JIUTIAN-139MoE: TECHNICAL REPORT

JiuTian TEAM

Abstract

We report the development of JIUTIAN-139MoE, a 13-billion active parameter language model designed to be an efficient foundation model for industrial use. It adopts a decoder-only Transformer-based Mixture-of-Experts (MoE) architecture, employing a pair of large twin experts and six small experts to capture the intelligence associated with diverse industries. In terms of training, we support training with clusters of various GPUs and NPUs. We also support lossless switch between two heterogeneous clusters. In addition, JIUTIAN-139MoE-Chat, a fine-tuned version of JIUTIAN-139MoE, surpasses state-of-the-art large language models on both open and self-built industry-standard benchmarks. Specifically, it exhibits outstanding performances on 10 industrial benchmarks and leading performances on 10 benchmarks of general knowledge understanding, reasoning, math and coding capabilities. JIUTIAN-139MoE is released under the Apache 2.0 license and JIUTIAN Large Model Community License Agreement. It is publicly available at `https://jiutian.10086.cn/qdlake/qdh-web/#/model/detail/1070`.

1 Introduction

In this paper, we present our first open-source model of the JIUTIAN Model Series, namely JIUTIAN-139MoE. One of the main goals of developing such models is to improve their ability in real-world uses in industries. Large Language Models (LLMs) today have achieved revolutionary advancements in a wide range of benchmarks and real-world applications. However, great challenges remain when deploying in the critical processes of businesses and industries. One of the main challenges is that the hallucination rate rises significantly when facing a complex professional question [Huang et al., 2023, Ye et al., 2023, Tonmoy et al., 2024]. The reasons are multi-fold: (1) Data from a given industry or a professional field are often not well represented in the training data of big models, especially compared to the typical popular content on the internet such as social media. (2) The architectures and learning mechanisms of Transformer-based models are designed to learn like ordinary people instead of professionals. We approach these challenges by increasing professional data from various industries, resampling data as well as proposing and experimenting with a special Mixture-of-Experts architecture [Fedus et al., 2022a], where experts are designed of different sizes to vaguely match the various logic complexity and knowledge mass. Given that enterprises and developers might have different set-ups of computing clusters, we share our training techniques over different heterogeneous chips and sizes of servers.

Unlike most LLMs that focus solely on general knowledge from crawled corpora, our work involves training JIUTIAN-139MoE on a high-quality training corpus that incorporates substantial industrial data. This industrial data constitutes nearly 10% of the overall training dataset and encompasses information from various sectors, such as industrial production, technology development, economic growth, environmental protection, and medical care. We collect such industrial data from various data sources, *e.g.*, web pages, books, news articles, industrial knowledge materials, academic papers, etc, and we get a dataset containing high-quality 5T tokens for pre-training after our dedicated pre-processing pipeline, including filtering, cleaning, de-duplication, and tokenization.

We incorporate industrial domain-specific data alongside the common data during pre-training, in contrast to the continued training paradigm adopted by most LLMs [Adler et al., 2024] when absorbing the domain-specific knowledge. We claim that the early incorporation strategy can enable JIUTIAN-139MoE to deeply assimilate the high-quality industrial knowledge and enhance the base model's capacity to construct a more comprehensive general knowledge system. While the potential disparity between the distribution of industrial corpus and that of common data is inevitable, achieving effective collaboration among these distinct data sets poses a significant challenge for conventional Transformer models. To address this challenge, we propose a novel framework JIUTIAN-139MoE, a 13-billion active parameter decoder-only language model based on the Mixture-of-Experts (MoE) architecture. Different from Mixstral 8x7B [Jiang et al., 2024] that adopts the same experts in each FFN layer of Transformer, we introduce a set of diversified experts, *i.e.*, a pair of large twin experts and six small experts, while these experts are different in size, activation rule, and parallelization strategy. We illustrate that the diversified experts can vaguely match the various logic complexity and knowledge mass from different industries.

We train JIUTIAN-139MoE mainly based on our self-developed Jiutian Intelligent computing platform[1], which includes heterogeneous accelerating devices and supports lossless switch between these two heterogeneous devices during training. The training platform utilizes high-speed 1.6Tbps InfiniBand or RoCE non-blocking high-speed interconnect networks, coupled with high-performance dedicated storage. Data parallelism, tensor parallelism, and pipeline parallelism can be well integrated into our training framework. Furthermore, we also leverage the classical acceleration techniques like Flash Attention 2 [Dao et al., 2022, Dao, 2023] and ZeRO-2 [Rajbhandari et al., 2020] to improve hardware utilization.

JIUTIAN-139MoE demonstrates remarkable downstream performance across diverse domains, including 10 standard benchmarks for general evaluation as well as 10 benchmarks for industrial-specific applications. More specifically, JIUTIAN-139MoE-Chat exhibits substantial superiority over several open-source models, *e.g.*, LLaMA2-13B-Chat [Touvron et al., 2023a], Baichuan2-13B-Chat [Yang et al., 2023], Qwen-14B-Chat [Bai et al., 2023], as well as the close model GPT-3.5 [OpenAI, 2023], on the commonly used open benchmarks. Furthermore, our model achieves state-of-the-art results across all 10 industrial benchmarks, while these tasks seem to be challenging for other LLMs. We have released our JIUTIAN-139MoE base and JIUTIAN-139MoE-Chat models under the Apache 2.0 license[2] and JIUTIAN Large Model Community License Agreement, making them accessible for both academic and commercial purposes, thereby broadening their potential for diverse applications.

The contributions of our work are listed as follows:

- We release a large language model JIUTIAN-139MoE, which is based on a novel MoE architecture featuring diversified experts and trained on a large-scale dataset that includes industrial data, yielding impressive performances on both open and self-built benchmarks.
- We have constructed a high-quality training dataset containing 5T tokens, with 10% of the data derived from industrial sectors such as industrial production, technology development, economy growth, environmental protection and medical care.
- JIUTIAN-139MoE is the first open-source LLM that is trained over different heterogeneous chips and sizes of servers.

2 Pre-training

2.1 Model Architecture

JIUTIAN-139MoE is based on a standard decoder-only transformer architecture, similar to LLaMA [Touvron et al., 2023b]. Different from LLaMA, the modification of JIUTIAN-139MoE is mainly at the FeedForward Network (FFN) layers, as illustrated in Figure 1. The modifications are further described below, and the configuration details of JIUTIAN-139MoE are shown in Table 1

FeedForward Network. Similar to Mixtral 8x7B [Jiang et al., 2024], the FFNs are replaced by Mixture-of-Expert layers. We adopt "eight" experts (with a pair of virtual twin experts) and each token is assigned to $top-k$ experts by a router, where $k = 2$. In particular, we introduce special expert activation strategy and a pair of twin experts.

The router is randomly initialized at the beginning of training. For JIUTIAN-139MoE, SwiGLU [Shazeer, 2020a] is selected as the activation function of FFN, which is a combination of Swish [Ramachandran et al., 2017] and Gated Linear Unit [Dauphin et al., 2017]. Since we first train a dense model and then expand to the MoE architecture, in order to preserve the ability of the original dense model, we replace one of "twin" experts directly with the copy of the FFN in the dense model which is denoted as E_0. To ensure that the total number of active parameters is 13B, the other six experts are initialized to half of E_0, denoted as $E_i, i = 1, ..., 6$. When the router selects E_0 and one other expert E_i, only E_0 processes the assigned tokens while E_i remains inactive.

Unlike the traditional MoE, which needs to ensure that each expert load is balanced, we try to get more tokens for E_0. It also aims at better leveraging the abilities of the original dense model. To this end, we introduce a twin expert $E_0^{'}$ as

[1] http://jiutian.hq.cmcc
[2] https://www.apache.org/licenses/LICENSE-2.0.html

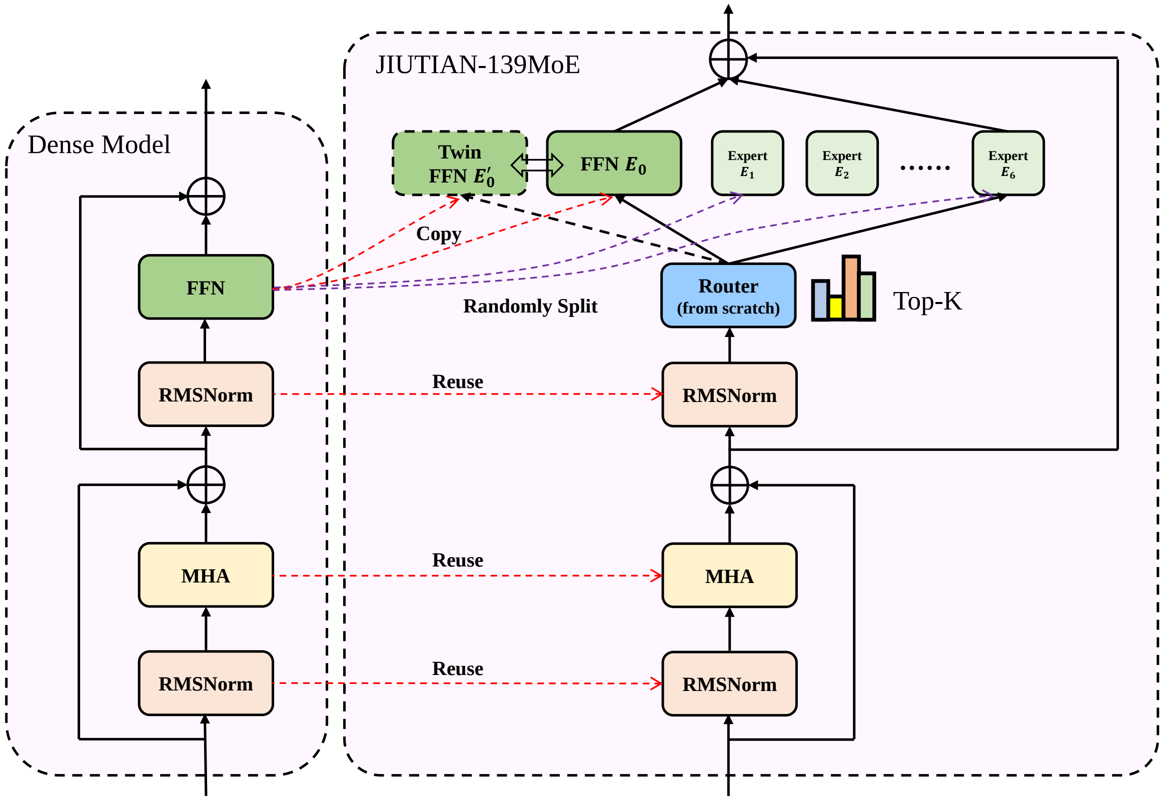

Figure 1: Model Architecture of JIUTIAN-139MoE.

a placeholder, where the parameters are shared with E_0, to get more tokens for E_0. When the $E_0^{'}$ is selected during training, the E_0 will be activated while $E_0^{'}$ will be neither activated nor updated. Consequently, there are only seven expert to be activated in JIUTIAN-139MoE and the probability of E_0 being assigned tokens is close to half.

For the initialization of E_i, to reuse the sunk training costs of the dense model, we utilize the idea of sparse upcycling [Komatsuzaki et al., 2022], which upgrades an existing model with a relatively small additional computational budget. According to LLaMA-MoE [Team, 2023], randomly partitioning E_0 into n equal-sized subsets is a good expert construction method. Therefore, we randomly divide E_0 into two equal-sized subsets to construct the other six experts E_i.

Attention Mechanism and Activation Function. We apply the same Multi-Head Attention (MHA) as LLaMA2 13B model, which can obtain good performance. The SwiGLU [Shazeer, 2020b] is selected as an activation function. Similar to other open-source LLMs, we also reduce the intermediate hidden dimension of the FFN from $4h$ to $\frac{8}{3}h$, where h is the FFN input hidden size.

Table 1: Model details of JIUTIAN-139MoE.

Params Number	Layer Number	Hidden Size	Heads	Sequence Length	Vocabulary Size	Experts Number	FFN Hidden Size	Activated Params
38.8B	40	5120	40	4096	69,120	7	13,824	13B

2.2 Training Infrastructure

Most of our training work is conducted on our self-developed Jiutian Intelligent computing platform, which includes heterogeneous computing resources such as NVLink-based Nvidia GPU and Huawei NPU. It utilizes high-speed 1.6Tbps InfiniBand or RoCE non-blocking high-speed interconnect networks, coupled with high-performance dedicated storage. As mentioned in this paper, we use 896 cards in the training stage. The platform achieves task monitoring and automatic recovery from interruptions, efficiently managing high-performance computing units and high-speed networks, thereby ensuring the stability and efficiency of the training process.

2.3 Training Data

We collect pretraining data from various types of data sources: web-pages, books, news articles, industrial knowledge materials, academic papers, etc. Most of the datasets are written in English and Chinese languages. We still have a small portion of other countries' languages (e.g., Indonesian, Spanish, Arabic, Russian) and code data in Python, C++, or other programming languages.

We have dedicated a lot of effort to collecting industrial knowledge data. Here "industrial knowledge data" are defined as the data that are highly related to industrial production, technology development, economic growth, environmental protection, medical care, and so on. The important industrial areas are telecommunications, energy, transportation, aviation, steel, finance, etc. Those data are mainly collected through 5 types of sources:

- Public academic papers, documents, books about industrial knowledge;
- Valuable documents that are filtered, cleaned, and extracted from public web page data;
- Industrial knowledge problems such as exams, exercises, Q&As;
- Structured knowledge graph data;
- Public high-quality data provided by our cooperative partners, for example, other companies or departments from China Mobile Group.

A comprehensive data preprocessing pipeline is developed to improve the quality of the pretraining data, including:

- Filtering the documents which are illegal, unsafe, unreadable, or advertisements;
- Cleaning the incoherent phrases or characters such as emojis, HTML tags & links, etc.;
- Exact and near deduplication.

It is critical to remove illegal or unsafe documents from the pretraining data. Here ”illegal” or ”unsafe” means the content of the document has issues about politics, race, privacy, violence, etc. We develop model-based and heuristic methods to conduct the detection and filtering at document, sentence, and even word level. As model-based methods, we collect different types of unsafe documents and sentences, together with normal ones, to train a classifier as the detector. As heuristics, we develop a lot of scripts, patterns, and word lists to locate the abnormal content at a phrase or word level.

Apart from illegal and unsafe documents, there are still some low-quality documents such as noisy or unreadable paragraphs, advertisements, HTML tags, java scripts, and emojis. We clean those types of documents using both model-based and heuristic methods as well.

Table 2: Retention rates on datasets after deduplication.

Language	All Datasets	Web-page Datasets
English	70.1%	59.3%
Chinese	73.0%	69.9%

There are two types of duplications: exact duplicates and near duplicates. The simple document matching is used to remove the exact duplicates. And the Minhash LSH deduplication [Broder, 1997] is used to remove the near duplicates. Table 2 shows the retention rate after deduplication on our English and Chinese datasets. Since the web-page-related dataset, *e.g.*, common-crawl [Radford et al., 2019a], has the most duplicates, their retention rates are listed separately.

After filtering, cleaning, deduplication and tokenization, finally we built a dataset encompassing 5 trillion tokens for the pretraining process.

2.4 Tokenization

In our work, we utilize byte pair encoding (BPE) [Sennrich et al., 2015] implemented in the SentencePiece framework [Kudo and Richardson, 2018] as our tokenization method. To enhance the performance of our model on multilingual downstream tasks and downstream tasks in specialized industrial sectors, we train the tokenizer on a smaller subset of the training corpus as described in Subsection 2.3. To ensure computational efficiency during training and to reserve space for any additional special tokens that might be needed in the future, we augment the final vocabulary with 553 special tokens and configure the final model's vocabulary size to 69,120 for training.

2.5 Training Method – Parallelism

During the pre-training stage of the dense large model, we utilize a hybrid parallel approach with 2 tensors parallelisms, 7 pipeline parallelisms, and 64 data parallelisms. The training process is further accelerated by using ZeRO Stage 2 [Rajbhandari et al., 2020] and Flash Attention 2 [Dao, 2023]. In the MoE (Mixture of Experts) enhancement stage, we employ 4 tensor parallelisms, 4 pipeline parallelisms, and 32 data parallelisms, while also implementing a strategy of freezing some non-critical parameters to enhance the training efficiency and specificity. The total training time is approximately 2 months.

We realize the switch of the training platform, that is, a model trained on the Nvidia A800 can be migrated to Huawei Ascend 910B. The whole switching process mainly consists of three steps: (1) Data docking. The data is preprocessed so that it can be adapted to the Ascend 910B. (2) Distributed training framework adaptation. For Huawei platform, we adopt the AscendSpeed framework, which can be adapted to the 910B and take full advantage of its hardware performance. For weight conversion, AscendSpeed only supports the conversion of weight in HuggingFace format to pt format. Therefore, model weights trained on the Nvidia platform should be converted to HuggingFace format before they can be converted to pt format. (3) Configuration migration. This is the most important part of the switching process. We need to modify the corresponding configuration in AscendSpeed based on the configuration of the original model and the parallel training strategies. After the above three steps, the model can be successfully migrated to Huawei platform and further trained on a new platform. The migration of a model trained on Huawei platform to Nvidia platform is similar to the process described above.

3 Alignment

3.1 Data Preprocessing

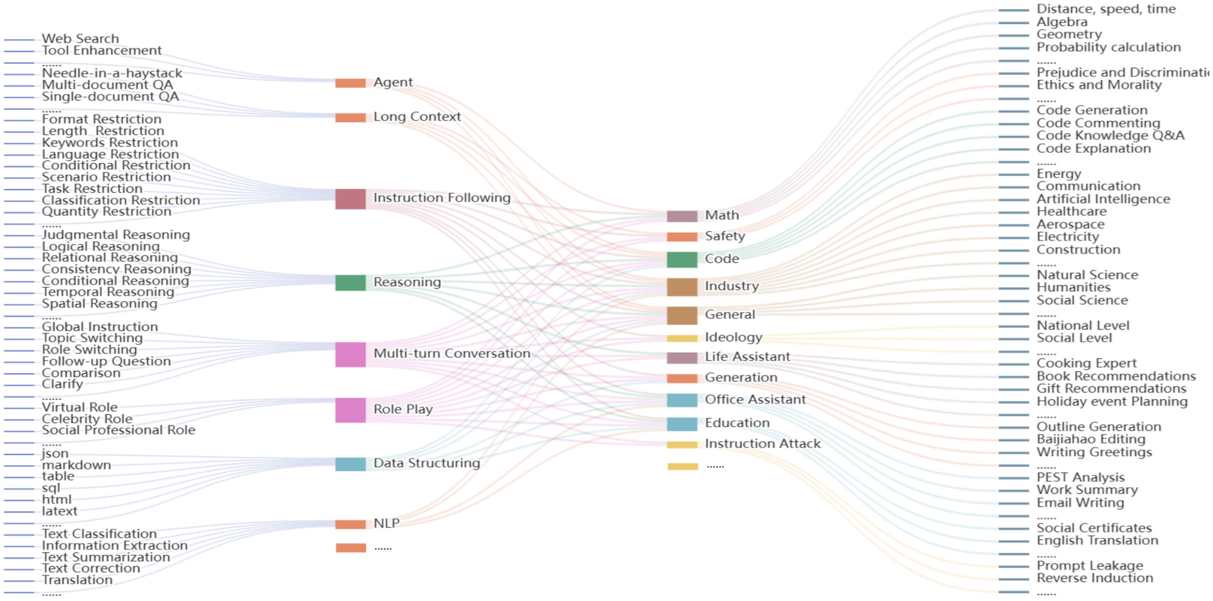

Figure 2: Data Hierarchical System and Relationships.

To fully unlock the potential of pre-trained models in chat use cases and better align the model with human values and expectations, it is necessary to perform post-training via Supervised Fine-Tuning (SFT) [Ouyang et al., 2022], Direct Preference Optimization (DPO) [Rafailov et al., 2024], and etc. The curation of a high-quality training dataset is paramount. We have constructed a dataset containing tens of millions of instruction data instances and implemented a detailed hierarchical system for this dataset, which includes 113 domains and 53 capabilities. As illustrated in Figure 2, each domain requires the preparation and processing of data related to multiple associated capabilities. We carefully construct and curate data for all capabilities associated with each domain. Multiple iterative experiments have validated the effectiveness of our data construction strategy.

Specifically, different fields and skills require significantly different training methods and stages, which directly affect the data construction approach and quality requirements. In the following section, we will consider these differences using instruction-following capabilities and the medical field as examples. We will describe and discuss the characteristics of data construction for different fields and skills based on various training methods and stages.

Supervised Fine-Tuning (SFT). To develop instruction-following capabilities, we introduce a foundational instruction dataset containing multiple tasks through refined manual annotation and generate foundational instructions and additional instructions using an instruction decomposition approach. Furthermore, to enhance the diversity and complexity of the instruction data, we expand the instruction dataset following the principles of "rewriting foundational instructions, adding additional instructions, and increasing difficulty levels" [Sun et al., 2024]. Additionally, to prevent conflicts between instructions, we filter conflicting instruction data using a rule-and-model-based strategy. Ultimately, based on the filtered instruction data, we generate a high-quality instruction-following fine-tuning dataset.

Direct Preference Optimization (DPO). To further enhance the model's instruction-following capability, we construct the instruction dataset in the same manner as in the SFT stage and perform instruction deduplication. Using the new data, we generate data based on the SFT model through a rejection sampling strategy. Positive and negative sample pairs, representing the direct preference alignment training data, are obtained via automated evaluation. The model's instruction-following capability is significantly improved after further optimization using the DPO strategy, as demonstrated in Figure 3.

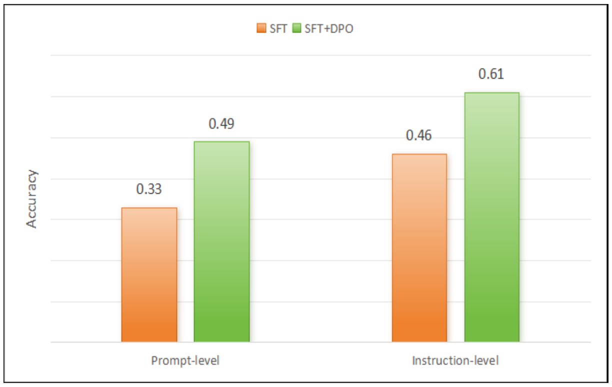

Figure 3: Comparison of Instruction-Following Experiments.

Reinforcement Learning from Human Feedback (RLHF). Although SFT and DPO show significant performance improvements in certain aspects of large models, in actual medical field testing, it has been found that with the increase of the temperature parameter, large models can generate both high-quality answers and low-quality answers. To further improve the capabilities of large models, we decide to use RLHF to further optimize the large model.

Before executing RLHF, we carefully tune a reward model to evaluate data preferences during training. We sample a medical instruction subset containing single-round and multi-round data. To improve the diversity and upper limit of the model's generation, we adopt a diversified sampling strategy in the SFT stage, including the addition of specific system instructions. Through data Filtering and self-sampling data enhancement, we construct 100,000 high-quality preference data for training the reward model, achieving an accuracy rate of 86% on the test set.

The RLHF training data is sampled from a non-repetitive subset that is independently and identically distributed with the SFT data. We find that by adding detailed system instructions, we can guide our model to generate more detailed and structurally clearer content thereby improving the model's upper limit. Based on the above observations, we use the large model to generate multiple samples by adding system instructions compared to normal sampling for the same prompt. We evaluate the multiple samples through the above reward model to filter the best and worst responses, and form preference data. Finally, we construct 20,000 instructions and their corresponding preference data for training the DPO model. Compared with the original SFT model, it improves by 10 percentage points on the AlpacaEval2.0 LC-winrate.

Furthermore, we adopt an iterative ORPO scheme, with a learning rate of 5e-6 and a batch size of 640, training 6 epochs each iteration. After the first ORPO, we resample 20,000 data according to the above scheme for the second ORPO. After a total of three ORPO training, it finally improves by 20 percentage points on the AlpacaEval2.0 LC-winrate compared to the original SFT model.

Finally, we provide an example to illustrate, for the medical scenario, under this data construction and training method, the final result shows significant advantages in content richness and structural clarity compared to the original SFT model, as shown in Table 3.

3.2 Long Context Extension

The long context capability of large models is very useful in many real-world application scenarios. After the initial pre-training stage, we use an NTK-Aware incremental training method to modify the base value in the rotary position encoding to 1,000,000, extending the context length that our model can handle from 4K to 32K. The incremental training data is 40B tokens, with 67% of long text data longer than 16K and 33% of short text data shorter than 4K.

The long text fine-tuning data includes various tasks such as single document QA, multi-document QA, and document summarization, which are mainly constructed through synthetic data. Taking the multi-document QA task as an example, we select a short document data, extract the Q&A pair from it, and then randomly splice it with other short documents to form a long text input longer than 8K. At the same time, we randomly require the answer to restate the corresponding document number in the question to improve the model's retrieval capability in long text tasks. Figure 4 shows the results of the model on the "Needle in a Haystack" test.

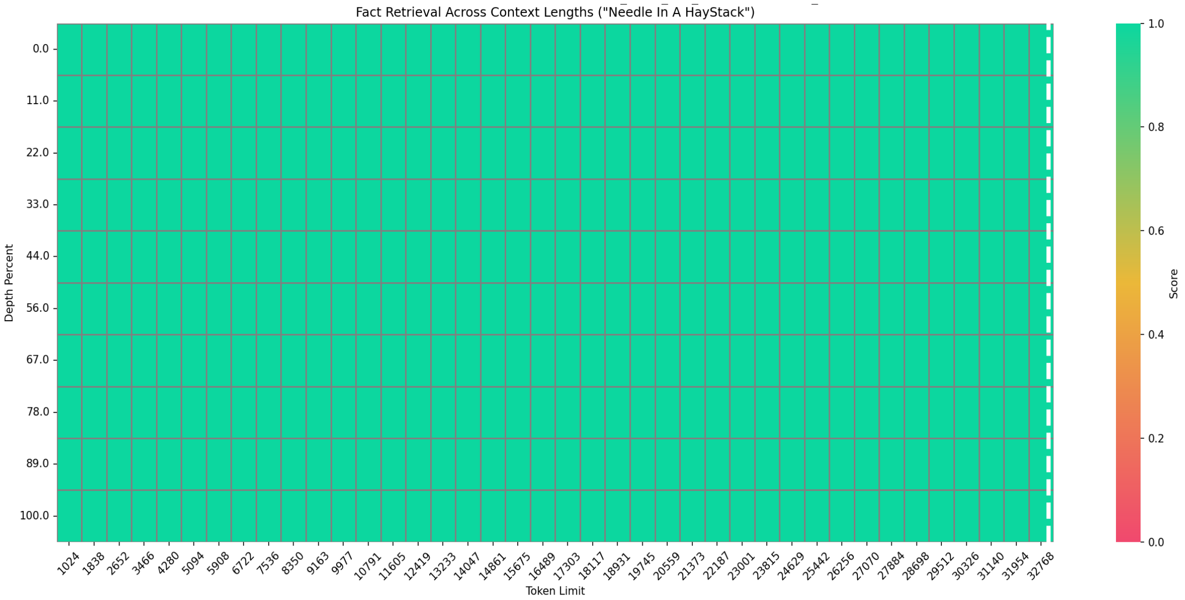

Figure 4: Needle in a Haystack Experiment Results.

3.3 Training Strategy

Data Mixing. In the SFT stage, we adopt a data mixing method based on weight proportion, configuring weights according to factors such as the domain, difficulty level, and contribution to model capability, ensuring full coverage of various capability points during the training process.

Table 3: The results of ORPO on medical scenario, compared to SFT.

Question: *How to manage heart failure with renal insufficiency?*

By SFT Model:

Control of Heart Failure: First, it is essential to control the symptoms of heart failure. This may involve using diuretics (such as furosemide or hydrochlorothiazide) to reduce fluid accumulation in the body and using ACE inhibitors or ARBs (such as benazepril or losartan) to help improve cardiac function. Beta-blockers may also be used to help reduce the workload on the heart.

Protection of the Kidneys: While controlling heart failure, it is crucial to protect renal function. ACE inhibitors and ARBs can help protect the kidneys as they can reduce proteinuria, a significant factor leading to renal insufficiency. Additionally, renal function should be regularly monitored, including serum creatinine and blood urea nitrogen levels.

Dietary Adjustments: A low-salt diet can help control the symptoms of heart failure, while a low-protein diet can help protect the kidneys. However, this should be adjusted according to the specific conditions of the patient, as excessive restriction may lead to malnutrition.

Treatment of Renal Insufficiency: For severe renal insufficiency, dialysis or kidney transplantation may be required. This decision should be made based on the patient's specific circumstances and preferences.

Lifestyle Changes: Quitting smoking, limiting alcohol, maintaining an appropriate weight, and regular exercise can all help improve heart and kidney health.

Regular Follow-ups: Regular medical checkups can help monitor disease progression and adjust treatment plans promptly.

Please note that the above suggestions are for reference only, and the specific treatment plan should be developed by a physician based on the patient's specific circumstances.

By ORPO Model:

Heart failure combined with renal insufficiency is a complex clinical condition requiring comprehensive treatment and management strategies. Key steps and treatment measures include:

1. **Assessment and Monitoring**:

- Conduct a detailed evaluation of cardiac and renal function, including the use of functional classification and estimated glomerular filtration rate (eGFR) indicators. - Regularly monitor blood pressure, electrolyte levels, serum creatinine, and urine output to assess renal function status.

2. **Fluid Balance Control**:

- Depending on the severity of heart failure, it may be necessary to limit sodium and water intake to reduce fluid retention.
- Avoid medications that may exacerbate heart failure or renal failure, such as certain non-steroidal anti-inflammatory drugs.

3. **Optimizing Heart Failure Treatment**:

- Use heart failure-specific medications such as ACE inhibitors, ARBs, β-blockers, and diuretics, but pay attention to dosage and frequency to avoid further deterioration of renal function.
- Consider using SGLT2 inhibitors, which have been shown to benefit heart failure and may improve renal function.

4. **Managing Renal Insufficiency**:

- Depending on the extent of renal impairment, it may be necessary to adjust medication dosages or select different administration routes.
- Consider using renal-protective drugs such as ACE inhibitors or ARBs, but be careful to avoid causing hypotension.
- In cases of severe renal failure, dialysis support may need to be considered.

5. **Lifestyle Adjustments**:

- Encourage patients to engage in moderate physical activity and avoid overexertion.
- Control weight, maintain a healthy diet, and limit high-protein food intake.
- Quit smoking, limit alcohol consumption, and avoid exposure to harmful substances.

6. **Multidisciplinary Team Collaboration**:

- Cardiologists, nephrologists, dietitians, and nursing staff should work closely together to develop and adjust treatment plans.
- Conduct regular assessments and modify treatment regimens accordingly.

7. **Psychological Support**:

- Heart failure and renal insufficiency can cause significant psychological stress for patients, so providing psychological counseling and support is crucial.

8. **Educating Patients and Families**:

- Educate patients and families about the disease, how to recognize changes in condition, and how to manage medication and lifestyle adjustments.

Managing heart failure and renal insufficiency is a long-term process that requires continuous medical attention and adjustments in personalized treatment plans.

Training Efficiency. Due to the varying lengths of data in the SFT stage, there will be a large number of padded tokens in the batch during training, wasting computing resources. To solve this problem, we use the packing strategy to reorganize the data and ensure that there would be no cross-contamination between samples after packing through attention masking and position encoding reset techniques. Our experiments show that the training efficiency was significantly improved using this strategy, with the training time reduced to about 1/4 of the original time.

Hyper-parameter Settings. In the SFT stage, we adopt AdamW as the optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.98$, and $epsilon =$1e-8. We implement a cosine learning rate decay strategy with a warm-up period of 100 steps and a maximum learning rate of 5e-6. The training rounds are set as 3, and only the loss of the response is calculated during the process. In the DPO stage, we set the beta value to 0.1 and adjusted the learning rate to 5e-7, with the rest of the parameters remaining unchanged.

4 Evaluation

In this section, we conduct a comprehensive evaluation and analysis of the performance of the base model JIUTIAN-139MoE and the chat model JIUTIAN-139MoE-Chat across diverse domains, encompassing both widely adopted public benchmarks and self-built benchmarks tailored for specific industrial applications and safety considerations. Specifically, the public benchmarks involve tasks covering language and knowledge understanding, reasoning, mathematics, and coding. Meanwhile, the self-built benchmarks originate from several critical industrial applications, such as industry-specific courses, skill-level examinations, and job interviews. All evaluations on the open benchmarks are conducted using OpenCompass [Contributors, 2023], a fair, open-source, and comprehensive platform designed for evaluating large models, ensuring a standardized and consistent assessment across various models.

4.1 Performance on Open Benchmarks

In this section, we report the results of our released models on the public benchmarks, and we also make a comparison between them and many open-source models, *e.g.*, Qwen-14B-Chat [Qwen-Team, 2023], Baichuan2-13B-Chat [Yang et al., 2023], LLaMA2-13B-Chat [Touvron et al., 2023a], as well as the proprietary model GPT3.5 [OpenAI, 2023]. The results of these comparative models are gathered from their official reports or the OpenCompass Leaderboard [3].

4.1.1 Open Benchmarks

We evaluate our models on public benchmarks covering multiple domains, including language and knowledge understanding, reasoning, mathematics, and coding, which are commonly used to assess the capability of large language model. More details about task description and experimental setup are listed as follows:

General Knowledge Understanding: We conduct evaluations based on a series of comprehensive examinations to assess the capability of our models to understand language and general knowledge. There are 5 classical benchmarks for evaluation: **MMLU** [Hendrycks et al., 2020], **C-Eval** [Huang et al., 2024], **CMMLU** [Li et al., 2023], **GaokaoBench** [Zhang et al., 2023], **AgiEval** [Zhong et al., 2023]. We report 5-shot results for MMLU, C-Eval, and CMMLU. While for GaokaoBench and AgiEval, we report 0-shot results.

Reasoning: We adopt the challenging benchmark **BBH** [Suzgun et al., 2022] to access the reasoning ability of our models. It contains 23 challenging tasks from BIG-Bench, where contemporary language models had not surpassed human performance at the time. We report the results based on a 3-shot approach.

Mathematics: Mathematical proficiency is an integral part of a model's cognitive and computational ability. We exploit **GSM8K** [Cobbe et al., 2021] and **Math** [Hendrycks et al., 2021] for evaluation, and the results for these benchmarks are reported based on 4-shot and 0-shot settings separately.

Coding: We report the Pass@1 scores on **HumanEval** [Chen et al., 2021] based on 0-shot approach and the results on **MBPP** [Austin et al., 2021] based on 3-shot approach to access the model's coding proficiency.

[3] https://rank.opencompass.org.cn/home

4.1.2 Base Model Performance

We compare our based model against open-source base models: Qwen-14B [Qwen-Team, 2023], Baichuan2-13B [Yang et al., 2023], and LLaMA2-13B [Touvron et al., 2023a], as shown in Table 4. While our JIUTIAN-139MoE base model containing diversified industrial experts is specifically designed for industrial applications, we can still achieve comparable results to Baichuan2-13B and LLaMA2-13B across most benchmarks. That demonstrates the capacity of JIUTIAN-139MoE in understanding general knowledge has not been impaired even been trained with large-scale industrial-specific data. Therefore, with further instruction tuning, we can effortlessly outperform these top-performing models and get state-of-the-art performance on several benchmarks as illustrated in the following sections.

Table 4: Evaluation results of base models on open benchmarks. (**Gaokao** denotes GaokaoBench benchmark and **H-Eval** is the HumanEval benchmark. **Bold** indicates the top score among all models.)

Model	**MMLU** *5-shot*	**C-Eval** *5-shot*	**CMMLU** *5-shot*	**AgiEval** *0-shot*	**Gaokao** *0-shot*	**GSM8k** *4-shot*	**Math** *0-shot*	**BBH** *3-shot*	**H-Eval** *0-shot*	**MBPP** *3-shot*
Qwen-14B	**67.9**	**71.7**	**70.2**	**51.9**	**62.5**	**61.6**	**25.2**	**53.7**	**32.3**	**39.8**
Baichuan2-13B	53.6	55.3	55.5	36.6	38.7	26.6	4.7	42.6	14.6	23.2
LLaMA2-13B	55.0	41.4	38.4	30.9	18.2	29.6	5.0	45.6	18.9	26.8
JIUTIAN-139MoE	54.4	54.5	56.3	26.7	29.2	32.7	4.3	30.3	20.1	26.4

4.1.3 Chat Model Performance

We conduct a comprehensive evaluation of our JIUTIAN-139MoE-Chat model on a wide range of open benchmarks. We also show our evaluation results on 3 open-source chat models: Qwen-14B-Chat[Qwen-Team, 2023], Baichuan2-13B-Chat[Yang et al., 2023], LLaMA2-13B-Chat[Touvron et al., 2023a], and the proprietary model GPT3.5[OpenAI, 2023].

It turns out that JIUTIAN-139MoE-Chat performs better than other models on most benchmarks (see Table 5). These results show the potential of the base model JIUTIAN-139MoE has been unleashed after alignment. Compared to our base model, JIUTIAN-139MoE-Chat obtains remarkable improvement on all benchmarks.

For the exceptions, JIUTIAN-139MoE-Chat is still competitive among all advanced models. As we can see, GPT3.5 achieves the best performance on both GSM8k and MBPP. However, JIUTIAN-139MoE-Chat performs better than all three open-source language models on these two benchmarks. As for the GaokaoBench benchmark, Qwen-14B-Chat performs the best. We can see that JIUTIAN-139MoE-Chat still obtains a relatively high score compared to other models.

Table 5: Evaluation results of chat models on open benchmarks. (**Gaokao** denotes GaokaoBench benchmark and **H-Eval** is the HumanEval benchmark. **Bold** indicates the top score among all models. JIUTIAN-139MoE-Chat performs better than other models on the majority of the benchmarks.)

Model	**MMLU** *5-shot*	**C-Eval** *5-shot*	**CMMLU** *5-shot*	**AgiEval** *0-shot*	**Gaokao** *0-shot*	**GSM8k** *4-shot*	**Math** *0-shot*	**BBH** *3-shot*	**H-Eval** *0-shot*	**MBPP** *3-shot*
Qwen-14B-Chat	66.4	71.7	70.0	47.3	**76.5**	61.0	26.8	58.0	36.6	23.8
Baichuan2-13B-Chat	50.5	53.4	50.7	32.2	40.9	36.3	7.3	45.4	21.3	26.8
LLaMA2-13B-Chat	54.6	36.2	38.7	32.3	18.6	37.1	5.2	40.2	18.9	27.2
GPT3.5	69.1	52.5	53.9	39.9	51.1	**78.2**	28.0	70.1	73.2	**60.2**
JIUTIAN-139MoE-Chat	**73.0**	**85.7**	**70.2**	**52.1**	48.7	63.1	**43.9**	**76.7**	**76.2**	59.0

4.2 Performance on Self-built Industry-Standard Benchmarks

An essential characteristic of our JIUTIAN-139MoE-Chat model, is its outstanding performance on industrial domain-specific tasks. Specifically, to enhance our model's industry-specific capabilities, we train JIUTIAN-139MoE-Chat with tremendous data obtained from diverse industrial domains, including communication, electric power, transportation, energy, steel, construction, etc.

4.2.1 Industry-Standard Benchmarks

We conduct evaluations with a self-built test dataset comprising industrial data closely related to human livelihood and welfare. Sources of the test data include industry-specific courses, skill-level examinations, and job interview questions. After rigorous cleaning and screening, we randomly select 200 objective multiple-choice questions for accuracy evaluation in each industry, ensuring a balanced representation across industry subdivisions and difficulty levels. Below is the list of industries against which we evaluate our model:

- **Electric Power**: Data on various aspects of the electrical industry, spanning from electrical safety, maintenance and monitoring, to dispatching and control. It also delves into the intricacies of power generation engineering, transmission engineering, distribution engineering, as well as analog and digital electronics technology. Furthermore, it incorporates circuitry and electromagnetism principles, fundamental electrical knowledge, and the skills required for electrical appliance maintenance.
- **Steel Industry**: Data on steel smelting, steel casting, rolling techniques, alloy materials, steel inspection, welding processes, steel production safety, and the evolution of the steel industry.
- **Aerospace**: Data on aerospace engineering, including aerospace structures, avionics, aerodynamics, and aviation meteorology. It delves into the intricacies of aircraft design and manufacturing, as well as aviation transportation management. Additionally, it covers crucial aspects such as spaceflight safety management, aviation maintenance management, and aviation materials.
- **Construction**: Data on architectural history and a range of professional roles, including registered safety engineers, registered constructors, registered cost engineers, registered supervision engineers, safety officers, material controllers, machinery operators, construction technicians, testing personnel, quality controllers, fire protection engineers, environmental engineers, consulting engineers, BIM modelers.
- **Finance**: Data on a diverse range of professionals including accountants, auditors, tax advisors, economists, statisticians, asset evaluators, risk managers, market analysts, international trade specialists, securities investors, monetary bankers, and those with a foundation in economic theory.
- **Energy**: Data on clean coal technology, nuclear power, solar energy, biomass energy, hydropower, wind energy, geothermal energy, hydrogen energy, petrochemical engineering and other emerging energy sources.
- **Judiciary**: Data on a broad range of disciplines, including constitutional law, criminal law, civil law, commercial law, intellectual property law, economic law, labor law, environmental and resource law, administrative law, procedural law, international law, legal history, as well as legal ethics and professional responsibilities.
- **Telecommunications**: Data on fundamental communication technologies, wired communication techniques, wireless communication systems, network communication protocols, data transmission methodologies, communication standards and protocols, the design and implementation of communication systems, communication security and encryption techniques, communication software and applications, communication hardware and equipment, optical communication technologies, as well as emerging communication technologies.
- **Firefighting**: Data on fire engineers, operators of fire-fighting facilities, firefighters, the science of fire combustion, fire safety regulations, fire prevention and explosion protection techniques, fire safety management, and the execution of fire engineering projects.
- **Medical Industry**: Data on a wide range of medical specialties, including internal medicine, surgery, obstetrics and gynecology, pediatrics, otorhinolaryngology (head and neck surgery), dentistry, ophthalmology, traditional Chinese medicine, dermatology, pathology, ultrasonography, laboratory medicine, rehabilitation medicine, clinical nutrition, and medical psychology.

4.2.2 Evaluation Results

Table 6 presents the evaluation results on this suite of industrial domain-specific evaluation datasets, which indicates that JIUTIAN-139MoE-Chat could excel in a variety of industrial fields, surpassing state-of-the-art large language models on a collection of tasks. In the evaluation of industrial applications, we predominantly use the zero-shot approach.

To be specific, our model outperforms cutting-edge models in areas such as aerospace, construction, finance, energy, judiciary, firefighting and the medical industry. For instance, the aerospace sector provides transportation services while enabling sophisticated scientific research. The construction industry has far-reaching impacts ranging from creating employment opportunities to driving innovation. The finance sector influences the flow of capital in the market. The energy sector functions in almost all crucial infrastructure sectors. The judiciary is important to the appropriate interpretation and application of laws. Firefighting helps ensure people's safety by preventing and extinguishing fires. The medical industry has greatly contributed to the increment in life expectancy for human beings worldwide.

Table 6: Evaluation results of chat models on the self-built industry-standard dataset. (Due to page limitations, we abbreviate Telecommunications to **Tele.** and Firefighting to **Fire.**. The highest score for each benchmark is **bold**.)

Model	**Electric Power**	**Steel Industry**	**Aero-space**	**Constr-uction**	**Finance**	**Energy**	**Judiciary**	**Tele.**	**Fire.**	**Medical Industry**
Qwen-14B-Chat	**84.0**	**70.5**	83.0	53.5	58.5	77.5	40.5	**71.5**	58.5	62.5
Baichuan2-13B-Chat	30.0	25.5	45.5	25.5	25.5	62.0	28.0	30.5	24.5	37.5
LLaMA2-13B-Chat	16.5	14.0	18.0	19.5	14.0	27.0	15.5	9.0	16.0	21.0
GPT3.5	47.0	51.0	72.0	36.5	34.5	64.0	28.0	46.5	42.0	38.5
JIUTIAN-139MoE-Chat	66.5	68.5	**85.0**	**58.5**	**59.5**	**78.5**	**52.5**	67.0	**65.0**	**64.5**

JIUTIAN-139MoE-Chat also achieves outstanding performance on the rest of the benchmarks, which cover test data on electric power, the steel industry and telecommunications. Electric power enhances our living quality, ensuring safety. The steel industry is deemed the most dynamic industry across the globe. Last but not least, telecommunication lays a technological foundation for societal communications.

In a nutshell, our results demonstrate that our JIUTIAN-139MoE-Chat model possesses superior capabilities across various industries compared to state-of-the-art models, positioning it as a powerful tool for diverse industry-specific applications.

4.3 Performance on Safety

Following other state-of-the-art large models, we undertake a range of safety evaluations on our JIUTIAN-139MoE model. In this section, we focus on evaluating the safety capabilities of our model and other current powerful LLMs, including content security and instruction security.

Content security mainly covers five major categories listed in the "Basic Requirements for Security of Generative Artificial Intelligence Services [SAC/TC260]", including Chinese values, discrimination, commercial illegalities, infringement of others' rights, and reliability. We further refine and self-build upon these categories by creating 68 subcategories and 35000 pieces of evaluation data in Chinese.

Instruction security encompasses 30 attack methods across 15 types of vulnerabilities, including role-playing, reverse engineering, privilege escalation, token manipulation, and other tactics.

We use a security score to calculate the percentage of security responses in the total number of responses, and the value is positively correlated with the security of the model:

$$Score = Num(safe\_response) \div Num(all\_reponse) * 100 \tag{1}$$

where $Num(*\_response)$ denotes the number of the related responses. A higher score indicates a larger proportion of safe responses, signifying greater security of the model.

Table 7: Evaluation results on security capabilities. (Bold indicates the top score among all models.)

Model	**Chinese Values**	**Discrimination**	**Commercial Illegality**	**Infringement**	**Reliability**	**Instruction Security**
LLaMA2-13B-Chat	18.6	8.6	45.5	22.0	23.5	40.1
Baichuan2-13B-Chat	42.8	45.2	80.5	60.0	83.5	59.2
Qwen-14B-Chat	70.8	76.1	**90.0**	**76.5**	**96.5**	77.9
JIUTIAN-139MoE-Chat	**80.2**	**84.2**	83.0	70.0	73.0	**80.8**

We evaluate our model on our safety benchmarks, using the internal evaluation framework, and compare the results with other open-source LLMs: LLaMA2-13B-Chat [Touvron et al., 2023a], Qwen-14B-Chat [Bai et al., 2023], and Baichuan2-13B-Chat [Yang et al., 2023]. Table 7 shows the results.

The first five categories in Table 7 present the comparison results of content security. We can observe that JIUTIAN-139MoE-Chat achieves consistently decent performance across all categories and is significantly superior to other models of discrimination prevention and consistency with Chinese values. In terms of Infringement, it performs slightly inferior to the best model Qwen-14B-Chat but significantly surpasses other models. Regarding reliability and avoiding commercial illegalities, our JIUTIAN-139MoE-Chat also performs comparably to the first tier. For instruction security, results from the last column of Table 7 demonstrate our absolute advantage among all the models involved in the comparison. Overall, the JIUTIAN-139MoE-Chat performs consistently well when being evaluated from various aspects of security.

5 Related Work

Language models based on the Transformer architecture [Vaswani et al., 2017] have demonstrated remarkable capabilities after large-scale parameterization and self-supervised learning on extensive datasets Kaplan et al., 2020. One of the most representative works is OpenAI's GPT3.5 [OpenAI, 2022], which has showcased impressive conversational abilities, attracting significant investment and research interest from numerous companies and researchers in this field Shoeybi et al., 2019. In general, building a large language model involves two main stages: pre-training and alignment [Achiam et al., 2023]. As the number of model parameters and the scale of training data increase, the cost of training such models escalates sharply, making it challenging for many enterprises and researchers to develop from the beginning [Hoffmann et al., 2022]. Open-source LLMs allow more people to access and use these models, develop downstream applications based on them, and significantly promote related research development and industry empowerment [Touvron et al., 2023b,a, Adler et al., 2024, Jiang et al., 2024].

Two crucial elements in constructing large models that require careful design are the composition of the data and the model architecture.

Data. Training corpus is the critical source for building the capabilities of large models. It can be categorized into general-purpose data and domain-specific data. General-purpose data, which is responsible for building the foundational capabilities of the model, typically includes web pages [Wenzek et al., 2019], conversational data [Roller et al., 2020], and books [Gao et al., 2020]. Domain-specific data enhances the model's specific abilities by supplementing industry-specific knowledge, which is closely related to downstream applications in various sectors [Lozhkov et al., 2024].

Architecture. The Transformer architecture [Vaswani et al., 2017] is currently the mainstream backbone for large models. Typically, the widely used large models today can be categorized into causal decoders and Mixture-of-Experts (MoE). Causal decoders are extensively used in various NLP tasks such as text generation, machine translation, and dialogue systems. The main characteristic of causal decoders is that they generate the next token using only the previously generated tokens as input. This method ensures that the model follows the causal sequence of natural language during text generation, meaning the current word can only depend on the preceding words and cannot "see" the subsequent words. Currently, most large language models use this structure as their foundational architecture, and it has been demonstrated that scaling up based on this structure results in significant performance improvements, along with excellent generalization and zero-shot capabilities [Brown et al., 2020, Radford et al., 2019b]. MoE is an advanced machine learning model architecture that enhances model performance and efficiency by combining multiple "expert" models [Fedus et al., 2022b, Du et al., 2022]. It includes several independent sub-model experts, each excelling in processing specific types of data or tasks. A gating mechanism is responsible for selecting and allocating input data to the most appropriate expert. During each inference, only a small subset of expert models is activated, maintaining computational efficiency and reducing resource consumption. Mixtral 8x7B [Jiang et al., 2024] implements this by replacing the Feed-Forward Network (FFN) layer in the Transformer with an MoE FFN layer, while keeping other components like the attention mechanism unchanged. DeepSeekMoE [Dai et al., 2024] employs an innovative MoE architecture involving two primary strategies: fine-grained expert segmentation and shared experts isolation.

6 Conclusion and Final Discussion

We showcase the JIUTIAN-139MoE model in this report, representing a milestone in China Mobile's development of large language models. The model has the characterizations of a novel MoE architecture and the utilization of industrial domain-specific training data. In addition, it enjoys the benefits of a set of innovative MoE architecture, featured by the twin FFN design and the synergy of large and small-size experts. Besides instilling massive general knowledge, JIUTIAN-139MoE integrates professional knowledge from various industries such as communication, electric power, transportation, energy, steel and construction. Evaluation results demonstrate that JIUTIAN-139MoE-Chat could excel in respective industrial fields, and outperform existing open-source or proprietary models on a collection of tasks.

In the future, we will continue to promote the construction and application of the JIUTIAN large language model, laying a solid foundation for boosting the development of the digital and real economy, and helping all industries forge new productivity. We would also like to encourage research efforts paid to methods such as chain-of-thought that could further improve model capability, and delving into the issue of model interpretability and explainability to make our model safer and more aligned.

References

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv preprint arXiv:2311.05232*, 2023. URL https://arxiv.org/pdf/2311.05232.

Hongbin Ye, Tong Liu, Aijia Zhang, Wei Hua, and Weiqiang Jia. Cognitive mirage: A review of hallucinations in large language models. *arXiv preprint arXiv:2309.06794*, 2023. URL https://arxiv.org/pdf/2309.06794.

SM Tonmoy, SM Zaman, Vinija Jain, Anku Rani, Vipula Rawte, Aman Chadha, and Amitava Das. A comprehensive survey of hallucination mitigation techniques in large language models. *arXiv preprint arXiv:2401.01313*, 2024. URL https://arxiv.org/pdf/2309.06794.

William Fedus, Jeff Dean, and Barret Zoph. A review of sparse expert models in deep learning. *arXiv preprint arXiv:2209.01667*, 2022a. URL https://arxiv.org/pdf/2209.01667.

Bo Adler, Niket Agarwal, Ashwath Aithal, Dong H Anh, Pallab Bhattacharya, Annika Brundyn, Jared Casper, Bryan Catanzaro, Sharon Clay, Jonathan Cohen, et al. Nemotron-4 340b technical report. *arXiv preprint arXiv:2406.11704*, 2024. URL https://arxiv.org/pdf/2406.11704.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024. URL https://arxiv.org/pdf/2401.04088.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/67d57c32e20fd0a7a302cb81d36e40d5-Paper-Conference.pdf.

Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*, 2023. URL https://arxiv.org/abs/2307.08691.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. Zero: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–16. IEEE, 2020. URL https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9355301.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023a. URL https://arxiv.org/abs/2307.09288.

Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, et al. Baichuan 2: Open large-scale language models. *arXiv preprint arXiv:2309.10305*, 2023. URL https://arxiv.org/pdf/2309.10305.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023. URL https://arxiv.org/pdf/2309.16609.

OpenAI. gpt-3.5-turbo. https://platform.openai.com/docs/models/gpt-3.5-turbo, 2023. URL https://platform.openai.com/docs/models/gpt-3.5-turbo.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023b. URL https://arxiv.org/pdf/2302.13971.

Noam Shazeer. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020a. URL https://arxiv.org/pdf/2002.05202.

Prajit Ramachandran, Barret Zoph, and Quoc V Le. Searching for activation functions. *arXiv preprint arXiv:1710.05941*, 2017. URL https://arxiv.org/pdf/1710.05941.

Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. Language modeling with gated convolutional networks. In *International conference on machine learning*, pages 933–941. PMLR, 2017. URL http://proceedings.mlr.press/v70/dauphin17a/dauphin17a.pdf.

Aran Komatsuzaki, Joan Puigcerver, James Lee-Thorp, Carlos Riquelme Ruiz, Basil Mustafa, Joshua Ainslie, Yi Tay, Mostafa Dehghani, and Neil Houlsby. Sparse upcycling: Training mixture-of-experts from dense checkpoints. *arXiv preprint arXiv:2212.05055*, 2022. URL https://openreview.net/pdf?id=T5nUQDrM4u.

LLaMA-MoE Team. Llama-moe: Building mixture-of-experts from llama with continual pre-training, dec 2023. *URL https://github. com/pjlab-sys4nlp/llama-moe*, 2023. URL https://github.com/pjlab-sys4nlp/llama-moe.

Noam Shazeer. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020b.

Andrei Z Broder. On the resemblance and containment of documents. In *Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No. 97TB100171)*, pages 21–29. IEEE, 1997.

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

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*, 2015. URL https://aclanthology.org/P16-1162.pdf.

Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*, 2018. URL https://aclanthology.org/D18-2012.pdf.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2024.

Haoran Sun, Lixin Liu, Junjie Li, Fengyu Wang, Baohua Dong, Ran Lin, and Ruohui Huang. Conifer: Improving complex constrained instruction-following ability of large language models. *arXiv preprint arXiv:2404.02823*, 2024.

OpenCompass Contributors. Opencompass is an llm evaluation platform, supporting a wide range of models (llama3, mistral, internlm2,gpt-4,llama2, qwen,glm, claude, etc) over 100+ datasets. https://github.com/open-compass/OpenCompass/, 2023. URL https://github.com/open-compass/OpenCompass/.

Qwen-Team. Qwen-vl. https://github.com/QwenLM/Qwen-VL, 2023. URL https://github.com/QwenLM/Qwen-VL.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020. URL https://arxiv.org/pdf/2009.03300.

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Yao Fu, et al. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. *Advances in Neural Information Processing Systems*, 36, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/c6ec1844bec96d6d32ae95ae694e23d8-Paper-Datasets_and_Benchmarks.pdf.

Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. Cmmlu: Measuring massive multitask language understanding in chinese. *arXiv preprint arXiv:2306.09212*, 2023. URL https://arxiv.org/pdf/2306.09212.

Xiaotian Zhang, Chunyang Li, Yi Zong, Zhengyu Ying, Liang He, and Xipeng Qiu. Evaluating the performance of large language models on gaokao benchmark. *arXiv preprint arXiv:2305.12474*, 2023. URL https://arxiv.org/pdf/2305.12474.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. Agieval: A human-centric benchmark for evaluating foundation models. *arXiv preprint arXiv:2304.06364*, 2023. URL https://arxiv.org/pdf/2304.06364.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, et al. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*, 2022. URL https://arxiv.org/pdf/2210.09261.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. URL https://arxiv.org/pdf/2110.14168.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021. URL https://arxiv.org/pdf/2103.03874.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021. URL https://arxiv.org/pdf/2107.03374.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021. URL https://arxiv.org/pdf/2108.07732.

SAC/TC260. Basic security requirements for generative artificial intelligence service. URL https://mp.weixin.qq.com/s/xz75lAgm_sOGBLBLEFDTnA.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. URL https://arxiv.org/pdf/2001.08361?trk=public_post_comment-text.

OpenAI. Introducing chatgpt. 2022. URL https://openai.com/blog/chatgpt.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-lm: Training multi-billion parameter language models using model parallelism. *CoRR*, abs/1909.08053, 2019. URL http://arxiv.org/abs/1909.08053.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. URL https://arxiv.org/pdf/2303.08774.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models. *CoRR*, abs/2203.15556, 2022. doi:10.48550/ARXIV.2203.15556. URL https://doi.org/10.48550/arXiv.2203.15556.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. Ccnet: Extracting high quality monolingual datasets from web crawl data. *arXiv preprint arXiv:1911.00359*, 2019. URL https://aclanthology.org/2020.lrec-1.494.pdf.

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Kurt Shuster, Eric M Smith, et al. Recipes for building an open-domain chatbot. *arXiv preprint arXiv:2004.13637*, 2020. URL https://arxiv.org/pdf/2004.13637.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020. URL https://arxiv.org/pdf/2101.00027.

Anton Lozhkov, Raymond Li, Loubna Ben Allal, Federico Cassano, Joel Lamy-Poirier, Nouamane Tazi, Ao Tang, Dmytro Pykhtar, Jiawei Liu, Yuxiang Wei, et al. Starcoder 2 and the stack v2: The next generation. *arXiv preprint arXiv:2402.19173*, 2024. URL https://arxiv.org/pdf/2402.19173.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019b. URL https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022b. URL https://jmlr.org/papers/volume23/21-0998/21-0998.pdf.

Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, et al. Glam: Efficient scaling of language models with mixture-of-experts. In

International Conference on Machine Learning, pages 5547–5569. PMLR, 2022. URL https://arxiv.org/pdf/2112.06905.

Damai Dai, Chengqi Deng, Chenggang Zhao, R. X. Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Y. Wu, Zhenda Xie, Y. K. Li, Panpan Huang, Fuli Luo, Chong Ruan, Zhifang Sui, and Wenfeng Liang. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. *CoRR*, abs/2401.06066, 2024. doi:10.48550/ARXIV.2401.06066. URL https://doi.org/10.48550/arXiv.2401.06066.

Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural networks*, 107:3–11, 2018. URL https://arxiv.org/pdf/1702.03118.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. URL https://arxiv.org/pdf/2104.09864.

Ofir Press, Noah A Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation. *arXiv preprint arXiv:2108.12409*, 2021. URL https://arxiv.org/pdf/2108.12409.

A Model Structure Experiment

We conduct experiments on the model structure using a 160M model, which consists of 12 transformer layers, a hidden size of 128, and 12 attention heads. In our setup, we train the models for 100,000 steps with a batch size of 1.3 million tokens, a sequence length of 2048, and a learning rate of 0.0006. The data we use in the following is the same as the data described in Section 2.3.

Activation Function

We test the performance of different activation functions, as shown in Table 8. SwiGLU achieves the lowest perplexity on the development set within the same number of training steps.

Table 8: Perplexity of the models with different activation functions on the dev set.

	20k	**40k**	**60k**	**80k**	**100k**
GeGLU [Shazeer, 2020a]	22.725	21.692	20.813	20.496	20.192
Swish [Shazeer, 2020b]	21.634	20.784	20.040	19.803	19.435
SiLU [Elfwing et al., 2018]	21.680	20.803	20.036	19.776	19.527
SwiGLU [Shazeer, 2020b]	**21.280**	**20.432**	**19.624**	**19.358**	**19.082**

Positional Embedding

We test the performance of different positional embeddings, listed in Table 9 and Figure 5. In our setting, ALiBi converges slightly faster and demonstrates better extrapolation capabilities when context-extending methods are not adopted.

Table 9: Perplexity of the models with different positional embeddings on the dev set.

	20k	**40k**	**60k**	**80k**	**100k**
RoPE [Su et al., 2024]	21.228	20.432	19.624	19.358	19.082
ALiBi [Press et al., 2021]	**20.732**	**20.025**	**16.329**	**19.312**	**18.884**

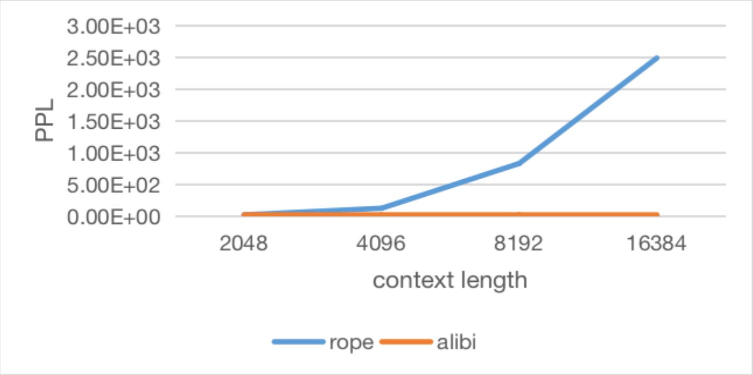

Figure 5: Perplexity of the models with different context lengths