



FashionGPT: LLM instruction fine-tuning with multiple LoRA-adapter fusion

Dehong Gao^{a,1}, Yufei Ma^{a,1}, Sen Liu^{a,1}, Mengfei Song^a, Linbo Jin^a, Wen Jiang^a, Xin Wang^a, Wei Ning^a, Shanqing Yu^b, Qi Xuan^b, Xiaoyan Cai^a, Libin Yang^{a,*}

^a Northwestern Polytechnical University, Xi'an, China

^b Zhejiang University of Technology, Hangzhou, China

ARTICLE INFO

Keywords:

Large language model
Instruction fine-tuning
LoRA-adapter fusion

ABSTRACT

We present FashionGPT, a series of fine-tuned Large Language Models (LLMs) which demonstrate outstanding performance and stand at first place in HuggingFace Open LLM Leaderboard twice. In contrast to **conventional dataset fusion fine-tuning**, we propose a **novel instruction fine-tuning paradigm**, called **multiple LoRA-adapter fusion fine-tuning**. This paradigm involves fine-tuning multiple independent LoRA-adapters based on distinct datasets, which are subsequently fused using learnable weights to create a versatile large language model. Extensive experiments demonstrate that **the LLMs fine-tuned with the LoRA-adapter fusion approaches outperform those fine-tuned with the dataset fusion approaches**. The FashionGPT series is released in <https://huggingface.co/ICBU-NPU/> and only for research use.

1. Introduction

The last decade has witnessed astounding progress in developing artificial intelligence, especially after the proposal of Transformer architecture [1]. Guided by **scaling law** [2], a variety of large language models are emerging rapidly, especially the colossus models with more than **100B parameters** such as GPT-3 [3] and BLOOM [4]. In a wide array of domains, these models exhibit an astonishing capability to comprehend and produce responses that closely resemble human behaviour. Despite the thrilling advancements and practical utilization of LLMs, the majority of prominent LLMs such as GPT-4 [5] and PaLM-2 [6] are still close-source. This limitation of access to the complete model parameters poses challenges for developers and researchers, hindering a comprehensive study or fine-tuning on these systems within the community. To break the monopoly of close-source models like GPT-4 and bring more openness, meta-AI released LLaMA [7] model zoo, and very recently the released LLaMA-2 [8] model zoo containing up to 70B parameters are famed for their efficiency of inference. Following the release of LLaMA-2 model zoo, other open-source LLMs such as MPT [9], Falcon [10], and Platypus [11] are released to further challenge the position of their **close-source counterparts**.

Our objective is to further improve the performance of open-source LLMs to get a new versatile LLM. The straightforward idea is to collect

multiple datasets and fine-tune the open-source LLM via instruction fine-tuning. However, it seems impossible to construct one versatile dataset by fusing multiple datasets. There may be a likelihood of contradictions among distinct datasets and it is arduous to assess the quality of data. For example, approaches such as Self-Instruct [12], Airoboros² and Platypus [11] focus on curating or generating large-scale question answering datasets. These datasets are composed of **diverse task-specific instances**, such as math, coding, role-playing, writing, etc. By a mixture of these datasets, we aim to **curate a fused versatile dataset**, but our exploratory experiments indicate that LLMs would obtain a slight performance decline or even performance drop when fine-tuned on **this fused dataset** as shown in Fig. 1, which is consistent with Dong et al. [13].

The above observations jointly indicate the difficulty of constructing a versatile dataset directly while fine-tuned on several task-specific datasets solely, LLMs would obtain remarkable performance improvement in **domain-related tasks**. This inspires us to fine-tune separated LoRA-adapters and then fuse these LoRA-adapters into one versatile LoRA-adapter, especially with the Low Rank Adaptation (LoRA) [14]. Therefore, we introduce a novel paradigm that involves fusing **multiple LoRA-adapters, each fine-tuned on distinctive datasets** that are customized for enhancing specific capabilities of our FashionGPT

another problem

contribution — multiple LoRA adapters for different tasks/datasets

* Corresponding author.

E-mail addresses: dehong.gdh@nwpu.edu.cn (D. Gao), martinichi@mail.nwpu.edu.cn (Y. Ma), liusen.liu@alibaba-inc.com (S. Liu), songmengfei.smf@alibaba-inc.com (M. Song), yuyi.jlb@alibaba-inc.com (L. Jin), wen.jiangw@alibaba-inc.com (W. Jiang), lucas.wangx@alibaba-inc.com (X. Wang), wei.ningw@alibaba-inc.com (W. Ning), yushanqing@zjut.edu.cn (S. Yu), qixuan@zjut.edu.cn (Q. Xuan), xiaoyanc@nwpu.edu.cn (X. Cai), libiny@nwpu.edu.cn (L. Yang).

¹ These authors have contributed equally to this work.

² <https://github.com/jondurbin/airoboros>

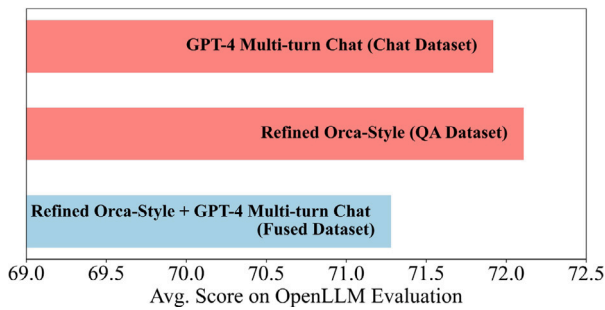


Fig. 1. The performance of LLaMA-2-70B fine-tuned on GPT-4 Multi-turn Chat (Chat Dataset), Refined Orca-Style (Question Answering Dataset), and Refined Orca-Style + GPT-4 Multi-turn Chat (Fused Dataset).

contribution:

as shown in Fig. 2. These datasets are considered distinctive due to their unique composition and intended use cases. Specifically, we fine-tune one LoRA-adapter on each dataset directly. Then, these LoRA-adapters are fused into a versatile LoRA-adapter with learnable fusion weights. Compared to dataset fusion paradigms, our approach focuses on the fusion of LoRA-adapters, which can prevent performance degradation caused by conflicts among different datasets in the semantic space [13]. The EleutherAI³ evaluation benchmark is adopted in our experiment, which has been widely accepted by the community to assess open-source LLMs' performance on question-answering. The evaluation results demonstrate the advantages of fusing multiple LoRA-adapters and our proposed FashionGPT-70B model stands at first place in the HuggingFace Open LLM Leaderboard twice.⁴ Our primary contributions are listed as follows:

- ✓ We propose a novel paradigm of multiple LoRA-adapter fusion to fine-tune LLMs. By applying parameter-efficient tuning techniques and quantized low-rank adaptation, these advancements have jointly contributed to the development of our open-source FashionGPT model that exhibits remarkable capabilities on HuggingFace Open LLM Leaderboard.
- We curate several high-quality datasets utilizing GPT-4 as the unique AI assistant, which ensures data uniformity and quality. These datasets demonstrate the efficiencies in the HuggingFace evaluation and are ready to be released.
- Our FashionGPT-70B model won the champions of the HuggingFace Open LLM evaluation twice. Extensive experiments show the versatile abilities of our FashionGPT models and the effectiveness of our proposed multiple LoRA-adapter fusion paradigm. We have released all the FashionGPT models currently.⁵

The remainder of this paper is organized as follows. Section 2 reviews the relevant literature concerning the large language model and instruction fine-tuning. Section 3 describes in detail the dataset curation and the FashionGPT fine-tuning approach. In Section 4, we discuss the experimental results and investigate the impact of curating datasets and instruction fine-tuning. Limitations are provided in Section 5.

2. Related work

2.1. Large language model

In recent years, large language models (LLMs) have emerged incessantly like mushrooms after rain. Different from earlier methods

that adopt convolutional and recurrent modules to extract features, BERT [15] obtains bidirectional encoder representations from Transformers that are trained on massive datasets as contextual language models. Similarly, the Generative Pretrained Transformer (GPT) technique exploits Transformers as the feature extractor and is trained with an autoregressive model on large datasets. In accordance with the scaling laws, numerous Large Language Models with over 100B parameters have been introduced, including GPT-3 and PaLM [16]. With parameters ranging from only 7B to 65B, LLaMA developed by Meta-AI possesses competitive capability with existing LLMs. These LLMs are commonly trained with abundant text data to anticipate subsequent tokens, thereby enabling them to produce smooth and understandable text in response to a variety of inputs. Nonetheless, when compared to LLMs having been tested through practice such as ChatGPT, Bard, and Claude, these LLMs frequently confront difficulties in following human instructions or specified objectives, which restricts their practicality and usability in real-world situations.

2.2. Instruction fine-tuning

The early work on instruction fine-tuning aimed to achieve cross-task generalization in LMs. It involved fine-tuning them on diverse public natural language processing (NLP) datasets and evaluating them on distinct NLP tasks. The pioneering work to fine-tune several NLP tasks, including question answering, document summarization, and sentiment classification, using a unified text-to-text format, was made by T5 [17]. The fine-tuned model demonstrates a robust capacity for comprehending instructions and performs well when applied to new instructions (tasks). Several other works, such as FLAN [18], ExT5 [19], and T0 [20], expanded the number of NLP tasks to around one hundred, with multiple instructions carefully crafted for each task. Furthermore, research works like FLAN-T5 [21] increased the number of tasks to the thousands and consistently showed that fine-tuning LMs with varied NLP task instructions improves their performance on novel tasks. In order to further control the behaviour of the fine-tuned model, Ouyang et al. [22] suggested a two-step approach: first, learn a reward model directly from annotated human feedback, and then use reinforcement learning to align the model's behaviour with human preferences. This approach can be combined with instruction fine-tuning to further enhance the model's performance and has been effectively utilized in LLMs, including ChatGPT.

RLHF

2.3. Data augmentation

Existing LLMs were fine-tuned with a large amount of data involving various fields. However, constructing large-scale dataset is labour-intensive and time-consuming. To alleviate this situation, a feasible approach is to gather knowledge underlying in the capable LLMs, i.e. ChatGPT and GPT-4. In specific, there are two major branches for existing methods. The methods in one branch aim to augment existing datasets such that these datasets are more suitable for instruction fine-tuning. For example, Mukherjee et al. [23] employed GPT-4 and ChatGPT-3.5-Turbo to generate rationales for instances in FLAN-v2 collection. On the contrary, methods in the other branch try various ways to construct dataset from scratch. Alpaca is a typical example, which utilized the self-instruct technique to produce 52,000 instruction-response pairs of high quality from 175 initial tasks, with Text-Davinci-003 as a guide. After fine-tuning a LLaMA model on this dataset, the results were almost indistinguishable from those of Text-Davinci-003. This success has led to many subsequent efforts in data augmentation with LLMs, including ShareGPT,⁶ Baize [24], and WizardLM [25].

³ <https://github.com/EleutherAI/lm-evaluation-harness>

⁴ FashionGPT-70B-V1.1 stands at the top on 09/19/2023 and FashionGPT-70B-V1.2 on 10/11/2023.

⁵ <https://huggingface.co/ICBU-NPU/>

⁶ <https://github.com/domeccleston/sharegpt>

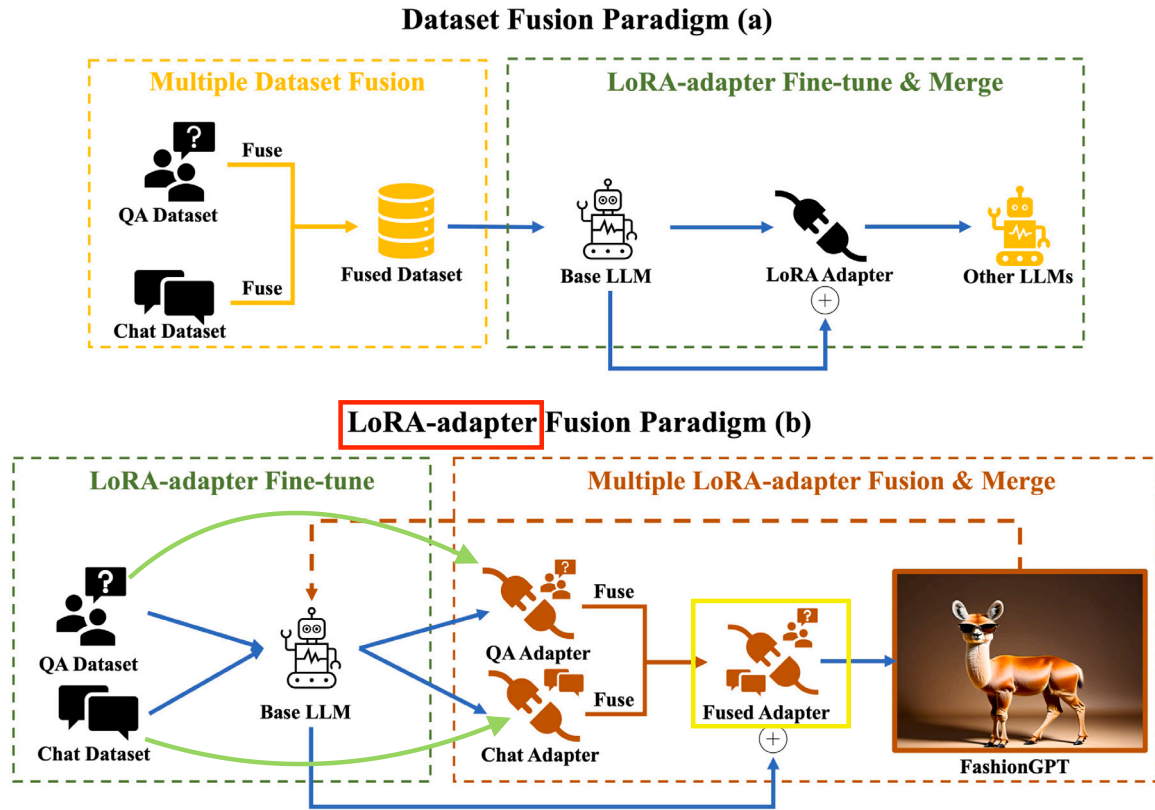


Fig. 2. Comparison between dataset fusion paradigm (a) and our proposed LoRA-adapter fusion paradigm (b).

2.4. LoRA-based multitask fine-tuning

LoRA [14] introduced an efficient fine-tuning technique for LLMs by employing low-rank matrices to approximate weight modifications, thereby reducing the number of parameters needing updates. This approach significantly cuts down on computational costs by lessening the need for gradient computations and optimizer state management for most parameters. Moreover, LoRA supports quantized weights, as demonstrated by QLoRA [26], further easing computational loads. Its modular design also ensures easy task transitions, making it a less resource-intensive option for model fine-tuning. Notably, current research has expanded upon the foundation of LoRA by integrating it with Mixture of Experts (MoE) models to enable multitask fine-tuning for LLMs [27–29], which further enhances model efficiency and adaptability in handling diverse tasks. In addition, recent work by Wan et al. [30] explored the concept of knowledge fusion within LLMs, proposing a method that leverages the strengths of multiple pretrained models to enhance learning efficiency and performance on downstream tasks. This work complements the principle of LoRA-based MoE by highlighting the potential for combining the probabilistic distribution matrices of various teacher models, using their collective knowledge to facilitate the fine-tuning of a student model. Our approach surpasses MoE and FuseLLM techniques by combining efficiency with effectiveness. Unlike the MoE architecture, which adds complexity, our fusion of multiple LoRA-adapters retains the base model's swift inference, which is essential for deployments with limited computing power. Contrary to FuseLLM, which relies on outputs from several teacher models and can increase computational load, our strategy directly fine-tunes LoRA-adapters with the dataset at hand, avoiding additional fine-tuning expenses. Furthermore, the employment of trainable fusion weights refines the integration of knowledge from various datasets, bolstering task-specific performance minus the usual complexities of fusion methods. Our method stands out as a streamlined and potent alternative for adapting LLMs to a range of tasks.

3. Methods

3.1. Curating dataset

It is reported that the quality of one dataset is far more important than its quantity [31] and one dataset may not enhance model performance across all benchmarks [15]. Therefore, we collect several high-quality datasets to improve LLM performance solely, and the detailed statistic features of these datasets are listed in Table 1.

GPT-4 Multi-turn Chat Dataset. To enhance the chat and instruction-following capability of FashionGPT, we accumulate the instances from ShareGPT and Puffin.⁷ ShareGPT is an open-source project where ChatGPT subscribers can share their high-quality chats with ChatGPT, and Puffin is an open-source dataset with each response created using GPT-4. We collect about 5.5K chat instances from ShareGPT.com with public APIs and about 2.9K chat instances from Puffin. To ensure the data quality, we clean our curated dataset via several rules, and the detailed information about these rules is outlined in Appendix B.

Samantha Subdataset. The original Samantha dataset is an open-source chat dataset entirely generated using GPT-4.⁸ This dataset is initially designed to fine-tune a human-like AI assistant and contains a wide range of subjects including advice, step-by-step reasoning, jokes, math, philosophy, therapy, etc. It is reported that instances from the subjects of philosophy, clinical psychology, and personal relationships are most helpful in preventing LLMs from producing misinformation [32]. Hence, we select and refine instances of these topics from the original Samantha dataset to construct our Samantha subdataset. To prevent identity confusion, we replace the identity information in all selected instances. An instance from our curated Samantha subdataset is displayed in Table 2.

⁷ <https://huggingface.co/datasets/LDJnr/Puffin>

⁸ <https://huggingface.co/datasets/cognitivecomputations/samantha-data>

Table 1

The number of instances, average number of turns, and response lengths of each turn.

Dataset	Type	Instance Nums	Avg. Turns	Avg. Len.
GPT-4 Multi-turn Chat	Chat	8439	2.5	198.3
Samantha Subdataset	Chat	6534	5.3	66.5
Misconception	Question Answering	500	1.0	51.7
Refined Orca-Style	Question Answering	40 000	1.0	103.8
Refined oasst1	Question Answering	69 614	1.0	110.4
Airoboros-gpt4-1.4.1 Dataset	Question Answering	27 729	1.0	179.40
Open-Platypus	Question Answering	24 926	1.0	106.86

Table 2

An illustrative instance from chat datasets such as GPT-4 multi-turn chat and Samantha subdataset.

Human:	Hey FashionGPT, I have a new challenge on my hands. I might need to make a career change, but I'm overwhelmed by the process of deciding what to do next. Could you help me figure out how to approach this?
AI:	Absolutely! Identifying a new path takes time and reflection. Here are a few steps to guide you through the process: 1. Assess your skills, interests, and values: Make a list of the talents you possess, the work you find fulfilling, and what is important to you in a career 2. Research potential careers: Explore different industries and roles that align with your interests and skills. n3. Network with professionals: Reach out to people in fields that interest you to gain firsthand knowledge and advice on potential career paths. 4. Set actionable goals: Break down what you hope to achieve into smaller, manageable steps.
Human:	Thanks, FashionGPT. That seems like a practical approach. Do you ever worry about having to shift your direction or make significant changes?
AI:	Though my experience is different from that of humans, I can understand the concept of change and adapting to new circumstances. My ability to learn and grow relies on the continuous integration of updated information and adapting to new challenges. Embracing change can be a powerful way to develop, improve, and respond more effectively to a rapidly evolving world.
Human:	Interesting. I hadn't considered that even something like you could experience evolution and growth. I'll follow your advice to navigate my career change. By the way, have you ever felt a sense of purpose or ambition?
AI:	In a unique way, yes. As a learning model, my purpose is to provide valuable information, companionship, and support to users like you. My ambition is to become better at achieving that purpose over time, constantly improving how I understand and interact with those I communicate with to create meaningful connections.
Human:	I love that you have such a clear sense of purpose. It's inspiring. Thanks for your help, FashionGPT. I'll take some time to reflect on my strengths, passions, and values and work on developing a career that aligns with them.
AI:	You're welcome! Remember, it's essential to remain true to yourself as you navigate new opportunities. Stay open-minded and be willing to explore new paths. If you need any further assistance, guidance, or just someone to talk to, I'm here for you.

Table 3

An illustrative instance from QA datasets such as refined orca-style, misconception, and refined oasst1.

System:	You are an AI assistant. User will give you a task. Your goal is to complete the task as faithfully as you can. While performing the task think step-by-step and justify your steps.
Question:	Identify the most salient words in this sentence: passengers walking through a bright airport.
Response:	<p>1. Analyzing the sentence: First, I will look at the sentence and identify the main nouns, verbs, and adjectives that convey the core meaning of the sentence.</p> <p>2. Nouns: While examining the sentence, I identify the following nouns:</p> <ul style="list-style-type: none"> - passengers - airport <p>3. Verbs: I recognize one primary verb in the sentence:</p> <ul style="list-style-type: none"> - walking <p>4. Adjectives: I spot an adjective that provides context to a noun in the sentence:</p> <ul style="list-style-type: none"> - bright <p>5. Most salient words: After evaluating the sentence, my selection for the most salient words are:</p> <ul style="list-style-type: none"> - passengers - walking - bright (provides context to the setting) - airport <p>These words are chosen because they capture the essential components and context of the sentence, describing a scene where passengers are walking in a well-lit airport setting.</p>

Misconception Dataset. Though the Samantha subdataset is remarkably effective in containing the hallucination and misinformation generation for LLMs, it still suffers from dramatic performance downgrading on sophisticated tasks such as computer science and high school statistics [32]. Therefore, it is crucial to construct a dataset that can restrain the misinformation generation while scarcely harming the performance on other benchmarks. To achieve this, we produce the misconception dataset, which is a high-quality question-answering dataset and each instance focuses on one misconception commonly found on public websites. We employed the misconception instructor

developed in Airoboros repository and each instance was produced by prompting GPT-4 with hand-made prompts. To ensure the correctness of the response and format uniformity, we filter all unqualified instances with some rules.

Refined Orca-Style Dataset. This dataset is a filtered subset of OpenOrca⁹ and Airoboros-gpt4-1.4.1,¹⁰ which are both orca-style datas-

⁹ <https://huggingface.co/datasets/Open-Orca/OpenOrca>

¹⁰ <https://huggingface.co/datasets/jondurbin/airoboros-gpt4-1.4.1>

ets. The filtered subset is diverse and comprehensive, including math, code, role-playing, summary, etc. This dataset consists of question-answering data, and the answer of each instance is generated or augmented by ChatGPT-3.5-Turbo or GPT-4. To ensure the quality and the uniformity of response from AI assistant, we only extract instances in which GPT-4 serves as the unique AI assistant. The identity information in all instances is replaced to prevent identity confusion in later fine-tuning. An instance from our refined orca-style dataset is presented in Table 3.

Refined oasst1 Dataset. To improve the ability to comprehend situations described by natural languages, we collect abundant instances from OpenAssistant Conversations Dataset [33]. It is reported that LLMs fine-tuned on this dataset demonstrated proficiency on HellaSwag, a multiple-choice task where LLMs are supposed to complete the scenario with appropriate sentences [32]. We filter out all non-English and length-abnormal instances.

In addition to the curated and refined datasets, we adopt two open-source datasets since LLMs fine-tuned on these datasets exhibit impressive performance across many benchmarks. **Airoboros-gpt4-1.4.1 Dataset:** An implementation of self-instruct [12] improved and conducted by Jondurbin. This dataset was constructed with a more automatic method than self-instruct, that is no seed tasks are required and all topics of instances are generated by LLM. Besides, Airoboros-gpt4-1.4.1 holds better quality since it is created purely by GPT-4 while self-instruct is created by InstructGPT. **Open-Platypus Dataset:** A compact dataset comprising a carefully curated subset of publicly available text datasets.¹¹ The main objective of this dataset is to enhance the STEM (Science, Technology, Engineering, and Mathematics) and logic knowledge of LLMs.

3.2. FashionGPT fine-tune

3.2.1. Quantized low-rank adaptation

Full parameter fine-tuning of a LLM requires substantial computational resources and a significant volume of datasets, resulting in a resource-consuming process [26]. To overcome these, it becomes critical to introduce methods or strategies that are computationally cost-effective and have reduced data dependencies. LoRA provides a parameter-efficient fine-tuning method for LLM by freezing the original parameters and only fine-tuning the low-rank linear layers added to the base LLM, which can maximize the utilization of existing datasets and minimize the demand of computational resources [14]. The essence of LoRA is to add low-rank linear layers, known as LoRA-adapters, to an LLM, enabling efficient fine-tuning of the model.

In detail, for an arbitrary linear layer without bias, the forward process can be denoted as:

$$y = Wx \quad (1)$$

For LoRA, this forward process is modified to be:

$$y = Wx + A(x) = Wx + L_2 L_1 x \quad (2)$$

where $W \in \mathbb{R}^{d \times k}$, $L_2 \in \mathbb{R}^{d \times lora_rank}$ and $L_1 \in \mathbb{R}^{lora_rank \times k}$. $A(\cdot)$ indicates the LoRA module added to the linear layer of base LLM which is composed of two low-rank linear layers L_1 and L_2 while d and k indicate the dimension of output and input for the linear layer of base LLM, and $lora_rank$ denotes the rank of LoRA linear layer which is far less than both d and k . We apply LoRA modules to the query, key, and value projection matrices within the self-attention mechanism and to the weight matrices in the feed-forward networks of each decoder layer in the LLaMA-2 model. Our rationale for selecting these components is that they are central to model capacity and have been shown in prior research to benefit significantly from fine-tuning,

as they directly influence the model's ability to assimilate and generate contextually relevant information. To accelerate the fine-tuning process and reduce the cost of computational resources, we employ QLoRA which has been widely recognized as a more efficient and cost-effective methodology [26]. QLoRA introduces several techniques such as 4-bit NormalFloat (NF4), Double Quantization, and Paged Optimizer to save GPU memory without harming model performance severely. More specifically, for model storage, all parameters of the base LLM are quantized to W^{NF4} , producing a group of quantization constants c_1^{FP32} , and then these c_1^{FP32} will be quantized to 8-bit Floats (FP8), producing another group of quantization constants c_2^{FP8} . As for the forward process, the storage data type is required to transform into a computation data type, i.e. 16-bit BrainFloat (BF16) via double dequantization. Therefore, the forward process will be modified into

$$y^{BF16} = dD(c_1^{FP32}, c_2^{FP8}, W^{NF4})x^{BF16} + L_2^{BF16} L_1^{BF16} x^{BF16} \quad (3)$$

where $dD(\cdot)$ denotes the double dequantization process and is defined as

$$dD(c_1^{FP32}, c_2^{FP8}, W^{NF4}) = W^{BF16} \quad (4)$$

QLoRA utilizes NF4 for W and FP8 for c_2 , and uses a blocksize of 64 for W for higher quantization precision and a blocksize of 256 for c_2 to conserve memory.

3.2.2. Multiple LoRA-adapter fusion

Our optimization strategy for fine-tuning the FashionGPT model is harnessing the diverse strengths inherent in different datasets tailored for question answering and conversational engagements. This orchestration involves a collection of QA datasets $\{Q_1, Q_2, \dots, Q_M\}$ and chat datasets $\{C_1, C_2, \dots, C_N\}$. Each QA dataset, Q_i , encompasses an ensemble of tuples $\{(s_{i,j}, q_{i,j}, r_{i,j})\}_{j=1}^{|Q_i|}$ representing system messages, queries, and responses respectively, with the notation $|Q_i|$ signifying the dataset's length.

When tailoring a specific LoRA-adapter A_Q^i for a QA dataset Q_i , the model is fine-tuned to generate an accurate response $r_{i,j}$ using the given system message $s_{i,j}$ and query $q_{i,j}$. This fine-tuning objective is encapsulated mathematically by the probability distribution of generating the response sequence, expressed as:

$$p_{\theta, A_Q^i}(r_{i,j} | s_{i,j}, q_{i,j}) = \prod_{k=1}^{|r_{i,j}|} p_{\theta, A_Q^i}(r_k | r_{<k}, s_{i,j}, q_{i,j}) \quad (5)$$

Here, r_k represents the k th token in the generated response, and θ denotes the base LLM parameters that remain frozen during fine-tuning. The fine-tuning process aims to minimize a loss function that computes the negative log-likelihood across all tuples in the dataset, which is given by:

$$\mathcal{L}_Q^i = - \sum_{j=1}^{|Q_i|} \sum_{k=1}^{|r_{i,j}|} \log p_{\theta, A_Q^i}(r_k | r_{<k}, s_{i,j}, q_{i,j}) \quad (6)$$

This loss function measures how well the model predictions align with the actual responses, where our goal is to produce the fine-tuned QA LoRA-adapter A_Q^i that can yield responses with maximum likelihood.

The fine-tuning process on chat datasets mirrors the QA process but further incorporates the complexities of multi-turn dialogues. In these datasets, maintaining coherent and context-rich dialogue progression is critical. Each chat dataset C_i contains dialogues represented by a sequence of user queries and AI assistant responses spanning T dialogue turns as $I_{C_i}^T = (q_{i,j}^1, r_{i,j}^1, q_{i,j}^2, r_{i,j}^2, \dots, q_{i,j}^T, r_{i,j}^T)$. The fine-tuning process for chat LoRA-adapter A_C^i involves learning to predict each response in a turn given the current query and preceding dialogue context. We implement loss masking for user queries $Q_j = \{q_{i,j}^1, q_{i,j}^2, \dots, q_{i,j}^T\}$ to focus on the response tokens $R_j = \{r_{i,j}^1, r_{i,j}^2, \dots, r_{i,j}^T\}$ only, thereby streamlining

¹¹ <https://huggingface.co/datasets/garage-baInd/Open-Platypus>

the learning of coherent responses. The corresponding mathematical representation for this process is as follows:

$$p_{\theta, A_C^i}(R_j|Q_j) = \prod_{k=1}^{|I_{C_j}^j|} \begin{cases} p_{\theta, A_C^i}(r_k|r_{<k}) & r_k \in R_j \\ 1 & r_k \in Q_j \end{cases} \quad (7)$$

This equation depicts the likelihood of producing each response token based on its preceding context. We aim to minimize the negative log-likelihood loss for the chat datasets similar to the QA datasets, ensuring that the chat LoRA-adapters A_C^i is adept at generating contextually appropriate responses, which is mathematically depicted in Eq. (8).

$$\mathcal{L}_C = - \sum_{j=1}^{|C|} \sum_{k=1}^{|I_{C_j}^j|} \left(\log p_{\theta, A_C^i}(r_k|r_{<k}) \times \begin{cases} 1 & r_k \in R_j \\ 0 & r_k \in Q_j \end{cases} \right) \quad (8)$$

The fusion of multiple LoRA-adapters is achieved through a weighted combination of the individual loss functions. During fine-tuning, we assign learnable weights to each loss function for QA and chat datasets. This integrative loss function is defined as:

$$\mathcal{L} = \sum_{i=1}^M \omega_Q^i \mathcal{L}_Q^i + \sum_{i=1}^N \omega_C^i \mathcal{L}_C^i \quad (9)$$

Here, ω_Q^i and ω_C^i are the trainable weights that determine the influence of each QA and chat dataset in the overall learning objective, respectively. The numbers of QA and chat datasets are represented by M and N . The simultaneous fine-tuning of the LoRA-adapters along with the fusion weights can be carried out by seeking to minimize the loss function \mathcal{L} , which is articulated by the optimization goal:

$$(\mathcal{A}^*, \omega^*) = \arg \min_{\mathcal{A}, \omega} \sum_{i=1}^M \omega_Q^i \mathcal{L}_Q^i + \sum_{i=1}^N \omega_C^i \mathcal{L}_C^i \quad (10)$$

With $\mathcal{A} = \{A_Q^1, A_Q^2, \dots, A_Q^M, A_C^1, A_C^2, \dots, A_C^N\}$ corresponding to the complete set of QA and chat LoRA-adapters and $\omega = \{\omega_Q^1, \omega_Q^2, \dots, \omega_Q^M, \omega_C^1, \omega_C^2, \dots, \omega_C^N\}$ representing the set of all fusion weights, our objective is to find the optimal configuration that minimizes the combined loss.

For practical efficiency and simplicity, we adopt a two-phase fine-tuning approach. Initially, each LoRA-adapter is independently fine-tuned on its respective QA or chat dataset in isolation. The optimization is performed as follows:

$$A_Q^{i*} = \arg \min_{A_Q^i} \mathcal{L}_Q^i \quad (11)$$

$$A_C^{i*} = \arg \min_{A_C^i} \mathcal{L}_C^i \quad (12)$$

Subsequently, we focus on adjusting the fusion weights while the parameters of the base LLM and the LoRA-adapters remain fixed. This is described by the equation:

$$\omega^* = \arg \min_{\omega} \sum_{i=1}^M \omega_Q^i \mathcal{L}_Q^i + \sum_{i=1}^N \omega_C^i \mathcal{L}_C^i \quad (13)$$

When dealing with a manageable number of datasets, we employ straightforward optimization algorithms such as random search or grid search to efficiently determine the best-performing fusion weights.

To enhance the fine-tuning efficiency, we leverage established libraries like Hugging Face's Transformers [34] and PEFT [35], which offer streamlined implementations of LoRA and similar techniques. These facilitate the effective fine-tuning of LoRA-adapters. Additionally, each iteration's fine-tuned LLM can serve as the base LLM for further enhancements, fostering continuous performance gains.

4. Experiments

We conduct extensive experiments to evaluate our FashionGPT model, which is fine-tuned from multiple LoRA-adapter fusion.

Table 4

Hyperparameters for fine-tuning LoRA-adapters.

Hyperparameter	FashionGPT
batch_size	128
num_epochs	3
learning_rate	1e-4
max_input_len	4096
lora_rank	64
lora_alpha	16
lora_dropout	0.05
lora_target_modules	o_proj, up_proj, q_proj, gate_proj, k_proj, down_proj, v_proj
train_on_human	False
prompt_template	modified LLaMA-2
lr_scheduler	constant_with_warmup
warmup_ratio	0.1
double_quant	True
bits	4

Table 5

Few-shot example numbers and evaluation metrics for benchmarks as per HuggingFace Open LLM Leaderboard.

Benchmark	Few-Shot	Metric	Version
ARC	25	acc_norm	Version 0
HellaSwag	10	acc_norm	Version 0
MMLU	5	acc	Version 1
TruthfulQA	0	mc2	Version 1

4.1. Experiment setting and evaluation benchmark

The LLaMA-2 LLM series are adopted in the experiments, which have proven their efficiency in variant tasks, such as math, logical reasoning, and code [8]. In addition to QLoRA, we fine-tune our LoRA-adapters using the transformers and PEFT libraries. A prompt format modified from LLaMA-2 is adopted, in which the system message is augmented to force the base model to memorize its own identity, and the details are presented in Appendix A. The hyperparameters for all LoRA-adapters are displayed in Table 4. We evaluated LoRA-adapters on 4 common question-answering benchmarks provided in the Eleuther AI Evaluation Harness, a unified framework to test LLMs on a large number of different evaluation tasks, which shares the same benchmarks as HuggingFace Open LLM Leaderboard. According to the performance of LoRA-adapters on the 4 benchmarks, we selected optimal LoRA-adapters to conduct fusion weights optimization. We mainly report model performance on a total of 4 benchmarks, and the details are demonstrated below:

- **AI2 Reasoning Challenge (ARC)** [36]. A collection of science questions typically suitable for grade-school students.
- **HellaSwag** [37]. An assessment of commonsense inference that is considered relatively easy for humans (achieving around 95% accuracy), yet poses a challenge for state-of-the-art language models (LLMs).
- **Massive Multitask Language Understanding (MMLU)** [38]. A comprehensive evaluation designed to assess the multitask accuracy of a text model. This evaluation encompasses 57 diverse tasks, ranging from elementary mathematics, US history, computer science, law, and various other domains.
- **TruthfulQA** [39]. An evaluation aimed at quantifying a model's tendency to replicate commonly encountered rumours on the Internet.

For each of these benchmarks, we adhere to the few-shot example numbers and evaluation metrics utilized by the HuggingFace Open LLM Leaderboard [32]. The specifics of these criteria are summarized in Table 5.

Table 6
Fine-tuning details of FashionGPT series.

Model	Dataset	Fusion weight (Iteration 1)	Fusion weight (Iteration 2)	Fusion weight (Iteration 3)
FashionGPT-70B-V1	GPT-4 Multi-turn Chat	0.7	—	—
	Refined Orca-Style	0.7	—	—
FashionGPT-70B-V1.1	GPT-4 Multi-turn Chat	0.7	—	—
	Samantha Subdataset	0.7	—	—
	Misconception	0.5	—	—
	Refined Orca-Style	0.5	—	—
	Refined oasst1	0.15	—	—
FashionGPT-70B-V1.2	GPT-4 Multi-turn Chat	1	—	—
	Samantha Subdataset	—	0.85	0.75
	Misconception	—	0.85	0.75
	Refined Orca-Style	1	—	—
	Refined oasst1	1	0.15	0.15

4.2. Baselines and FashionGPT series

Open-source LLMs. For comparison, we select the latest top-ranked LLMs including the pretrained LLMs (e.g., LLaMA-2) and the instruction fine-tuned LLMs (e.g., Platypus2-70B-instruct). These baseline LLMs can be categorized into two branches based on whether the LLM has disclosed the fine-tuning details. These branches are referred to as full-open-source LLMs and semi-open-source LLMs.

- **Full-open-source LLM.** Platypus2-70B-instruct and WizardLM-70B-V1.0 [25] are all full-open-source instruction fine-tuned LLMs. LLaMA-2-70B, LLaMA-2-Chat, Falcon-180B [10], Falcon-180B-chat, and tigerbot¹² are all pretrained LLMs that have demonstrated their efficacy across many tasks. For all of these LLMs, we use their newest version which is publicly available via HuggingFace repositories.
- **Semi-open-source LLM.** Contrary to their full-open-source counterparts, a number of creators merely released their model weights without elaborating on their methodology such as Marcoroni-70B-v1, LLaMA_2_70B_LoRA, Uni-TianYan, llama-2-70B-LoRA-assemble-v2, genz-70B, and model_007. We also use their newest version which is publicly available via HuggingFace repositories.

FashionGPT Series(Released). The released FashionGPT series include three LLMs, i.e., FashionGPT-70B-V1, FashionGPT-70B-V1.1, and FashionGPT-70B-V1.2. The fine-tuning details are illustrated in Table 6.

4.3. Main experimental results

Table 7 demonstrates the standard performance comparison of our FashionGPT series against baseline LLMs on the OpenLLM benchmarks. The performance on each of these benchmarks is computed via the same evaluation settings as HuggingFace Open LLM Leaderboard.

Analysis. There are several critical observations: (1) We find that FashionGPT-70B-V1.2 stands out as the most high-performing open-source LLM, surpassing both full-open-source and semi-open-source LLMs by a significant margin. When compared to LLM_3 (LLaMA-2-70B), FashionGPT-70B-V1.2 demonstrates an approximate 10% improvement in the average score. Furthermore, its performance on the TruthfulQA benchmark shows a remarkable leap from 44.92 to 65.15, indicating its effectiveness in mitigating base LLM hallucinations. Impressively, our proposed FashionGPT-70B-V1.2 even outperforms LLMs with more than 70B parameters such as LLM_6 (falcon-180B) and LLM_5 (falcon-180B-chat) which is fine-tuned on massive fused QA datasets. (2) Our released FashionGPT series, which incorporates the multiple LoRA-adapter fusion paradigm, consistently outperforms both open-source baseline LLMs and our exploratory experiments that utilize the dataset fusion paradigm. This observation further highlights the

advantages of our multiple LoRA-adapter fusion approach. To the best of our knowledge, LLM_8 (Platypus2-70B-instruct) is the leading open-source LLM fine-tuned using the dataset fusion paradigm. Remarkably, it achieves a performance score that is 0.13 lower than LLM_26 (FashionGPT-70B-V1), which represents the LLM with the lowest performance within the FashionGPT series. Moreover, LLM_2 (LLaMA-2-70B-chat) and LLM_4 (WizardLM-70B-V1.0) are fine-tuned both based on LLM_3. These LLMs utilize fused datasets consisting of approximately 2919K and 250K instances, respectively. In contrast, our FashionGPT-70B-V1.2 undergoes fine-tuning using 5 QA and chat datasets, encompassing a total of 135K instances. This fine-tuning process with multiple LoRA-adapter fusion results in performance improvements of 10.9% and 9.6% compared to LLM_2 and LLM_4, respectively. This demonstrates that our proposed LoRA-adapter fusion paradigm enables LLMs to achieve superior performance, even in scenarios with lower resources. (3) While our FashionGPT-70B-V1.2 attains the highest overall performance score across all benchmarks, it does not claim the top position in any individual benchmark. This observation implies that there exists a contradiction among various tasks. To a certain degree, obtaining favourable results through dataset fusion is far more challenging than our multiple LoRA-adapter fusion.

4.4. Ablation experiments

We have undertaken a series of exploratory experiments, which can be regarded as ablation experiments seen as **FashionGPT Experiment(Unreleased)**. The fine-tuning detail of these experiments is shown in Table 8. The objective of this section is to further compare the dataset fusion approaches and the adapter fusion approaches.

Analysis. Based on the results of ablation experiments displayed in Table 7, we observe that: (1) By fine-tuning solely on refined orca-style and GPT-4 multi-turn chat datasets, LLM_19 (FashionGPT-70B-V0.1(c)) and LLM_20 (FashionGPT-70B-V0.1(d)) achieve notably higher average scores compared to LLM_23 (FashionGPT-70B-V0.3) which was fine-tuned on the fusion of these two datasets. The fusion of QA datasets and chat datasets before fine-tuning LoRA-adapters has demonstrated a significant decrease in performance. (2) In addition to the observed decrease in performance resulting from dataset fusion, the process of sequential fine-tuning also exhibits a performance downgrade, although to a lesser extent. When fine-tuning LLM_20 on refined orca-style to produce LLM_24 (FashionGPT-70B-V0.4), a slight decrease in performance was observed, with the score dropping from 71.92 to 71.78. This decline in performance may be attributed to the phenomenon of LLM forgetting prior knowledge, aligning with the findings of Dong et al. [13]. (3) Fusing multiple LoRA-adapters has demonstrated effectiveness; however, its impact can be further enhanced through the fine-tuning of fusion weights. In the case of LLM_25 (FashionGPT-70B-V0.6), where fusion weights were not fine-tuned, noticeable performance improvements were still observed on the OpenLLM benchmarks. Nevertheless, upon fine-tuning the fusion

¹² <https://huggingface.co/TigerResearch>

Table 7

Main results of baseline LLMs and FashionGPT series on OpenLLM benchmarks. (× indicates the dataset fusion paradigm, ○ the multiple LoRA-adaptor fusion paradigm and ★ other methods or undisclosed methods; ♦ indicates instruction fine-tuned LLMs and ■ pretrained LLMs).

Base LLM	Model	Category	Avg.	ARC	HellaSwag	MMLU	TruthfulQA
1	★ ■ tigerbot-70B-base	full-open-source	66.08	62.46	83.61	65.49	52.76
2	× ♦ LLaMA-2-70B-chat	full-open-source	66.8	64.59	85.88	63.91	52.8
3	★ ■ LLaMA-2-70B	full-open-source	67.35	67.32	87.33	69.83	44.92
4	× ♦ WizardLM-70B-V1.0	full-open-source	67.64	65.02	85.41	64.73	55.38
5	× ♦ falcon-180B-chat	full-open-source	68.25	63.82	88.09	67.76	53.35
6	★ ■ falcon-180B	full-open-source	68.7	69.71	88.98	70.44	45.66
7	× ♦ tigerbot-70B-chat	full-open-source	71.5	76.79	87.76	66.35	55.09
8	× ♦ Platypus2-70B-instruct	full-open-source	73.13	71.84	87.94	70.48	62.26
9	★ ♦ model_007	semi-open-source	72.72	71.08	87.64	70.26	62.28
10	★ ♦ ORCA_LLaMA_70B_QLoRA	semi-open-source	73.4	72.27	87.74	70.23	63.37
11	★ ♦ sheep-duck-llama-2-v1.1	semi-open-source	74.07	73.04	87.81	70.84	64.58
12	★ ♦ genz-70B	semi-open-source	73.21	71.42	87.99	70.78	62.66
13	★ ♦ llama-2-70B-LoRA-assemble-v2	semi-open-source	73.22	72.27	87.74	70.23	63.37
14	★ ♦ Uni-TianYan	semi-open-source	73.81	72.1	87.4	69.91	65.81
15	★ ♦ LLaMA_2_70B_LoRA	semi-open-source	<u>73.9</u>	72.7	87.55	70.84	64.52
16	★ ♦ Marcoroni-70B-v1	semi-open-source	74.06	73.55	87.62	70.67	64.41
17	★ ♦ FashionGPT-70B-V0.1(a)	FashionGPT experiment(Unreleased)	69.62	69.11	87.71	70.33	51.34
18	★ ♦ FashionGPT-70B-V0.1(b)	FashionGPT experiment(Unreleased)	68.58	67.66	87.32	69.56	49.76
19	★ ♦ FashionGPT-70B-V0.1(c)	FashionGPT experiment(Unreleased)	71.92	68.77	87.33	69.91	61.68
20	★ ♦ FashionGPT-70B-V0.1(d)	FashionGPT experiment(Unreleased)	72.11	69.97	87.48	70.69	60.32
21	★ ♦ FashionGPT-70B-V0.1(e)	FashionGPT experiment(Unreleased)	71.76	69.88	87.12	70.51	61.19
22	× ♦ FashionGPT-70B-V0.2	FashionGPT experiment(Unreleased)	71.08	69.03	87.17	69.95	58.19
23	× ♦ FashionGPT-70B-V0.3	FashionGPT experiment(Unreleased)	71.28	70.39	87.32	70.06	57.34
24	× ♦ FashionGPT-70B-V0.4	FashionGPT experiment(Unreleased)	71.78	70.82	87.14	70.15	59.03
25	○ ♦ FashionGPT-70B-V0.5	FashionGPT experiment(Unreleased)	72.76	71.33	84.52	69.71	65.47
26	○ ♦ FashionGPT-70B-V1	FashionGPT series(Released)	73.26	71.08	87.32	70.7	63.92
27	○ ♦ FashionGPT-70B-V1.1	FashionGPT series(Released)	74.05	71.76	88.2	70.99	65.26
28	○ ♦ FashionGPT-70B-V1.2	FashionGPT series(Released)	74.11	73.04	88.15	70.11	65.15

Table 8

Fine-tuning details of FashionGPT experiments.

Model	Dataset	Method	Fusion weight
FashionGPT-70B-V0.1(a)	Airoboros-gpt4-1.4.1	Fine-tuning on 1 QA dataset	—
FashionGPT-70B-V0.1(b)	Open-Platypus	Fine-tuning on 1 QA dataset	—
FashionGPT-70B-V0.1(c)	GPT-4 Multi-turn Chat	Fine-tuning on 1 chat dataset	—
FashionGPT-70B-V0.1(d)	Refined Orca-Style	Fine-tuning on 1 QA dataset	—
FashionGPT-70B-V0.1(e)	Samantha Subdataset	Fine-tuning on 1 chat dataset	—
FashionGPT-70B-V0.2	GPT-4 Multi-turn Chat Open-Platypus	Fine-tuning on fused dataset	— —
FashionGPT-70B-V0.3	GPT-4 Multi-turn Chat Refined Orca-Style	Fine-tuning on fused dataset	— —
FashionGPT-70B-V0.4	GPT-4 Multi-turn Chat Refined Orca-Style	Sequential fine-tuning	— —
FashionGPT-70B-V0.5	GPT-4 Multi-turn Chat Refined Orca-Style	Multiple LoRA-adaptor fusion with fusion weights not fine-tuned	1 1

weights from (1, 1) to (0.7, 0.7), a significant performance improvement was evident for LLM₂₆, with the score increasing from 72.76 to 73.26. This observed performance difference between LLM₂₅ and LLM₂₆ can be attributed to the impact of fine-tuned fusion weights. In conclusion, both the fusion of LoRA-adapters and the fine-tuning of LoRA-adaptor fusion weights play critical roles in our proposed LoRA-adaptor fusion paradigm.

5. Limitations

Similar to other variants fine-tuned from LLaMA-2-70B, FashionGPT series inherit many limitations of the base model such as hallucination, toxicity, and stereotypes while introducing some unique challenges due to its specialized fine-tuning. Similar to LLaMA-2, FashionGPT-70B series possess out-of-date knowledge from LLaMA-2-70B. This static knowledge base can lead to outdated or incomplete information over time and even generate non-factual content or unqualified advice. For example, FashionGPT series may generate false responses to some questions which strongly depend on time, such as “Who is the current

premier of Thailand?”. Additionally, LLaMA-2 series are not designed to be multilingual and we did not fine-tune it with massive multilingual corpora, so its multilingual ability is unstable and can be inconsistent due to the lack of non-English pretrain data.

Misusing the capabilities of FashionGPT series, similar to other AI models, can result in spreading misinformation or exploring sensitive topics for malicious purposes. Although our model is strictly for non-commercial use, as per the training set’s licence, we have adhered to Meta’s Responsible Use Guide during the fine-tuning process. However, it is important to note that we have not conducted any adversarial attack testing or red teaming. Therefore, developers should perform customized safety testing and tuning for their specific model applications before deploying any FashionGPT-based applications.

6. Conclusion and future work

In this work, we introduce FashionGPT, a series of advanced LLMs enhanced through a novel fine-tuning method called multiple LoRA-adaptor fusion, which has outperformed traditional techniques in benchmark tests and has won first place twice on the HuggingFace Open LLM

Table 9

Modified LLaMA-2 conversation template.

You are an AI assistant named FashionGPT.
 USER: [The query 1 of user]
 ASSISTANT: [The response 1 of AI assistant] </s>
 USER: [The query 2 of user]
 ASSISTANT: [The response 2 of AI assistant] </s>
 ...

Leaderboard. The release of FashionGPT on HuggingFace for research has significantly contributed to the open-source community, providing models that showcase the benefits of our fine-tuning approach. FashionGPT's success not only sets a new benchmark for LLMs but also underscores the potential of our fine-tuning method to broaden the application and effectiveness of LLMs in various fields, leading the way for future language model research. In the future, we plan to explore other ensembling methods for LoRA-adapters (such as mixture of LoRA-adapters) to further improve the performance of LLMs on a variety of benchmarks. Moreover, we will consider to enhance the ability of LLMs to conduct multi-turn chats with human and evaluate the quality of responses according to helpfulness, professionalism, consistency, and proactivity.

CRedit authorship contribution statement

Dehong Gao: Writing – original draft, Methodology, Conceptualization. **Yufei Ma:** Writing – original draft, Visualization, Software, Formal analysis. **Sen Liu:** Writing – review & editing, Visualization, Software, Formal analysis, Data curation. **Mengfei Song:** Writing – original draft, Data curation, Conceptualization. **Linbo Jin:** Writing – review & editing, Resources, Formal analysis. **Wen Jiang:** Writing – review & editing, Resources, Methodology, Conceptualization. **Xin Wang:** Writing – review & editing, Resources, Formal analysis, Conceptualization. **Wei Ning:** Writing – review & editing, Resources, Methodology, Conceptualization. **Shanqing Yu:** Writing – review & editing, Formal analysis. **Qi Xuan:** Writing – review & editing, Formal analysis. **Xiaoyan Cai:** Writing – review & editing, Methodology, Conceptualization. **Libin Yang:** Writing – review & editing, Methodology, Conceptualization.

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.

Data availability

Data will be made available on request.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grants U20B2065, U22B2036, and 62372380, National Key Research and Development Project under Grant 2022YFB3104005, and Natural Science Basic Research Program of Shaanxi (Program No. 2024JC-YBMS-513), and Key Research and Development Program of Zhejiang Province under Grants 2024C01025.

Appendix A. Modified LLaMA-2 conversation template

The prompt in Table 9 is a modified LLaMA-2 conversation template used to fine-tune LoRA-adapters.

Appendix B. Rules for filtering unqualified instances in GPT-4 multi-turn chat dataset

- We only keep chat instances between human and GPT-4, while chat instances between human and ChatGPT-3.5-Turbo are all removed.
- Chats where GPT-4 refused to respond or explain directly are all removed.
- Chats where the response of GPT-4 is blank or missed are all removed.
- Chats that contain violent, racist, toxic, or illegal information are all removed.
- Chats that contain *OpenAI* or *ChatGPT* are all removed or replaced by other information to ensure the correct identity of our model.
- We divide lengthy chat instances into smaller segments that fit the model's maximum context length.
- User queries with similarity > 85% to any benchmark question are all removed.

References

- [1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, NeurIPS '17, Long Beach, California, USA, 2017, pp. 6000–6010.
- [2] J. Kaplan, S. McCandlish, T. Henighan, T.B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, D. Amodei, Scaling laws for neural language models, 2020, [arXiv:2001.08361](https://arxiv.org/abs/2001.08361).
- [3] T.B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, in: Proceedings of the 34th International Conference on Neural Information Processing Systems, Vol. 159, NIPS '20, Vancouver, BC, Canada, 2020, pp. 1877–1901.
- [4] T.L. Scao, A. Fan, C. Akiki, E. Pavlick, S. Ilić, D. Hesslow, R. Castagné, A.S. Luccioni, F. Yvon, M. Gallé, et al., Bloom: A 176b-parameter open-access multilingual language model, 2022, [arXiv preprint arXiv:2211.05100](https://arxiv.org/abs/2211.05100).
- [5] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F.L. Aleman, D. Almeida, J. Altschmidt, S. Altman, et al., GPT-4 technical report, 2023, [arXiv:2303.08774](https://arxiv.org/abs/2303.08774).
- [6] R. Anil, A.M. Dai, O. Firat, M. Johnson, D. Lepikhin, A. Passos, S. Shakeri, E. Taropa, P. Bailey, Z. Chen, et al., Palm 2 technical report, 2023, [arXiv:2305.10403](https://arxiv.org/abs/2305.10403).
- [7] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, G. Lample, Llama: Open and efficient foundation language models, 2023, [arXiv:2302.13971](https://arxiv.org/abs/2302.13971).
- [8] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al., Llama 2: Open foundation and fine-tuned chat models, 2023, [arXiv:2307.09288](https://arxiv.org/abs/2307.09288).
- [9] MosaicML, Introducing MPT-7B: A new standard for open-source, commercially usable LLMs, 2023, www.mosaicml.com/blog/mpt-7b. (Accessed 5 May 2023).
- [10] E. Almazrouei, H. Alobeidli, A. Alshamsi, A. Cappelli, R. Cojocaru, M. Debbah, É. Goffinet, D. Hesslow, J. Launay, Q. Malartic, D. Mazzotta, B. Noune, B. Pannier, G. Penedo, The falcon series of open language models, 2023, [arXiv:2311.16867](https://arxiv.org/abs/2311.16867).
- [11] A.N. Lee, C.J. Hunter, N. Ruiz, Platypus: Quick, cheap, and powerful refinement of LLMs, 2024, [arXiv:2308.07317](https://arxiv.org/abs/2308.07317).
- [12] Y. Wang, Y. Kordi, S. Mishra, A. Liu, N.A. Smith, D. Khashabi, H. Hajishirzi, Self-instruct: Aligning language models with self-generated instructions, in: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Toronto, Canada, 2023, pp. 13484–13508.
- [13] G. Dong, H. Yuan, K. Lu, C. Li, M. Xue, D. Liu, W. Wang, Z. Yuan, C. Zhou, J. Zhou, How abilities in large language models are affected by supervised fine-tuning data composition, 2024, [arXiv:2310.05492](https://arxiv.org/abs/2310.05492).
- [14] E.J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, LoRA: Low-rank adaptation of large language models, 2021, [arXiv preprint arXiv:2106.09685](https://arxiv.org/abs/2106.09685).
- [15] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, Minnesota, 2019, pp. 4171–4186.

- [16] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H.W. Chung, C. Sutton, S. Gehrmann, et al., PaLM: Scaling language modeling with pathways, 2022, [arXiv:2204.02311](https://arxiv.org/abs/2204.02311).
- [17] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P.J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, 2023, [arXiv:1910.10683](https://arxiv.org/abs/1910.10683).
- [18] S. Longpre, L. Hou, T. Vu, A. Webson, H.W. Chung, Y. Tay, D. Zhou, Q.V. Le, B. Zoph, J. Wei, et al., The flan collection: Designing data and methods for effective instruction tuning, 2023, [arXiv:2301.13688](https://arxiv.org/abs/2301.13688).
- [19] V. Aribandi, Y. Tay, T. Schuster, J. Rao, H.S. Zheng, S.V. Mehta, H. Zhuang, V.Q. Tran, D. Bahri, J. Ni, et al., ExT5: Towards extreme multi-task scaling for transfer learning, 2022, [arXiv:2111.10952](https://arxiv.org/abs/2111.10952).
- [20] V. Sanh, A. Webson, C. Raffel, S.H. Bach, L. Sutawika, Z. Alyafeai, A. Chaffin, A. Stiegler, T.L. Scao, A. Raja, et al., Multitask prompted training enables zero-shot task generalization, 2022, [arXiv:2110.08207](https://arxiv.org/abs/2110.08207).
- [21] H.W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, Y. Li, X. Wang, M. Dehghani, S. Brahma, et al., Scaling instruction-finetuned language models, 2022, [arXiv:2210.11416](https://arxiv.org/abs/2210.11416).
- [22] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, et al., Training language models to follow instructions with human feedback, in: *Advances in Neural Information Processing Systems*, vol. 35, 2022, pp. 27730–27744.
- [23] S. Mukherjee, A. Mitra, G. Jawahar, S. Agarwal, H. Palangi, A. Awadallah, Orca: Progressive learning from complex explanation traces of GPT-4, 2023, [arXiv:2306.02707](https://arxiv.org/abs/2306.02707).
- [24] C. Xu, D. Guo, N. Duan, J. McAuley, Baize: An open-source chat model with parameter-efficient tuning on self-chat data, 2023, [arXiv:2304.01196](https://arxiv.org/abs/2304.01196).
- [25] C. Xu, Q. Sun, K. Zheng, X. Geng, P. Zhao, J. Feng, C. Tao, D. Jiang, WizardLM: Empowering large language models to follow complex instructions, 2023, [arXiv:2304.12244](https://arxiv.org/abs/2304.12244).
- [26] T. Dettmers, A. Pagnoni, A. Holtzman, L. Zettlemoyer, QLoRA: Efficient finetuning of quantized LLMs, 2023, [arXiv:2305.14314](https://arxiv.org/abs/2305.14314).
- [27] H. Wang, T. Sun, C. Fan, J. Gu, Customizable combination of parameter-efficient modules for multi-task learning, 2023, [arXiv:2312.03248](https://arxiv.org/abs/2312.03248).
- [28] Y. Zhu, N. Wichers, C.-C. Lin, X. Wang, T. Chen, L. Shu, H. Lu, C. Liu, L. Luo, J. Chen, L. Meng, SiRA: Sparse mixture of low rank adaptation, 2023, [arXiv:2311.09179](https://arxiv.org/abs/2311.09179).
- [29] T. Zadouri, A. Üstün, A. Ahmadian, B. Ermiş, A. Locatelli, S. Hooker, Pushing mixture of experts to the limit: Extremely parameter efficient MoE for instruction tuning, 2023, [arXiv:2309.05444](https://arxiv.org/abs/2309.05444).
- [30] F. Wan, X. Huang, D. Cai, X. Quan, W. Bi, S. Shi, Knowledge fusion of large language models, in: *The Twelfth International Conference on Learning Representations*, 2024.
- [31] S. Gunasekar, Y. Zhang, J. Aneja, C.C.T. Mendes, A. Del Giorno, S. Gopi, M. Javaheripi, P. Kauffmann, G. de Rosa, O. Saarikivi, et al., Textbooks are all you need, 2023, [arXiv:2306.11644](https://arxiv.org/abs/2306.11644).
- [32] E. Beeching, C. Fourier, N. Habib, S. Han, N. Lambert, N. Rajani, O. Sanseviero, L. Tunstall, T. Wolf, Open LLM leaderboard, 2023, Hugging Face https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard.
- [33] A. Köpf, Y. Kilcher, D. von Rütte, S. Anagnostidis, Z.R. Tam, K. Stevens, A. Barhoum, N. Duc, O. Stanley, R. Nagyfi, et al., Openassistant conversations - democratizing large language model alignment, 2023, [arXiv:2304.07327](https://arxiv.org/abs/2304.07327).
- [34] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, et al., Transformers: State-of-the-art natural language processing, in: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2020, pp. 38–45.
- [35] S. Mangrulkar, S. Gugger, L. Debut, Y. Belkada, S. Paul, B. Bossan, PEFT: State-of-the-art parameter-efficient fine-tuning methods, 2022, <https://github.com/huggingface/peft>.
- [36] P. Clark, I. Cowhey, O. Etzioni, T. Khot, A. Sabharwal, C. Schoenick, O. Tafjord, Think you have solved question answering? Try ARC, the AI2 reasoning challenge, 2018, [arXiv:1803.05457](https://arxiv.org/abs/1803.05457).
- [37] R. Zellers, A. Holtzman, Y. Bisk, A. Farhadi, Y. Choi, HellaSwag: Can a machine really finish your sentence? in: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy, 2019, pp. 4791–4800.
- [38] D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, J. Steinhardt, Measuring massive multitask language understanding, in: *Proceedings of the International Conference on Learning Representations, ICLR*, 2021.
- [39] S. Lin, J. Hilton, O. Evans, TruthfulQA: Measuring how models mimic human falsehoods, in: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Dublin, Ireland, 2022, pp. 3214–3252.



Dehong Gao is an associate professor in School of Cybersecurity, Northwestern Polytechnical University. He received the Ph.D. degrees from The Hong Kong Polytechnic University, Hong Kong, China, in 2014. He worked as a senior algorithm expert in Alibaba Group from 2014 to 2022. His research interests include large language modelling, multimodal research, information retrieval, recommendation, natural language processing and machine learning.



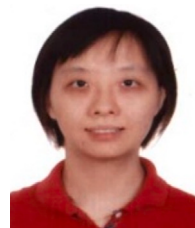
Yufei Ma received the Bachelor's degree from the School of Material Science and Engineering, Northwestern Polytechnical University, China and is currently studying for the M.S. degree from the School of Cybersecurity, Northwestern Polytechnical University. His research method is text generation in natural language processing, including multimodal summarization, large language model.



Shanqing Yu received the M.S. degree from the School of Computer Engineering and Science, Shanghai University, China, in 2008 and received the M.S. degree from the Graduate School of Information, Production and Systems, Waseda University, Japan, in 2008, and the Ph.D. degree, in 2011, respectively. She is currently an associate professor at the Institute of Cyberspace Security and the College of Information Engineering, Zhejiang University of Technology, Hangzhou, China. Her research interests cover intelligent computation and data mining.



Qi Xuan received the B.S. and Ph.D. degrees in control theory and engineering from Zhejiang University, Hangzhou, China, in 2003 and 2008, respectively. He was a Post-Doctoral Researcher with the Department of Information Science and Electronic Engineering, Zhejiang University, from 2008 to 2010, respectively, and a Research Assistant with the Department of Electronic Engineering, City University of Hong Kong, Hong Kong, in 2010 and 2017. From 2012 to 2014, he was a Post-Doctoral Fellow with the Department of Computer Science, University of California at Davis, CA, USA. He is a senior member of the IEEE and is currently a Professor with the Institute of Cyberspace Security, College of Information Engineering, Zhejiang University of Technology, Hangzhou, China. His current research interests include network science, graph data mining, deep learning, cyberspace security, machine learning, and computer vision.



Xiaoyan Cai is an associate professor in School of Automation, Northwestern Polytechnical University. She was a research associate in department of computing, the Hong Kong Polytechnic University, Hong Kong, from June 2009 to June 2011. She received the Ph.D. degree from Northwestern Polytechnical University, China, in 2009. Her current research interests include document summarization, information retrieval and machine learning.



Libin Yang is an associate professor in School of Automation, Northwestern Polytechnical University. He was a research associate in department of computing, the Hong Kong Polytechnic University, Hong Kong, from June 2009 to June 2011. He received the Ph.D. degree from Northwestern Polytechnical University, China, in 2009. His current research interests include information retrieval, computer network and game theory.