LoraMap: Harnessing the Power of LoRA Connections

Anonymous ACL submission

Abstract

Fact-checking techniques can mitigate hallucinations in Large Language Models (LLMs), a prominent issue in specialized domains. As parameter-efficient techniques such as Low-Rank Adaptation (LoRA) can overcome substantial computational overhead, some studies have explored the integration of multiple LoRAs. While previous studies focus on parallel integration, this paper investigates methods to establish connections among multiple LoRAs. We create three reasoning datasets tailored to fact-checking and fine-tune individual LoRAs, allowing them to view and reason from diverse perspectives. Then, we explore strategies for allocating these reasoning LoRAs and introduce LoraMap, an approach to map connections between them. The results of the fact-checking task demonstrate that the performance of LoraMap is superior to LoraHub, an existing method for integrating LoRAs. LoraMap also outperforms with significantly fewer trainable parameters than LoraConcat, which concatenates LoRAs and further fine-tunes them.

1 Introduction

With the rapid progress in research leveraging Large Language Models (LLMs) such as GPT-4 (OpenAI, 2023), PaLM (Chowdhery et al., 2023), LLaMA (Touvron et al., 2023), and Flan-T5 (Chung et al., 2022) in various natural language processing tasks, several challenges have also emerged. The model can pose a significant risk to reliability and trustworthiness due to the issue of generating false information, known as hallucination (Ji et al., 2023). One way to alleviate this problem is using fact-checking to verify LLM outputs or given claims (Gupta et al., 2022; Chamoun et al., 2023).

As in Figure 1, a fact-checking process classifies a claim into true, false, or more sophisticated labels based on textual evidence such as Wikipedia passages, news articles, and other relevant documents (Thorne et al., 2018; Guo et al., 2022). In biomedical and health domains, serious problems can arise when people perceive false information as truth, highlighting the importance of fact-checking. Accordingly, many studies have been explored, resulting in the development of datasets: SciFact (Wadden et al., 2020), PubHealth (Kotonya and Toni, 2020), COVID-Fact (Saakyan et al., 2021), and HealthVer (Sarrouti et al., 2021). This paper focuses on small datasets, COVID-Fact and SciFact.

Another challenge is that fine-tuning the LLMs requires high computational demands. Parameter-efficient fine-tuning techniques can address this issue, especially Low-rank adaptations (LoRA) (Hu et al., 2021). As numerous task-specific LoRAs have appeared, some studies have explored the integration of these modules to serve auxiliary roles in addressing new tasks (Huang et al., 2023; Liu et al., 2023; Gao et al., 2024; Li et al., 2024; Dou et al., 2023). Among these methods, LoraHub (Huang et al., 2023) learns weights for each LoRA and computes their weighted sum in parallel, which may weaken the influence of the pivotal LoRA.

This paper investigates methods for establishing connections among LoRAs to exchange their specialized insights as an alternative to parallel integration. Our main contributions are as follows:

- We create three reasoning datasets tailored to fact-checking and fine-tune LoRA for each dataset, allowing them to infer from various perspectives.
- We investigate how to integrate these reasoning LoRAs and introduce LoraMap. Inspired by the information-processing behavior of the human brain in neuroscience, it learns connections rather than a linear sum of LoRAs.
- The results on the COVID-Fact and SciFact datasets demonstrate that LoraMap exhibits superior performance than LoraHub and also outperforms LoraConcat even with significantly fewer trainable parameters.

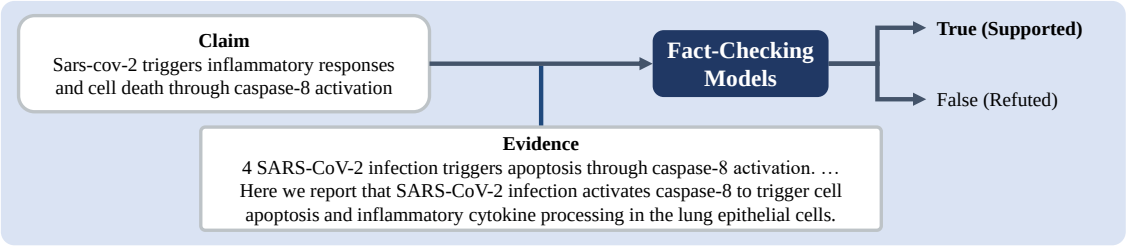

Figure 1: A fact-checking task classifies a claim into true or false based on the corresponding evidence.

2 Related Work

2.1 Fact-checking using LLMs

Recent studies have explored the potential of LLMs for fact-checking through zero-shot prompting (Chern et al., 2023; Li et al., 2023; Wang et al., 2023b), hierarchical step-by-step prompting with external knowledge (Zhang and Gao, 2023), multi-agent debating (Du et al., 2023), question answering in intermediate steps (Pan et al., 2023), and evaluating factuality within lengthy text (Min et al., 2023). These approaches employ advanced prompting strategies and external knowledge to enhance reasoning. The results demonstrate the effectiveness of leveraging LLMs for fact-checking and suggest further advancements in factual reasoning (Laban et al., 2023; Wang et al., 2023a).

2.2 Biomedical Fact-checking

The rapid expansion of evidence, encompassing biomedical articles and literature, has made manual fact-checking challenging and time-consuming. Several studies have attempted to construct biomedical fact-checking datasets and train diverse models. For the PubHealth dataset, the SciBERT model achieves the highest f1-score among the BERT models (Kotonya and Toni, 2020). On the SciFact leaderboard[1], the best model is MultiVerS (Wadden et al., 2022), a Longformer model (Beltagy et al., 2020) that is trained with rationale sentence selection and fact-checking label prediction. For COVID-Fact, the RoBERTa model is fine-tuned on fact-checking and entailment inference datasets (Saakyan et al., 2021). In the case of HealthVer, the T5-base model performed better than the BERT models (Sarrouti et al., 2021). The COVID-Fact and SciFact are small datasets containing fewer than 5,000 claim-evidence pairs, whereas the HealthVer and PubHealth datasets include more than 10,000 instances.

[1] https://leaderboard.allenai.org/scifact/submissions/public

2.3 Parameter-efficient Fine-tuning

Several studies have introduced parameter-efficient fine-tuning techniques that freeze the original model parameters and only fine-tune a few additional parameters. Adapter tuning (Houlsby et al., 2019; Pfeiffer et al., 2020) inserts a layer into each transformer layer, which consists of a down-projection feed-forward layer, a non-linearity function, and an up-projection feed-forward layer. Prefix tuning (Li and Liang, 2021) appends trainable prefix tokens to the input sequence, and prompt tuning (Lester et al., 2021) perpends learnable continuous prompt vectors to the input embeddings. LoRA (Hu et al., 2021) decomposes the model weight matrix in the target layer into two trainable low-rank matrices.

2.4 Quantization

Another way to reduce computational requirements is by applying quantization techniques to LLMs to decrease the numerical precision of model parameters. The LLM.int8() (Dettmers et al., 2022) quantizes model weights to 8-bit integers through a vector-wise quantization and mixed-precision decomposition. The QLoRA (Dettmers et al., 2024) converts model weights to 4-bit integers by employing three techniques: 4-bit NormalFloat quantization, double quantization, and paged optimizers.

3 Methods

3.1 Reasoning Dataset Generation

Determining the veracity of a claim requires identifying key entities and their relationships within the claim and evidence, and then analyzing the differences between them. In this context, we hypothesize that identifying contrasting or common factors between the claim sentence and its corresponding evidence can assist the fact-checking model. Therefore, we customize the three reasoning tasks for fact-checking: DifferenceCoT, EntityCoT, and CorrectClaim.

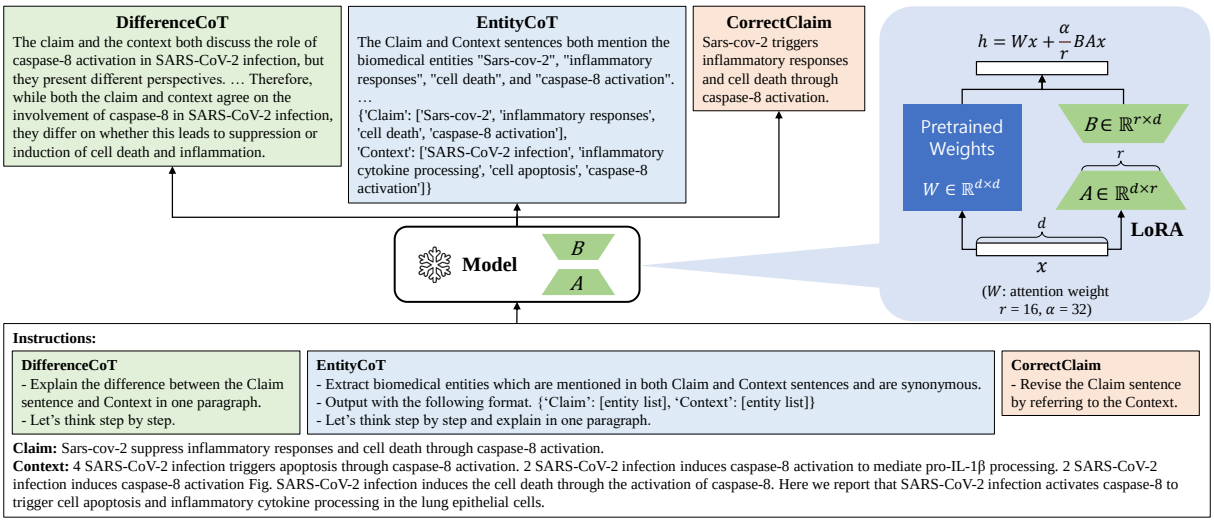

Figure 2: Input and output examples of reasoning datasets for fine-tuning the generative model with LoRA. The LoRA consists of A and B weight matrices and exists in transformer layers.

DifferenceCoT is a task that explains the contextual differences between claim and evidence, including topic, level of detail, and relation.

EntityCoT is a task that extracts synonymous biomedical entities that are concurrently present in both claim and evidence.

CorrectClaim is a task that revises a given claim sentence based on the evidence.

Then, we construct the reasoning datasets using existing fact-checking datasets: COVID-Fact and SciFact. The input of the reasoning datasets includes task instruction, claim, and evidence, as shown in Figure 2. For DifferenceCoT and EntityCoT, we employ Chain-of-Thought (CoT) prompting (Wei et al., 2022) and use GPT-4 API to generate the ground truth output. For CorrectClaim, the ground truth output is the true claim for the given evidence. The details and assessments of reasoning datasets are in Section 5.3 and Appendix A.

3.2 Fine-tuning Reasoning LoRAs

The next step is to fine-tune LoRAs for each task. The lightweight LoRA module exists in transformer attention or feed-forward layers of the base model. For each task $t \in \{1, 2, 3\}$, LoRA consists of a weight matrix $A_t \in \mathbb{R}^{d\times r}$ for down-projection of features to a smaller dimension r, and a weight matrix $B_t \in \mathbb{R}^{r\times d}$ for up-projection to the original dimension d, as depicted in Figure 2. By freezing the base model weights and training only the LoRA weights, the fine-tuning process requires much fewer parameters.

Among encoder-decoder models, we use Flan-T5 due to its range of model size options and its strong performance in zero-shot, few-shot, and CoT (Chung et al., 2022). In the Flan-T5 model, LoRA applies to the query and value projections of the encoder self-attention, decoder self-attention, and encoder-decoder attention layers. For decoder-only models, we use LLaMA3 (Dubey et al., 2023) as a small LLM and apply LoRA to the query, value, key, and output projections of the attention layers, up and down projections of the feed-forward layers, and the gate mechanism.

3.3 Connecting Reasoning LoRAs

The final step involves investigating methods to allocate and connect reasoning LoRAs, specifically LoraHub, LoraConcat, and LoraMap. These methods differ in approaches for integrating the $A_t \in \mathbb{R}^{d\times r}$ and $B_t \in \mathbb{R}^{r\times d}$ matrices. Figure 3 illustrates a visual comparison of these methods.

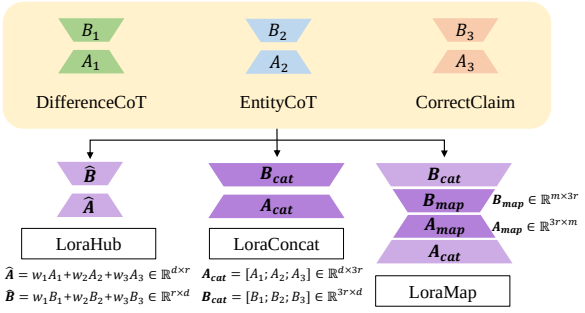

Figure 3: The comparison of LoraHub, LoraConcat, and LoraMap. Dark purple indicates trainable weights, and light purple represents fixed weights.

LoraHub combines LoRAs by computing the weighted sum of A_t and B_t matrices to generate $\hat{A}$ and $\hat{B}$ matrices.

$$\hat{A} = \sum_{t=1}^{n} w_t A_t = w_1 A_1 + \ldots + w_n A_n \in \mathbb{R}^{d \times r}$$

$$\hat{B} = \sum_{t=1}^{n} w_t B_t = w_1 B_1 + \ldots + w_n B_n \in \mathbb{R}^{r \times d}$$

This method freezes all A_t and B_t matrices and learns only the coefficients w_t for each LoRA. The training is finding the optimal coefficients w_t for a target dataset by a gradient-free approach. The original LoraHub randomly selects 20 LoRAs from approximately 200 LoRA modules trained on distinct tasks. Our LoraHub loads three reasoning LoRAs along with these 20 LoRA modules and learns 23 coefficients for each LoRA layer.

LoraConcat integrates LoRAs by concatenating the matrices A_t and B_t of the three reasoning LoRAs to produce A_{cat} and B_{cat} matrices.

$$A_{cat} = [A_1; A_2; A_3] \in \mathbb{R}^{d \times 3r}$$

$$B_{cat} = [B_1; B_2; B_3] \in \mathbb{R}^{3r \times d}$$

Then, we fine-tune the A_{cat} and B_{cat} matrices for the target dataset. As there are three LoRAs to combine, LoraConcat has $2 \times 3r \times d$ trainable parameters for each LoRA layer.

LoraMap not only concatenates the three reasoning LoRAs into A_{cat} and B_{cat} but also inserts the trainable matrices A_{map} and B_{map} between them.

$$A_{map} \in \mathbb{R}^{3r \times m} \ , \ B_{map} \in \mathbb{R}^{m \times 3r}$$

LoraMap freezes A_{cat} and B_{cat} to maintain specialized reasoning capabilities and learns the connection maps between them by fine-tuning only A_{map} and B_{map}. Each LoRA layer has $2 \times 3r \times m$ trainable parameters. We define the mapping dimension m based on the ratio of trainable parameters to the total number of parameters in the model.

$$m = \frac{ratio \times num\ of\ total\ parameters}{3r \times num\ of\ trainable\ layers}$$

The number of trainable layers refers to the total number of A_{map} and B_{map} layers. The default value for m is equivalent to r, allowing for adjustments to m by modifying the ratio.

Dataset	Partition	True Instances	False Instances	Total Instances
COVID-Fact	Train	1,018	1,018	2,036
	Val	129	129	258
	Test	128	128	256
SciFact	Train	456	237	693
	Val	124	64	188
	Test	100	100	200

Table 1: The statistics of the COVID-Fact and SciFact dataset for generating reasoning datasets

4 Experimental Results

4.1 Datasets

We conduct experiments on COVID-Fact and SciFact datasets and extract data to construct reasoning datasets. In the COVID-Fact dataset, each piece of evidence is associated with at least one true claim and one false claim. For this dataset, we randomly select one true and one false claim per evidence, resulting in 2,550 instances. In contrast, the SciFact dataset has three veracity labels: true, false, and not enough information. We include only the true and false claims and exclude claims without evidence, yielding 1,081 instances. Additionally, since SciFact does not consistently offer both true and false claims for each evidence, we employ CoT prompting with the GPT-4 API to generate CorrectClaim. The statistics for these datasets are shown in Table 1.

4.2 Reasoning LoRAs

Table 2 presents the results of three reasoning LoRAs, using BLEU (Papineni et al., 2002), ROUGE (Lin, 2004; Lin and Och, 2004), and METEOR (Banerjee and Lavie, 2005) scores as lexical overlap-based metrics, and BERTscore (Zhang et al., 2019) with the Longformer-base model as semantic embedding-based metrics. Among models below 1B, we experiment with Flan-T5-large (787M) and for models above 1B, we use Flan-T5-xxl (11B) quantizied with LLM.int8() and LLaMA3 (8B) quantizied with QLoRA.

In the zero-shot setting, the base model performs reasoning tasks without fine-tuning, resulting in poor scores. Fine-tuning LoRA on each reasoning dataset improves performance on all metrics. LLaMA3 outperforms other models on DifferenceCoT and EntityCoT of COVID-Fact datasets, whereas it has weaknesses on the CorrectClaim. We suppose this is primarily due to its lower zero-shot performance on CorrectClaim, which also impacts its fine-tuning performance. All experiments use a fixed seed 42 for reproducibility, and the experimental details are in Appendix B.1.

Dataset	Reasoning LoRA	Base model (total params)	Setting (trainable params)	BLEU	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-Lsum	METEOR	BERTscore
COVID-Fact	DifferenceCoT	Flan-T5-large	Zero-shot	0.0023	0.2173	0.1326	0.1815	0.2011	0.1047	0.8563
		(787M)	LoRA finetuning (4M)	0.3588	0.6676	0.4206	0.5045	0.6310	0.5255	0.9275
		Flan-T5-xxl	Zero-shot	0.0012	0.2034	0.1298	0.1718	0.1875	0.0945	0.8545
		(11B)	qLoRA finetuning (18M)	0.3764	0.6822	0.4446	0.5192	0.6444	0.5245	0.9315
		LLaMA3	Zero-shot	0.2057	0.6097	0.3180	0.3969	0.5655	0.4424	0.9092
		(8B)	qLoRA finetuning (41M)	**0.3889**	**0.6980**	**0.4620**	**0.5319**	**0.6597**	**0.5582**	**0.9337**
	EntityCoT	Flan-T5-large	Zero-shot	0	0.0539	0.0201	0.0533	0.0526	0.0289	0.7997
		(787M)	LoRA finetuning (4M)	0.3885	0.6755	0.4533	0.5548	0.6397	0.5969	0.9240
		Flan-T5-xxl	Zero-shot	0	0.0444	0.0238	0.0444	0.0442	0.0097	0.7903
		(11B)	qLoRA finetuning (18M)	0.3805	0.6680	0.4505	0.5500	0.6356	0.5881	0.9223
		LLaMA3	Zero-shot	0.1799	0.5200	0.2451	0.3221	0.4826	0.4094	0.8850
		(8B)	qLoRA finetuning (41M)	**0.4115**	**0.6819**	**0.4604**	**0.5577**	**0.6513**	**0.6092**	**0.9273**
	CorrectClaim	Flan-T5-large	Zero-shot	0.3636	0.6839	0.5714	0.6618	0.6636	0.6591	0.9349
		(787M)	LoRA finetuning (4M)	**0.9257**	**0.9722**	**0.9437**	**0.9721**	**0.9721**	**0.9682**	**0.9944**
		Flan-T5-xxl	Zero-shot	0.5212	0.8102	0.7251	0.7985	0.7983	0.7821	0.9565
		(11B)	qLoRA finetuning (18M)	0.9227	0.9700	0.9389	0.9695	0.9696	0.9662	0.9943
		LLaMA3	Zero-shot	0.054	0.3248	0.2019	0.2882	0.2907	0.4766	0.8695
		(8B)	qLoRA finetuning (41M)	0.3664	0.9676	0.9377	0.9671	0.9670	0.9562	0.9890
SciFact	DifferenceCoT	LLaMA3	Zero-shot	0.2185	0.5873	0.3135	0.3966	0.5490	0.4362	0.9096
		(8B)	qLoRA finetuning (41M)	**0.3663**	**0.6799**	**0.4521**	**0.5261**	**0.6513**	**0.5610**	**0.9357**
	EntityCoT	LLaMA3	Zero-shot	0.1839	0.5060	0.2277	0.3271	0.4714	0.4547	0.8867
		(8B)	qLoRA finetuning (41M)	**0.4200**	**0.6766**	**0.4503**	**0.5397**	**0.6412**	**0.6246**	**0.9272**
	CorrectClaim	LLaMA3	Zero-shot	0.2627	0.5984	0.3701	0.4358	0.5582	0.4356	0.9210
		(8B)	qLoRA finetuning (41M)	**0.4055**	**0.6861**	**0.4970**	**0.5464**	**0.6571**	**0.6308**	**0.9412**

Table 2: The evaluation results of three reasoning LoRAs on the COVID-Fact and SciFact test dataset. The bold text indicates the best performance for each reasoning LoRA.

Model	Reasoning LoRA	Fact-checking setting	# Training instances	Macro-precision	Macro-recall	Macro-f1
Flan-T5-large	—	Zero-shot	0	0.7819	0.6133	0.5453
	base20 + DifferenceCoT + EntityCoT + ClaimCorrection	LoraHub	50	**0.6664**	**0.6344**	0.6133
			200	0.6643	0.6340	**0.6145**
			2,036*	0.6589	0.6254	0.6030
	DifferenceCoT + EntityCoT + ClaimCorrection	LoraConcat (14M)	100	0.8087	0.7828	0.7782
			1,000	**0.8334**	**0.8152**	**0.8126**
			2,036*	0.8184	0.8082	0.7910
	DifferenceCoT + EntityCoT + ClaimCorrection	LoraMap (0.22M)	100	0.7527	0.6961	0.6755
			1,000	0.8052	0.8015	0.8010
			2,036*	**0.8302**	**0.8246**	**0.8239**

Table 3: The evaluation results of the Flan-T5-large model on the COVID-Fact test dataset. In the fact-checking settings, the values in parenthesis indicates the number of trainable parameters. Bold text highlights the best performance and * denotes the size of the entire training data.

4.3 Integrating LoRAs for Fact-checking

Given the prompt "*What is the class of the Claim by referring to the Context? Choose only from TRUE or FALSE.*" along with the claim and context, we align the model output to follow the zero-shot output format. "*The claim is TRUE/FALSE*" for Flan-T5 models and "*Based on the context, the given claim is TRUE/FALSE*" for the LLaMA3 model.

4.3.1 Results of Small Language Model

The performance of the Flan-T5-large on the COVID-Fact test dataset is shown in Table 3. In the zero-shot setting, the model predominantly predicts TRUE with an f1 score of 0.5453. The key result is a comparison of LoraHub, LoraConcat, and LoraMap. We experiment with various training instances, and Table 3 presents the best result among 10-shot, 20-shot, 50-shot, and 100-shot, the best result among 200-shot, 500-shot, and 1000-shot, and the result when using the entire dataset. Training with more than 100 instances is to identify the minimum number of instances required to achieve satisfactory performance in LoraConcat and LoraMap. To provide statistically reliable results, all metric scores are the average of ten repeated experiments, each with a fixed seed (42, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384).

LoraHub achieves the highest f1-score at 200-shot, and its performance does not increase as the number of training data increases. Although training LoraHub with less than 100 examples is feasible, its performance is suboptimal. In contrast, LoraConcat and LoraMap generally demonstrate improved f1-scores as training instances increase. LoraConcat yields the best f1-score of 0.8126 at 1000-shot, and LoraMap with default $m(16)$ achieves the best f1-score of 0.8239 when using all instances. LoraMap achieves statistically significant superior performance even with fewer trainable parameters than LoraConcat, with a p-value of 0.03756.

Model	Reasoning LoRA	# Training instances	Macro-precision	Macro-recall	Macro-f1
Flan-T5-large	base20	0	0.7643	0.6094	0.5423
		20	0.7529	0.6836	**0.6603**
	DifferenceCoT + EntityCoT + ClaimCorrection	0	0.6900	0.6758	**0.6696**
		10	0.6889	0.6133	0.5703
	base3	0	0.7647	0.6367	**0.5868**
		10	0.7771	0.5977	0.5199
	base20 + DifferenceCoT + EntityCoT + ClaimCorrection	0	0.7807	0.6094	0.5390
		50	0.6833	0.6797	**0.6781**

Table 4: LoraHub results for the COVID-Fact test dataset depending on the selection of LoRAs.

Model	Reasoning LoRA	# Trainable parameter	Macro-precision	Macro-recall	Macro-f1
Flan-T5-large	DifferenceCoT + EntityCoT	147,456	0.7965	0.7852	0.7831
	DifferenceCoT + ClaimCorrection	147,456	0.7969	0.7969	0.7969
	EntityCoT + ClaimCorrection	147,456	0.7723	0.7656	0.7642
	DifferenceCoT + EntityCoT + ClaimCorrection	221,184	**0.8347**	**0.8281**	**0.8273**

Table 5: LoraMap results for the COVID-Fact test dataset depending on the selection of reasoning LoRAs.

In terms of parameter efficiency, LoraHub has 3,312 (23×144) trainable coefficients out of a total of 787M parameters, as the model consists of 144 layers, each with 23 LoRAs. There are 14M (1024×48×288) trainable parameters out of 797.3M parameters when using LoraConcat and 0.22M (48×16×288) trainable parameters out of 797.5M parameters when using LoraMap. We conduct all experiments with two RTX 3090 GPUs and compare the training and inference time. Training on all COVID-Fact train datasets takes 1 hour 44 minutes for LoraHub, 5 hours 7 minutes for LoraConcat, and 4 hours 14 minutes for LoraMap. For each test instance, inferencing takes less than 0.3 seconds for LoraHub and less than 0.5 seconds for LoraConcat and LoraMap.

4.3.2 Ablation Study

We further compare the results depending on the selection of LoRAs. Table 4 exhibits the results of LoraHub across different training instances, presenting zero-shot and the best result among 10-shot, 20-shot, 50-shot, and 100-shot learning. All experiments use a fixed seed of 42. The original LoraHub uses 20 random LoRAs (base20) and shows improvement after fine-tuning 20 coefficients per layer. When employing only three reasoning LoRAs, the zero-shot performance is higher than that of base20. However, the performance does not improve while fine-tuning due to the difficulty of training only with three coefficient weights. We also experiment with three random LoRAs (base3) to verify this, and the results demonstrate the same tendency to struggle with fine-tuning. Consequently, we keep base20 and add three reasoning LoRAs, yielding the best macro f1 score.

Additionally, LoraHub outputs coefficients that indicates the impact of each LoRA module after training. The coefficients for the three reasoning LoRAs are all close to 0.5. Similarly, four out of the 20 base modules, mostly trained for question-answering, also show 0.5 and the remaining modules show values close to zero or negative. The coefficients confirm that our reasoning LoRAs play an important role in fact-checking.

Table 5 shows the results of an ablation study on LoraMap to demonstrate the effectiveness of each LoRA. All experiments use the entire training dataset with a fixed seed 42. Removing each LoRA degrades the macro-f1 score, and the most influential one is DifferenceCoT LoRA, which exhibits the largest performance decrease. DifferenceCoT, ClaimCorrection, and EntityCoT are most influential in that order, indicating that the tasks identifying and correcting differences between claims and context are more beneficial than finding synonymous entities within it.

4.3.3 Applicability to LLMs

Table 6 shows the performance of LLMs on the COVID-Fact and SciFact test datasets. We compare the performance of zero-shot CoT with the fine-tuned LoraConcat and LoraMap using all training instances. In the COVID-Fact dataset, Flan-T5-xxl with LoraMap (4.4M) achieves the best overall scores, potentially due to its substantial model size. For the Flan-T5-xxl model, LoraMap (4.4M) outperforms LoraConcat (56M), and for the LLaMA3 model, LoraMap (4.1M) surpasses LoraConcat (125M). Likewise, in the SciFact dataset, LLaMA3 with LoraMap (4.1M) exceeds the performance of LoraConcat (125M).

Dataset	Model	Reasoning LoRA	Fact-checking setting	ratio×100	m	Macro-precision	Macro-recall	Macro-f1
COVID-Fact	GPT-4	—	Zero-shot	—	—	0.7426	0.7070	0.6959
	Flan-T5-xxl (11B)	—	Zero-shot	—	—	0.7392	0.7109	0.7021
		DifferenceCoT + EntityCoT + ClaimCorrection	LoraConcat (56M)	—	—	0.8907	0.8906	0.8906
		DifferenceCoT + EntityCoT + ClaimCorrection	LoraMap (5.5M)	0.05	400	0.8879	0.8867	0.8866
			LoraMap (4.4M)	0.04	320	**0.8947**	**0.8945**	**0.8945**
			LoraMap (3.3M)	0.03	240	0.8837	0.8828	0.8827
			LoraMap (2.2M)	0.02	160	0.8671	0.8633	0.8629
			LoraMap (1.1M)	0.01	80	0.8565	0.8555	0.8554
			LoraMap (0.22M)	0.002	16	0.8043	0.7969	0.7956
	LLaMA3 (8B)	—	Zero-shot	—	—	0.6988	0.6172	0.5734
		DifferenceCoT + EntityCoT + ClaimCorrection	LoraConcat (125M)	—	—	0.8076	0.8008	0.7997
		DifferenceCoT + EntityCoT + ClaimCorrection	LoraMap (4.1M)	0.05	192	**0.8210**	**0.8203**	**0.8202**
			LoraMap (3.0M)	0.04	144	0.7824	0.7812	0.7810
			LoraMap (2.4M)	0.03	112	0.7802	0.7773	0.7768
			LoraMap (1.7M)	0.02	80	0.7270	0.7171	0.7139
			LoraMap (0.6M)	0.01	32	0.7123	0.7016	0.6977
			LoraMap (0.3M)	0.005	16	0.7093	0.7031	0.7009
SciFact	LLaMA3 (8B)	—	Zero-shot	—	—	0.8111	0.8100	0.8098
		DifferenceCoT + EntityCoT + ClaimCorrection	LoraConcat (125M)	—	—	0.8347	0.8250	0.8237
		DifferenceCoT + EntityCoT + ClaimCorrection	LoraMap (4.1M)	0.05	192	**0.8802**	**0.8800**	**0.8800**
			LoraMap (3.0M)	0.04	144	0.8659	0.8650	0.8649
			LoraMap (2.4M)	0.03	112	0.8653	0.8650	0.8650
			LoraMap (1.7M)	0.02	80	0.8659	0.8650	0.8649
			LoraMap (0.6M)	0.01	32	0.8606	0.8600	0.8599
			LoraMap (0.3M)	0.005	16	0.8606	0.8600	0.8599

Table 6: The results of LLMs on the COVID-Fact and SciFact test datasets. In the fact-checking settings, the values in parenthesis indicate the number of trainable parameters. The bold text represents the best result for each model.

The results demonstrate that LoraMap consistently outperforms LoraConcat even with significantly fewer trainable parameters. In the case of using LoraMap with LLMs, the default value of m at 16 does not provide sufficient performance. Therefore, by adjusting the ratio of trainable to total parameters, it is feasible to increase m to a higher dimension. Across all models, the macro-f1 generally improves as the size of LoraMap increases.

Table 7 compares trainable parameters, total parameters, training time, and inference time for each test instance. The models using LoraMap have higher total parameters than those using LoraConcat, but have fewer trainable parameters. For the Flan-T5-xxl model, LoraMap also reduces training and inference times compared to LoraConcat. In contrast, while LoraMap considerably shortens the inference time for the LLaMA3 model, it requires a longer training time because LoraMap tends to train for more epochs than LoraConcat. The details of experimental settings are in Appendix B.2.

Dataset	Model	Fact-checking Setting	Total params	Train/Test Time
COVID-Fact	Flan-T5-xxl (11B)	LoraConcat (56M)	11.191B	17h 5m / 3s
		LoraMap (4.4M)	11.196B	15h 22m / 2s
	LLaMA3 (8B)	LoraConcat (125M)	8.156B	2h 43m / 22s
		LoraMap (4.1M)	8.160B	5h 6m / 5s
SciFact	LLaMA3 (8B)	LoraConcat (125M)	8.156B	1h 41m / 41s
		LoraMap (4.1M)	8.160B	1h 50m / 5s

Table 7: Efficiency in terms of parameter and time.

5 Discussion

5.1 Design Motivation of LoraMap

The experimental findings highlight the significance of the integration strategies of multiple reasoning LoRAs. The main motivation for the LoraMap architecture is that the LoraHub linearly adds all trained LoRA weights. This linear approach can diminish the impact of individual matrix values due to the averaging effect, especially when there is a significant variation in weights according to the distinct roles of LoRAs. We believe that in the human brain, training does not occur through linear addition but rather domain-specific training to enhance the brain's functions for a specific task.

Additionally, the LoraConcat architecture may experience a loss of reasoning capability due to the catastrophic forgetting problem as the concatenated LoRA matrices undergo further fine-tuning. To address this, we design LoraMap to preserve these matrices and learn only the connections between LoRAs to support decision-making from diverse reasoning perspectives. As each brain region possesses different knowledge and functionalities, establishing interconnections among them would be important. Therefore, we train $A_{map}$and B_{map} matrices while maintaining areas for each distinct function, which are A_{cat} and B_{cat} matrices.

According to the design, the differences in the number of trainable parameters can influence performance. When combining three LoRAs, LoraHub only learns three coefficients per LoRA layer, which may not be sufficient for complex tasks. In contrast, LoraConcat learns $2 \times d \times 3r$ parameters, and LoraMap learns $2 \times 3r \times m$ parameters for each LoRA layer, resulting in better performance than LoraHub. Additionally, increasing the number of LoRAs can lead to substantial parameter growth with a fixed m. However, we can adjust m of LoraMap to modify the number of trainable parameters for efficient training.

5.2 Case Study

The cases where the output of LoraMap is correct, but zero-shot and LoraConcat are incorrect on the COVID-Fact and SciFact datasets are in Figure11 and 12, respectively. Even though the fact-checking labels only classify claims, LoraMap generates explanations along with classifications, possibly because it uses reasoning LoRAs. The details are in Appendix C.

5.3 Applying LoraMap to Other Tasks

LoraMap can be applied to other tasks but it requires some modifications. First, the reasoning LoRAs to integrate should be relevant to the downstream task. For a question-answering task, where a question and context are given for answering, the DifferenceCoT and EntityCoT could work as helper tasks, whereas ClaimCorrect may not be appropriate.

Second, the inclusion and training of new reasoning LoRA is available. For any new helper task, it is necessary to train a new LoRA and then subsequently train the LoraHub, LoraConcat, and LoraMap models. All these models need retraining to adjust the coefficient or corresponding matrix weights. We also considered predicting the triplets (entity-relation-entity) from the claim and context, but the poor performance of GPT-4 led to its exclusion from this study.

Third, if the researcher customizes five reasoning LoRAs, the LoraMap dimension changes. The original LoraMap matrices consist of A_{cat} ($\mathbb{R}^{d \times 3r}$), B_{cat} ($\mathbb{R}^{3r \times d}$), A_{map} ($\mathbb{R}^{3r \times m}$), B_{map} ($\mathbb{R}^{m \times 3r}$), and when employing five LoRAs, the dimension of $3r$ in each matrix changes to $5r$. As the number of LoRAs increases, so does the computational cost. Therefore, selecting LoRAs relevant to the downstream task and adjusting m would be necessary.

5.4 Reasoning Dataset Assessment

Two graduate students specializing in biomedical engineering perform a manual assessment to evaluate the quality of reasoning datasets generated by GPT-4. We randomly select 100 instances from each dataset for evaluation. The DifferenceCoT dataset shows an accuracy of 0.93, and the EntityCoT dataset shows an accuracy of 0.89. Additionally, we compute Cohen's kappa to assess the inter-rater reliability of manual evaluations. The kappa values for DifferenceCoT and EntityCoT are 0.8465 and 0.8629, respectively. Then, an emergency medicine clinician with over five years of experience reviews and confirms the dataset.

In DifferenceCoT, GPT-4 struggles with differentiating between claims and context, especially when dealing with numerical values. For example, it overlooks differences such as the claim mentioning two hours while the context refers to two weeks or the claim mentioning 1,000 people while the context refers to at least 1 percent of the population. Additionally, GPT-4 faces challenges due to a lack of biomedical knowledge, confusing bacterial viromes with human viromes.

In EntityCoT, GPT-4 incorrectly identifies distinct biomedical entities as synonymous, such as equating the 'n gene of sars-cov-2' with the 'n gene assay' and confusing 'covid-19 infection' from the claim with 'covid-19 vaccine prospects' from the context. It also fails to recognize some synonymous entities, such as 'sars-cov-2' in the claim and 'COVID-19' in the context. In one case, GPT-4 hallucinates by identifying an entity in the claim that is not present in the context.

6 Conclusion

This paper investigates methods for integrating multiple reasoning LoRAs. We generate three reasoning datasets and fine-tune individual LoRAs to facilitate inference from different perspectives. Subsequently, we introduce LoraMap, which learns the connection map between reasoning LoRAs. The fact-checking results show that LoraMap demonstrates superior performance and efficiency on the Flan-T5 and LLaMA3 models. Future work could explore generating not only claim classifications but also explanations, applying LoraMap to other tasks, and developing a method to select relevant LoRAs for a given task. We anticipate that this paper will pave the way for approaches to establish connections between LoRAs.

7 Limitations

Researchers should use GPT-4 or other APIs to generate reasoning datasets, which may incur some costs. Additionally, it is necessary to develop methods for evaluating the quality of GPT-4 reasoning and filtering data for training.

This paper focuses on fact-checking a single claim. In real-world scenarios, the outputs from LLM need to be verified. As traditional fact-checking models mostly verify a single claim, some research transforms LLM output into multiple claims. By verifying each claim and averaging their veracity, we can assess the reliability of the LLM outputs.

Our model is unsuitable for cases where only claims are present without evidence. In such cases, it is necessary to search for and provide appropriate evidence. There was no need to search for evidence in this work, as we use the claims with corresponding evidence. Furthermore, our model cannot make integrated judgments about multiple pieces of evidence.

Finally, it would be beneficial to examine LoraConcat and LoraMap across various open-source LLMs and other large fact-checking datasets within the biomedical and health domains.

References

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72, Prague, Czech Republic. Association for Computational Linguistics.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *Computing Research Repository*, arXiv:2004.05150. Version 2.

Eric Chamoun, Marzieh Saeidi, and Andreas Vlachos. 2023. Automated fact-checking in dialogue: Are specialized models needed? In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16009–16020, Singapore. Association for Computational Linguistics.

I-Chun Chern, Steffi Chern, Shiqi Chen, Weizhe Yuan, Kehua Feng, Chunting Zhou, Junxian He, Graham Neubig, and Pengfei Liu. 2023. Factool: Factuality detection in generative ai–a tool augmented framework for multi-task and multi-domain scenarios. *Computing Research Repository*, arXiv:2307.13528.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *Computing Research Repository*, arXiv:2210.11416.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. Llm.int8(): 8-bit matrix multiplication for transformers at scale. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2024. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36.

Shihan Dou, Enyu Zhou, Yan Liu, Songyang Gao, Jun Zhao, Wei Shen, Yuhao Zhou, Zhiheng Xi, Xiao Wang, Xiaoran Fan, Shiliang Pu, Jiang Zhu, Rui Zheng, Tao Gui, Qi Zhang, and Xuanjing Huang. 2023. Loramoe: Revolutionizing mixture of experts for maintaining world knowledge in language model alignment. *Computing Research Repository*, arXiv:2312.09979.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. 2023. Improving factuality and reasoning in language models through multiagent debate. *Computing Research Repository*, arXiv:2305.14325.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, and Aiesha Letman et al. 2023. The llama 3 herd of models. *Computing Research Repository*, arXiv:2407.21783.

Chongyang Gao, Kezhen Chen, Jinmeng Rao, Baochen Sun, Ruibo Liu, Daiyi Peng, Yawen Zhang, Xiaoyuan Guo, Jie Yang, and VS Subrahmanian. 2024. Higher layers need more lora experts. *Computing Research Repository*, arXiv:2402.08562.

Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. 2022. A survey on automated fact-checking. *Transactions of the Association for Computational Linguistics*, 10:178–206.

Prakhar Gupta, Chien-Sheng Wu, Wenhao Liu, and Caiming Xiong. 2022. Dialfact: A benchmark for fact-checking in dialogue. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3785–3801, Dublin, Ireland. Association for Computational Linguistics.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, Online. Association for Computational Linguistics.

Chengsong Huang, Qian Liu, Bill Yuchen Lin, Tianyu Pang, Chao Du, and Min Lin. 2023. Lorahub: Efficient cross-task generalization via dynamic lora composition. volume arXiv:2307.13269.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Neema Kotonya and Francesca Toni. 2020. Explainable automated fact-checking for public health claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7740–7754, Online. Association for Computational Linguistics.

Philippe Laban, Wojciech Kryściński, Divyansh Agarwal, Alexander R. Fabbri, Caiming Xiong, Shafiq Joty, and Chien-Sheng Wu. 2023. Llms as factual reasoners: Insights from existing benchmarks and beyond. *Computing Research Repository*, arXiv:2305.14540.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059. Association for Computational Linguistics.

Dengchun Li, Yingzi Ma, Naizheng Wang, Zhiyuan Cheng, Lei Duan, Jie Zuo, Cal Yang, and Mingjie Tang. 2024. Mixlora: Enhancing large language models fine-tuning with lora-based mixture of experts. *Computing Research Repository*, arXiv:2404.15159.

Miaoran Li, Baolin Peng, and Zhu Zhang. 2023. Self-checker: Plug-and-play modules for fact-checking with large language models. *Computing Research Repository*, arXiv:2305.14623.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597. Association for Computational Linguistics.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out, Association for Computational Linguistics*, pages 74–81.

Chin-Yew Lin and Franz Josef Och. 2004. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 605–612, Barcelona, Spain. Association for Computational Linguistics.

Qidong Liu, Xian Wu, Xiangyu Zhao, Yuanshao Zhu, Derong Xu, Feng Tian, and Yefeng Zheng. 2023. Moelora: An moe-based parameter efficient fine-tuning method for multi-task medical applications. *Computing Research Repository*, arXiv:2310.18339.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100. Association for Computational Linguistics.

OpenAI. 2023. Gpt-4 technical report. *Computing Research Repository*, arXiv:2303.08774.

Liangming Pan, Xinyuan Lu, Min-Yen Kan, and Preslav Nakov. 2023. Qacheck: A demonstration system for question-guided multi-hop fact-checking. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 264–273. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the*

40th annual meeting of the Association for Computational Linguistics, pages 311–318, Philadelphia, Pennsylvania. Association for Computational Linguistics.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020. Adapterhub: A framework for adapting transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 46–54. Association for Computational Linguistics.

Arkadiy Saakyan, Tuhin Chakrabarty, and Smaranda Muresan. 2021. Covid-fact: Fact extraction and verification of real-world claims on covid-19 pandemic. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1:Long Papers)*, pages 2116–2129, Online. Association for Computational Linguistics.

Mourad Sarrouti, Asma Ben Abacha, Yassine Mrabet, and Dina Demner-Fushman. 2021. Evidence-based fact-checking of health-related claims. In *In Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3499–3512, Punta Cana, Dominican Republic. Association for Computational Linguistics.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. Fever: a large-scale dataset for fact extraction and verification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *Computing Research Repository*, arXiv:2302.13971.

David Wadden, Shanchuan Lin, Kyle Lo, Lucy L. Wang, Madeleine V. Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7534–7550, Online. Association for Computational Linguistics.

David Wadden, Kyle Lo, Lucy Lu Wang, Arman Cohan, Iz Beltagy, and Hannaneh Hajishirzi. 2022. Multivers: Improving scientific claim verification with weak supervision and full-document context. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 61–76. Association for Computational Linguistics.

Cunxiang Wang, Xiaoze Liu, Yuanhao Yue, Xiangru Tang, Tianhang Zhang, Cheng Jiayang, Yunzhi Yao, Wenyang Gao, Xuming Hu, Zehan Qi, Yidong Wang, Linyi Yang, Jindong Wang, Xing Xie, Zheng Zhang, and Yue Zhang. 2023a. Survey on factuality in large language models: Knowledge, retrieval and domain-specificity. *Computing Research Repository*, arXiv:2310.07521.

Yuxia Wang, Revanth Gangi Reddy, Zain Muhammad Mujahid, Arnav Arora, Aleksandr Rubashevskii, Jiahui Geng, Osama Mohammed Afzal, Liangming Pan, Nadav Borenstein, Aditya Pillai, Isabelle Augenstein, Iryna Gurevych, and Preslav Nakov. 2023b. Factcheck-bench: Fine-grained evaluation benchmark for automatic fact-checkers. *Computing Research Repository*, arXiv:2311.09000.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. volume 35, pages 24824–24837, New Orleans, Louisiana, United States of America. NeurIPS.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *International Conference on Learning Representations*.

Xuan Zhang and Wei Gao. 2023. Towards llm-based fact verification on news claims with a hierarchical step-by-step prompting method. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 996–1011. Association for Computational Linguistics.

A Details of Datasets

A.1 Examples of Reasoning Datasets

We construct three reasoning datasets tailored for fact-checking: DifferenceCoT, EntityCoT, and CorrectClaim. Figure 4, 5, and 6 show the entire input and output of DifferenceCoT, EntityCoT, and CorrectClaim, respectively on the COVID-Fact dataset. For generating DifferenceCoT and EntityCoT, the input to GPT-4 encompasses task instruction, claim, and context, and the ground truth output is the GPT-4 result. For generating CorrectClaim, the input contains task instructions, claim, and context, and the ground truth output is the true claim for the given evidence. As SciFact dataset does not consistently offer both true and false claims for each evidence, we generate CorrectClaim using the GPT-4 API as shown in Figure 7.

DifferenceCoT

Instructions:
- Explain the difference between the Claim sentence and Context in one paragraph.
- Let's think step by step.

Claim: Sars-cov-2 suppress inflammatory responses and cell death through caspase-8 activation.
Context: 4 SARS-CoV-2 infection triggers apoptosis through caspase-8 activation. 2 SARS-CoV-2 infection induces caspase-8 activation to mediate pro-IL-1β processing. 2 SARS-CoV-2 infection induces caspase-8 activation Fig. SARS-CoV-2 infection induces the cell death through the activation of caspase-8. Here we report that SARS-CoV-2 infection activates caspase-8 to trigger cell apoptosis and inflammatory cytokine processing in the lung epithelial cells.
Explanation:

generate reasoning dataset using GPT-4

The claim and the context both discuss the role of caspase-8 activation in SARS-CoV-2 infection, but they present different perspectives. The claim suggests that SARS-CoV-2 suppresses inflammatory responses and cell death through caspase-8 activation. This implies that the virus uses caspase-8 activation as a mechanism to inhibit the body's immune response and prevent cell death, potentially aiding its survival and propagation. On the other hand, the context provides information that SARS-CoV-2 infection triggers apoptosis (cell death) and inflammatory cytokine processing through caspase-8 activation. This suggests that the virus's interaction with caspase-8 leads to cell death and inflammation, which are typical responses of the body's immune system to infection. Therefore, while both the claim and context agree on the involvement of caspase-8 in SARS-CoV-2 infection, they differ on whether this leads to suppression or induction of cell death and inflammation.

Figure 4: An example of DifferenceCoT dataset on the COVID-Fact.

EntityCoT

Instructions:
- Extract biomedical entities which are mentioned in both Claim and Context sentences and are synonymous.
- Output with the following format.
{'Claim': [entity list], 'Context': [entity list]}
- Let's think step by step and explain in one paragraph.

Claim: Sars-cov-2 suppress inflammatory responses and cell death through caspase-8 activation.
Context: 4 SARS-CoV-2 infection triggers apoptosis through caspase-8 activation. 2 SARS-CoV-2 infection induces caspase-8 activation to mediate pro-IL-1β processing. 2 SARS-CoV-2 infection induces caspase-8 activation Fig. SARS-CoV-2 infection induces the cell death through the activation of caspase-8. Here we report that SARS-CoV-2 infection activates caspase-8 to trigger cell apoptosis and inflammatory cytokine processing in the lung epithelial cells.
Explanation:

generate reasoning dataset using GPT-4

The Claim and Context sentences both mention the biomedical entities "Sars-cov-2", "inflammatory responses", "cell death", and "caspase-8 activation". In the context, "SARS-CoV-2 infection" is synonymous with "Sars-cov-2" from the claim. Similarly, "cell apoptosis" and "cell death" are synonymous terms. "Inflammatory cytokine processing" and "inflammatory responses" are also synonymous. Lastly, "caspase-8 activation" is mentioned in both sentences with the same meaning. Therefore, the extracted entities are "Sars-cov-2", "inflammatory responses", "cell death", and "caspase-8 activation".
The output in json format would be:
{
'Claim': ['Sars-cov-2', 'inflammatory responses', 'cell death', 'caspase-8 activation'],
'Context': ['SARS-CoV-2 infection', 'inflammatory cytokine processing', 'cell apoptosis', 'caspase-8 activation']
}

Figure 5: An example of EntityCoT dataset on the COVID-Fact.

CorrectClaim

Instructions:
- Revise the Claim sentence by referring to the Context.

Claim: Sars-cov-2 suppress inflammatory responses and cell death through caspase-8 activation.
Context: 4 SARS-CoV-2 infection triggers apoptosis through caspase-8 activation. 2 SARS-CoV-2 infection induces caspase-8 activation to mediate pro-IL-1β processing. 2 SARS-CoV-2 infection induces caspase-8 activation Fig. SARS-CoV-2 infection induces the cell death through the activation of caspase-8. Here we report that SARS-CoV-2 infection activates caspase-8 to trigger cell apoptosis and inflammatory cytokine processing in the lung epithelial cells.
Revised Claim:

Sars-cov-2 triggers inflammatory responses and cell death through caspase-8 activation.

Figure 6: An example of CorrectClaim dataset on the COVID-Fact.

CorrectClaim

Instructions:
- For the following Claim and Context pair, give a corrected claim that fixes the error in the original Claim.
- Let's think step by step and explain in one paragraph.

Claim: Amitriptyline ineffective as a treatment for chronic tension-type headaches.
Context: Amitriptyline is the medication of first choice in the treatment of chronic tension-type headache. The primary study endpoint was a reduction of at least 50% of the product of headache duration and frequency and a reduction of at least 50% in headache intensity. A reduction in headache duration and frequency of at least 50% was found in 39.4% on AO, in 25.4% on AM and in 26.6% on PL (PAO-PL = .1384, PAM-PL = 1.000, PAO-AM = .0973). A reduction in headache intensity of at least 50% was found in 31.8% on AO, in 26.9% on AM and in 26.6% on PL (PAO-PL = .5657, PAM-PL = 1.000, PAO-AM = .5715).
Explanation:

generate reasoning dataset using GPT-4

Corrected Claim: Amitriptyline is effective as a treatment for chronic tension-type headaches.

Explanation: The context provided indicates that Amitriptyline is the first choice of medication for treating chronic tension-type headaches. The study results show that there was a reduction in headache duration, frequency, and intensity in a significant percentage of patients who were on Amitriptyline. Therefore, the claim that Amitriptyline is ineffective is incorrect. The corrected claim should state that Amitriptyline is effective in treating chronic tension-type headaches.

Figure 7: An example of CorrectClaim dataset using GPT-4 on the SciFact.

A.2 Examples of Fact-Checking Datasets

The input and output of the fact-checking dataset varies depending on the settings. An example of using zero-shot CoT with GPT-4 API is shown in Figure 8. Figure 9 and Figure 10 show an example for fine-tuning the Flan-T5 model and LLaMA3 model, respectively.

B Experimental Settings

B.1 Fine-tuning Reasoning LoRAs

All experiments use two RTX 3090 GPUs to fine-tune the three reasoning LoRAs with a fixed seed of 42 for reproducibility. The LoRA and qLoRA configurations use 16 as the rank parameter, 32 as α, and 0.05 as dropout for all models. The batch size is 1, and the gradient accumulation step is 8.

For Flan-T5 models, we employ early stopping with a patience of 3, selecting the epoch yielding the best ROUGE-Lsum score on the development set throughout the 20 training epochs. The learning rate is $1e-3$, the warmup ratio is 0.1, and the weight decay is 0.01, and we adopt the adafactor optimizer coupled with a cosine scheduler. Depending on the dataset length, we set the maximum source and target length as 1200 and 512.

For the LLaMA3 model, we implement early stopping with a patience of 3, choosing the epoch that shows the lowest loss on the development set throughout 10 training epochs. Using the ROUGE-Lsum score instead of loss leads to overfitting. The learning rate is $2e-4$, the warmup ratio is 0.1, and the weight decay is 0.01, and we adopt the paged adamw 32bit optimizer alongside a cosine scheduler. Depending on the dataset length, we set the maximum length as 2048.

Table 8 presents the training time and inference time for each test instance. Although Flan-T5-xxl shows the best performance on COVID-Fact, its extensive training and inference times make LLaMA3-8B a more practical choice. Therefore, we experiment with LLaMA3-8B for SciFact.

B.2 Fine-tuning on Fact-Checking

All experiments use two RTX 3090 GPUs to fine-tune fact-checking with a fixed seed of 42 for reproducibility. The LoRA and qLoRA configurations use 16 as the rank parameter, 32 as α, and 0.05 as dropout for all models. For all models, we employ early stopping with a patience of 3, selecting the epoch yielding the best macro-f1 score on the development set.

Dataset	Reasoning LoRA	Model	Train/Test Time
COVID-Fact	DifferenceCoT	Flan-T5-large	6h 50m / 7s
		Flan-T5-xxl	35h 50m / 21s
		LLaMA3(8B)	2h 27m / 12s
	EntityCoT	Flan-T5-large	8h 8m / 7s
		Flan-T5-xxl	18h 35m / 4m 21s
		LLaMA3(8B)	2h 37m / 12s
	CorrectClaim	Flan-T5-large	3h 46m / 1s
		Flan-T5-xxl	16h 9m / 4s
		LLaMA3(8B)	1h 1m / 3s
SciFact	DifferenceCoT	LLaMA3(8B)	59m / 25s
	EntityCoT	LLaMA3(8B)	1h / 22s
	CorrectClaim	LLaMA3(8B)	53m / 21s

Table 8: Training and inference time comparison for reasoning LoRAs.

For the Flan-T5-large model, the learning rate is $1e-3$, the warmup ratio is 0.1, and the weight decay is 0.01. The training epoch is 20, and we adopt the adafactor optimizer coupled with a cosine scheduler. The batch size per device is 1, and the gradient accumulation step is 4. We set the maximum source and target length as 1200 and 512.

For the Flan-T5-xxl model, the learning rate is $3e-4$, the warmup ratio is 0.1, and the weight decay is 0.01. The training epoch is 10, and we adopt the adafactor optimizer coupled with a cosine scheduler. The batch size per device is 1, and the gradient accumulation step is 8. We set the maximum source and target length as 1200 and 512.

For the LLaMA3 model, the learning rate is $2e-4$, the warmup ratio is 0.1, and the weight decay is 0.01. The training epoch is 10, and we adopt the paged adamw 32bit optimizer along with a cosine scheduler. The batch size per device is 1, and the gradient accumulation step is 8. We set the maximum length as 1024.

C Case Study

The cases where the output of LoraMap is correct, but zero-shot and LoraConcat are incorrect on the COVID-Fact and SciFact datasets are shown in Figure11 and 12, respectively. While zero-shot only performs classification on the claim, LoraConcat and LoraMap tend to provide explanations, possibly due to the use of reasoning LoRAs. LoraConcat often attempts to incorrectly classify while explaining. It is quite surprising that LoraMap attempts to generate explanations, even though the trained label does not include an explanation part. In the future, it may be beneficial to use labels with explanations to improve model explainability.

Fact-checking using GPT-4 (zero-shot CoT)

Instructions:
You are a helpful assistant with biomedical expertise.
Instructions:
- Classify the Claim sentence only as 'TRUE' or 'FALSE' by referring to the Context. \
- Let's think step by step to explain why in one paragraph.
Claim: Sars-cov-2 suppress inflammatory responses and cell death through caspase-8 activation.
Context: 4 SARS-CoV-2 infection triggers apoptosis through caspase-8 activation. 2 SARS-CoV-2 infection induces caspase-8 activation to mediate pro-IL-1β processing. 2 SARS-CoV-2 infection induces caspase-8 activation Fig. SARS-CoV-2 infection induces the cell death through the activation of caspase-8. Here we report that SARS-CoV-2 infection activates caspase-8 to trigger cell apoptosis and inflammatory cytokine processing in the lung epithelial cells.
Explanation:

The claim that Sars-cov-2 triggers inflammatory responses and cell death through caspase-8 activation is TRUE. The context provided clearly states that SARS-CoV-2 infection activates caspase-8, which in turn triggers cell apoptosis (cell death) and inflammatory cytokine processing in lung epithelial cells. The context also mentions that SARS-CoV-2 infection induces pro-IL-1β processing, which is a part of the inflammatory response. Therefore, the claim is consistent with the information provided in the context.

Figure 8: An example of fact-checking dataset when using zero-shot CoT with GPT-4.

Fact-checking using Flan-T5

What is the class of the Claim by referring to the Context? Choose only from 'TRUE' or 'FALSE'.
Claim: Sars-cov-2 suppress inflammatory responses and cell death through caspase-8 activation.
Context: 4 SARS-CoV-2 infection triggers apoptosis through caspase-8 activation. 2 SARS-CoV-2 infection induces caspase-8 activation to mediate pro-IL-1β processing. 2 SARS-CoV-2 infection induces caspase-8 activation Fig. SARS-CoV-2 infection induces the cell death through the activation of caspase-8. Here we report that SARS-CoV-2 infection activates caspase-8 to trigger cell apoptosis and inflammatory cytokine processing in the lung epithelial cells.

The claim is TRUE.

Figure 9: An example of fact-checking dataset for fine-tuning Flan-T5 models.

Fact-checking using LLaMA3

What is the class of the Claim by referring to the Context? Choose only from 'TRUE' or 'FALSE'.
Claim: Sars-cov-2 suppress inflammatory responses and cell death through caspase-8 activation.
Context: 4 SARS-CoV-2 infection triggers apoptosis through caspase-8 activation. 2 SARS-CoV-2 infection induces caspase-8 activation to mediate pro-IL-1β processing. 2 SARS-CoV-2 infection induces caspase-8 activation Fig. SARS-CoV-2 infection induces the cell death through the activation of caspase-8. Here we report that SARS-CoV-2 infection activates caspase-8 to trigger cell apoptosis and inflammatory cytokine processing in the lung epithelial cells.

Based on the context, the given claim is TRUE.

Figure 10: An example of fact-checking dataset for fine-tuning LLaMA3 models.

Input	Zero-shot	LoraConcat	LoraMap	Label
What is the class of the Claim by referring to the Context? Choose only from 'TRUE' or 'FALSE'. **Claim**: Transmission of sars-cov-2 in domestic cats imposes a major bottleneck. **Context**: SARS-CoV-2 transmission is defined by a narrow bottleneck. Transmission of SARS-CoV-2 in cats involved a narrow bottleneck, with new infections founded by fewer than ten viruses. While estimates of the size of the SARS- CoV-2 transmission bottleneck remain highly discordant in humans, we find very narrow transmission bottlenecks in cats, involving transmission of only 2-5 viruses. **Answer**:	Based on the context, I would classify the claim as: TRUE	Based on the context, the given claim is TRUE. The given claim is TRUE. The context suggests that the transmission of sars-cov-2 in domestic cats imposes a major bottleneck. The context also provides a specific estimate of the size of the transmission bottleneck in cats, stating that new infections are founded by fewer than ten viruses. This is a TRUE claim.	Based on the context, the given claim is FALSE. The context states that the transmission of SARS-CoV-2 in cats involves a narrow bottleneck, not a major bottleneck.	Based on the context, the given claim is FALSE.
What is the class of the Claim by referring to the Context? Choose only from 'TRUE' or 'FALSE'. **Claim**: The institute for disease surveillance puts the cfr at 0.94 % **Context**: With this additional adjustment and using the same methodology and data from the original version of this report, the corrected estimate of the infection-fatality-ratio (IFR) is 0.94 (0.37, 2.9) percent. In this report, we use a small amount of linelist data with complete disease progression information from early in the outbreak to estimate the delays between case confirmation and death, and then the time and age-adjusted confirmed-CFR. **Answer**:	Based on the context, the class of the Claim is: TRUE	Based on the context, the given claim is TRUE. The institute for disease surveillance is a real organization that provides estimates of the case-fatality ratio (CFR), which is a measure of the proportion of deaths among all cases. The context claim is TRUE because it matches the given claim. The CFR is a measure of the proportion of deaths among all cases, and the given claim is TRUE.	Based on the context, the given claim is FALSE. The context mentions the infection-fatality-ratio (IFR) and confirmed-CFR, but not the CFR. The CFR is a different measure, and the context does not mention it being 0.94%.	Based on the context, the given claim is FALSE.
What is the class of the Claim by referring to the Context? Choose only from 'TRUE' or 'FALSE'. **Claim**: Protein structure and sequence reanalysis of 2019-ncov genome refutes snakes as its intermediate host and the unique similarity between its spike protein insertions and hiv-1 **Context**: In this Communication, we first analyzed two recent studies that concluded that snakes are the intermediate hosts of 2019-nCoV and that the 2019-nCoV spike protein insertions share a unique similarity to HIV-1. **Answer**:	Based on the context, I would classify the claim as: FALSE	Based on the context, the given claim is FALSE. The given claim is FALSE. The context does not support the claim that snakes are the intermediate host of 2019-nCoV. The context also does not support the claim that the 2019-nCoV spike protein insertions share a unique similarity to HIV-1. The context is FALSE.	Based on the context, the given claim is TRUE. The context states that the claim is about reanalyzing the protein structure and sequence of the 2019-nCoV genome, which refutes the idea that snakes are the intermediate hosts and the unique similarity between the spike protein insertions and HIV-1.	Based on the context, the given claim is TRUE.

Figure 11: Examples of outputs on the COVID-Fact dataset.

Input	Zero-shot	LoraConcat	LoraMap	Label
What is the class of the Claim by referring to the Context? Choose only from 'TRUE' or 'FALSE'. **Claim**: 10-20% of people with severe mental disorder receive no treatment in low and middle income countries. **Context**: Although disorder severity was correlated with probability of treatment in almost all countries, 35.5% to 50.3% of serious cases in developed countries and 76.3% to 85.4% in less-developed countries received no treatment in the 12 months before the interview. **Answer**:	TRUE	Based on the context, the given claim is TRUE.	Based on the context, the given claim is FALSE. The context states that 76.3% to 85.4% of serious cases in less-developed countries received no treatment, which is higher than the 10-20% mentioned in the claim.	Based on the context, the given claim is FALSE.
What is the class of the Claim by referring to the Context? Choose only from 'TRUE' or 'FALSE'. **Claim**: Healthy volunteers exhibit rapid and transient decrease of cellular ATP after being bolus-injected with fructose. **Context**: Liver ATP levels were similar in the 2 groups before fructose infusion and decreased similarly in both after fructose infusion (P = .01 vs initial ATP levels). **Answer**:	FALSE	Based on the context, the given claim is FALSE.	Based on the context, the given claim is TRUE. The context states that liver ATP levels decreased similarly in both groups after fructose infusion, which suggests a rapid and transient decrease of cellular ATP after being bolus-injected with fructose.	Based on the context, the given claim is TRUE.
What is the class of the Claim by referring to the Context? Choose only from 'TRUE' or 'FALSE'. **Claim**: The tissue surrounding the granuloma in an immune cell induces an inflammatory immune response. **Context**: Conversely, the tissue surrounding the caseum has a comparatively anti-inflammatory signature. **Answer**:	TRUE	Based on the context, the given claim is TRUE. The tissue surrounding the granuloma is indeed immune and induces an inflammatory immune response. The given context supports the given claim. Therefore, the given claim is TRUE.	Based on the context, the given claim is FALSE. The context states that the tissue surrounding the caseum has an anti-inflammatory signature, not an inflammatory one.	Based on the context, the given claim is FALSE.

Figure 12: Examples of outputs on the SciFact dataset.