-tuning warm-up step t_0 and the final fine-tuning step t_1, we take a stochastic gradient step to update $U_j^{(t)}$, $\Lambda_j^{(t)}$, $Q_j^{(t)}$ for $j \in \Gamma$. Specifically, for $\Lambda_j^{(t)}$:

$$\hat{\Lambda}_j^{(t)} = \Lambda_j^{(t)} - \eta \nabla_{\Lambda_j} \mathcal{L}(\mathcal{U}^{(t)}, \mathcal{E}^{(t)}, \mathcal{V}^{(t)}) \tag{A.3}$$

where η denotes the learning rate. We further donate $\mathcal{G}_p = \{U_{*p}, \Lambda_p, V_{p*}\}$ as the triplet containing the pth singular value and vectors. The gradient is then trimmed according to the importance of each triplet to obtain $\hat{\Lambda}_j^{(t+1)}$, retaining only the singular values whose importance satisfies the requirement. Given the importance score $S_j^{(t)}$, the singular values are pruned as follows:

$$\hat{\Lambda}_j^{(t+1)} = \mathcal{F}(\hat{\Lambda}_j^{(t)}, S_j^{(t)}) \tag{A.4}$$

$$\mathcal{F}(\hat{\Lambda}_j^{(t)}, S_j^{(t)})_{pp} = \begin{cases} \hat{\Lambda}_{j,pp}^{(t)} & S_{j,p}^{(t)} \text{ is in the top-}b^{(t)} \text{ of } S^{(t)}, \\ 0 & \text{otherwise,} \end{cases} \tag{A.5}$$

where $S^{(t)} = \{S_{j,p}^{(t)}\}_{1\le j\le m, 1\le p\le r}$ contains the importance scores of all triplets, $b^{(t)}$ is the budget of remaining singular values at the tth step. The importance of a particular triplet is computed as follows:

$$S_{j,p} = s(\lambda_{j,p}) + \frac{1}{d_1}\sum_{q=1}^{d_1} s(U_{j,qp}) + \frac{1}{d_2}\sum_{q=1}^{d_2} s(V_{j,pq}) \tag{A.6}$$

where $S_{j,p}$ denotes the importance score of the pth triple of the jth weight matrix, $\lambda_{j,p}$ denotes the pth element of the Λ matrix of the jth weight matrix, $U_{j,qp}$ denotes the (q, p) element of the U matrix of the jth weight matrix, and $V_{j,pq}$ denotes the (p, q) element of the V matrix of the jth weight matrix. $s(.)$ denotes the importance of a parameter, which is defined as the magnitude of any trainable parameter w_{pq} and its gradient $\nabla_{w_{pq}}\mathcal{L}$: $s(w_{pq}) = |w_{pq}\nabla_{w_{pq}}\mathcal{L}|$. This formula approximates the change in the loss function when the parameter becomes zero, meaning that if a parameter is cropped and the loss function changes significantly, we should keep it. For more detailed explanation of the formulas, please refer to the AdaLoRA (Zhang et al., 2023) paper.

Data availability

Data will be made available on request.

References

Abacha, A. B., & Demner-Fushman, D. (2019). A question-entailment approach to question answering. *BMC Bioinformatics*, *20*, URL https://api.semanticscholar.org/CorpusID:59222825.

Anil, R., Dai, A. M., Firat, O., Johnson, M., Lepikhin, D., Passos, A., et al. (2023). Palm 2 technical report. arXiv preprint arXiv:2305.10403.

Aribandi, V., Tay, Y., Schuster, T., Rao, J., Zheng, H. S., Mehta, S. V., et al. (2022). ExT5: Towards extreme multi-task scaling for transfer learning. In *International conference on learning representations*. URL https://openreview.net/forum?id=Vzh1BFUCiIX.

Ben Abacha, A., & Demner-Fushman, D. (2019). On the summarization of consumer health questions. In *Proceedings of the 57th annual meeting of the association for computational linguistics* (pp. 2228–2234). Florence, Italy: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/P19-1215, URL https://aclanthology.org/P19-1215.

Bodenreider, O. (2004). The unified medical language system (UMLS): integrating biomedical terminology. *Nucleic Acids Research*, *32*(suppl_1), D267–D270.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., et al. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, *33*, 1877–1901.

Chen, Z., Cano, A. H., Romanou, A., Bonnet, A., Matoba, K., Salvi, F., et al. (2023). Meditron-70b: Scaling medical pretraining for large language models. arXiv preprint arXiv:2311.16079.

Chen, S., Zhang, Y., & Yang, Q. (2021). Multi-task learning in natural language processing: An overview. arXiv preprint arXiv:2109.09138.

CLiPS Research Group (2003). Conll-2003 shared task: Language-independent named entity recognition. URL https://www.clips.uantwerpen.be/conll2003/ner/. (Accessed 10 January 2025).

Dai, D., Deng, C., Zhao, C., Xu, R., Gao, H., Chen, D., et al. (2024). Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. arXiv preprint arXiv:2401.06066.

Dettmers, T., Pagnoni, A., Holtzman, A., & Zettlemoyer, L. (2023). Qlora: Efficient finetuning of quantized llms. arXiv preprint arXiv:2305.14314.

Ding, N., Qin, Y., Yang, G., Wei, F., Yang, Z., Su, Y., et al. (2022). Delta tuning: A comprehensive study of parameter efficient methods for pre-trained language models. arXiv preprint arXiv:2203.06904.

Ding, N., Qin, Y., Yang, G., Wei, F., Yang, Z., Su, Y., et al. (2023). Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence*, *5*(3), 220–235.

Du, N., Huang, Y., Dai, A. M., Tong, S., Lepikhin, D., Xu, Y., et al. (2022). GLaM: Efficient scaling of language models with mixture-of-experts. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, S. Sabato (Eds.), *Proceedings of machine learning research*: *vol. 162*, *Proceedings of the 39th international conference on machine learning* (pp. 5547–5569). PMLR, URL https://proceedings.mlr.press/v162/du22c.html.

Fedus, W., Zoph, B., & Shazeer, N. (2022). Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, *23*(120), 1–39.

Gu, Y., Han, X., Liu, Z., & Huang, M. (2021). Ppt: Pre-trained prompt tuning for few-shot learning. arXiv preprint arXiv:2109.04332.

Hansen, N., & Ostermeier, A. (1996). Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. In *Proceedings of IEEE international conference on evolutionary computation* (pp. 312–317). IEEE.

He, R., Liu, L., Ye, H., Tan, Q., Ding, B., Cheng, L., et al. (2021). On the effectiveness of adapter-based tuning for pretrained language model adaptation. ArXiv abs/2106.03164. URL https://api.semanticscholar.org/CorpusID:235359141.

Hu, Z., Chan, H. P., & Huang, L. (2022). MOCHA: A multi-task training approach for coherent text generation from cognitive perspective. In Y. Goldberg, Z. Kozareva, & Y. Zhang (Eds.), *Proceedings of the 2022 conference on empirical methods in natural language processing* (pp. 10324–10334). Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/2022.emnlp-main.705, URL https://aclanthology.org/2022.emnlp-main.705.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., et al. (2021). Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685.

Huang, K., Altosaar, J., & Ranganath, R. (2019). Clinicalbert: Modeling clinical notes and predicting hospital readmission. arXiv preprint arXiv:1904.05342.

Jacobs, R. A., Jordan, M. I., Nowlan, S. J., & Hinton, G. E. (1991). Adaptive mixtures of local experts. *Neural Computation*, *3*(1), 79–87.

Jensen, P. B., Jensen, L. J., & Brunak, S. (2012). Mining electronic health records: towards better research applications and clinical care. *Nature Reviews. Genetics*, *13*(6), 395–405.

Jin, Q., Dhingra, B., Liu, Z., Cohen, W. W., & Lu, X. (2019). Pubmedqa: A dataset for biomedical research question answering. arXiv preprint arXiv:1909.06146.

Jin, D., Pan, E., Oufattole, N., Weng, W.-H., Fang, H., & Szolovits, P. (2021). What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, *11*(14), 6421.

Johnson, A. E., Pollard, T. J., Shen, L., Lehman, L.-w. H., Feng, M., Ghassemi, M., et al. (2016). MIMIC-III, a freely accessible critical care database. *Scientific Data*, *3*(1), 1–9.

Kenton, J. D. M.-W. C., & Toutanova, L. K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. *vol. 1*, In *Proceedings of naacL-HLT* (p. 2).

Khetan, V., Rizvi, M. I., Huber, J., Bartusiak, P., Sacaleanu, B., & Fano, A. (2022). MIMICause: Representation and automatic extraction of causal relation types from clinical notes. In *Findings of the association for computational linguistics: ACL 2022* (pp. 764–773). Dublin, Ireland: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/2022.findings-acl.63, URL https://aclanthology.org/2022.findings-acl.63.

Kim, Y. (2014). Convolutional neural networks for sentence classification. In *Proceedings of the 2014 conference on empirical methods in natural language processing* (pp. 1746–1751). Association for Computational Linguistics.

Laskar, M. T. R., Bari, M. S., Rahman, M., Bhuiyan, M. A. H., Joty, S. R., & Huang, J. (2023). A systematic study and comprehensive evaluation of ChatGPT on benchmark datasets. In *Annual meeting of the association for computational linguistics*. URL https://api.semanticscholar.org/CorpusID:258967462.

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., et al. (2020). Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, *36*(4), 1234–1240.

Lester, B., Al-Rfou, R., & Constant, N. (2021). The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 conference on empirical methods in natural language processing* (pp. 3045–3059). Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/2021.emnlp-main.243, URL https://aclanthology.org/2021.emnlp-main.243.

Li, Y., Li, Z., Zhang, K., Dan, R., Jiang, S., & Zhang, Y. (2023). ChatDoctor: A medical chat model fine-tuned on a large language model meta-AI (LLaMA) using medical domain knowledge. *Cureus*, *15*(6).

Li, X. L., & Liang, P. (2021). Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: long papers)* (pp. 4582–4597). Online: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/2021.acl-long.353, URL https://aclanthology.org/2021.acl-long.353.

Li, J., xia Liu, R., Su, L., & Zhang, S. (2022). Chinese electronic medical record named entity recognition model based on pre-training and multi-task learning. In *Other conferences*. URL https://api.semanticscholar.org/CorpusID:251227629.

Liu, X., Ji, K., Fu, Y., Tam, W. L., Du, Z., Yang, Z., et al. (2022). P-Tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Annual meeting of the association for computational linguistics*. URL https://api.semanticscholar.org/CorpusID:248780177.

Liu, H., Tam, D., Muqeeth, M., Mohta, J., Huang, T., Bansal, M., et al. (2022). Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems*, *35*, 1950–1965.

Liu, S., Wang, X., Hou, Y., Li, G., Wang, H., Xu, H., et al. (2023). Multimodal data matters: Language model pre-training over structured and unstructured electronic health records. *IEEE Journal of Biomedical and Health Informatics*, *27*(1), 504–514. http://dx.doi.org/10.1109/JBHI.2022.3217810.

Masoudnia, S., & Ebrahimpour, R. (2014). Mixture of experts: a literature survey. *Artificial Intelligence Review*, *42*, 275–293.

Mullenbach, J., Wiegreffe, S., Duke, J., Sun, J., & Eisenstein, J. (2018). Explainable prediction of medical codes from clinical text. In *Proceedings of the 2018 conference of the North American chapter of the association for computational linguistics: human language technologies (NAACL-HLT 2018)* (pp. 1101–1111). Association for Computational Linguistics.

OpenAI (2023). ChatGPT (feb 13 version) [large language model]. URL https://chat.openai.com.

Pampari, A., Raghavan, P., Liang, J., & Peng, J. (2018). Emrqa: A large corpus for question answering on electronic medical records. arXiv preprint arXiv:1809.00732.

Peng, C., Yang, X., Chen, A., Smith, K. E., Pournejatian, N. M., Costa, A. B., et al. (2023). A study of generative large language model for medical research and healthcare. ArXiv abs/2305.13523. URL https://api.semanticscholar.org/CorpusID:258841310.

Peters, M. E., Ruder, S., & Smith, N. A. (2019). To tune or not to tune? Adapting pretrained representations to diverse tasks. *ACL 2019*, 7.

Qin, C., Zhang, A., Zhang, Z., Chen, J., Yasunaga, M., & Yang, D. (2023). Is ChatGPT a general-purpose natural language processing task solver?. ArXiv abs/2302.06476. URL https://api.semanticscholar.org/CorpusID:256827430.

Schick, T., & Schütze, H. (2021). Few-shot text generation with natural language instructions. In *Proceedings of the 2021 conference on empirical methods in natural language processing* (pp. 390–402).

Shang, J., Ma, T., Xiao, C., & Sun, J. (2019). Pre-training of graph augmented transformers for medication recommendation. In *Proceedings of the twenty-eighth international joint conference on artificial intelligence* (pp. 5953–5959). International Joint Conferences on Artificial Intelligence Organization, http://dx.doi.org/10.24963/ijcai.2019/825.

Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A. F., Le, Q. V., Popov, I., et al. (2017). Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. arXiv preprint arXiv:1701.06538.

Shivade, C., et al. (2019). Mednli-a natural language inference dataset for the clinical domain. In *Proceedings of the 2018 conference on empirical methods in natural language processing, Brussels, Belgium. association for computational linguistics* (pp. 1586–1596).

Shulan, M., Gao, K., & Moore, C. D. (2013). Predicting 30-day all-cause hospital readmissions. *Health Care Management Science*, *16*, 167–175.

Singhal, K., Azizi, S., Tu, T., Mahdavi, S. S., Wei, J., Chung, H. W., et al. (2023). Large language models encode clinical knowledge. *Nature*, *620*(7972), 172–180.

Singhal, K., Tu, T., Gottweis, J., Sayres, R., Wulczyn, E., Hou, L., et al. (2023). Towards expert-level medical question answering with large language models. arXiv preprint arXiv:2305.09617.

Sun, T., He, Z., Zhu, Q., Qiu, X., & Huang, X.-J. (2023). Multitask pre-training of modular prompt for chinese few-shot learning. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)* (pp. 11156–11172).

Šuster, S., & Daelemans, W. (2018). CliCR: a dataset of clinical case reports for machine reading comprehension. In *Proceedings of the 2018 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long papers)* (pp. 1551–1563). New Orleans, Louisiana: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/N18-1140, URL https://aclanthology.org/N18-1140.

Text Machine Lab (2023). CliNER. https://github.com/text-machine-lab/CliNER.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., et al. (2023). Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., et al. (2023). Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288.

Vashishth, S., Newman-Griffis, D., Joshi, R., Dutt, R., & Rosé, C. P. (2021). WikiMed and PubMedDS: Two large-scale datasets for medical concept extraction and normalization research. *Journal of Biomedical Informatics*, *121*, Article 103880. http://dx.doi.org/10.5281/zenodo.5755155.

Wang, H., Liu, C., Xi, N., Qiang, Z., Zhao, S., Qin, B., et al. (2023). Huatuo: Tuning llama model with chinese medical knowledge. arXiv preprint arXiv:2304.06975.

World Health Organization (2022). International classification of diseases. URL https://www.who.int/standards/classifications/classification-of-diseases. (Accessed 02 June 2023).

Wornow, M., Xu, Y., Thapa, R., Patel, B., Steinberg, E., Fleming, S., et al. (2023). The shaky foundations of large language models and foundation models for electronic health records. *Npj Digital Medicine*, *6*(1), 135.

Wu, C., Zhang, X., Zhang, Y., Wang, Y., & Xie, W. (2023). Pmc-llama: Further finetuning llama on medical papers. arXiv preprint arXiv:2304.14454.

Xiong, H., Wang, S., Zhu, Y., Zhao, Z., Liu, Y., Wang, Q., et al. (2023). Doctorglm: Fine-tuning your chinese doctor is not a herculean task. arXiv preprint arXiv:2304.01097.

Xun, G., Jia, X., Gopalakrishnan, V., & Zhang, A. (2017). A survey on context learning. *IEEE Transactions on Knowledge and Data Engineering*, *29*(1), 38–56. http://dx.doi.org/10.1109/TKDE.2016.2614508.

Yang, X., Chen, A., PourNejatian, N., Shin, H. C., Smith, K. E., Parisien, C., et al. (2022). A large language model for electronic health records. *NPJ Digital Medicine*, *5*(1), 194.

Zaken, E. B., Ravfogel, S., & Goldberg, Y. (2021). Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. arXiv preprint arXiv:2106.10199.

Zeng, A., Liu, X., Du, Z., Wang, Z., Lai, H., Ding, M., et al. (2022). Glm-130b: An open bilingual pre-trained model. arXiv preprint arXiv:2210.02414.

Zhang, T., Cai, Z., Wang, C., Qiu, M., Yang, B., & He, X. (2021). SMedBERT: A knowledge-enhanced pre-trained language model with structured semantics for medical text mining. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: long papers)* (pp. 5882–5893). Online: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/2021.acl-long.457, URL https://aclanthology.org/2021.acl-long.457.

Zhang, Q., Chen, M., Bukharin, A. W., He, P., Cheng, Y., Chen, W., et al. (2023). Adaptive budget allocation for parameter-efficient fine-tuning. ArXiv abs/2303.10512. URL https://api.semanticscholar.org/CorpusID:257631760.