LoraMap: Harnessing the Power of LoRA Connections

Anonymous ACL submission

Abstract

Large Language Models (LLMs) can benefit from mitigating hallucinations through fact-checking and overcoming substantial computational overhead with parameter-efficient techniques such as Low-Rank Adaptation (LoRA). While some studies have explored the parallel integration of multiple LoRAs, these approaches need attention to the connections between them. 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 on the fact-checking task demonstrate that the performance of LoraMap is superior to LoraHub, an existing LoRA composition method. LoraMap also outperforms with significantly fewer 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 stand-alone 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 the COVID-Fact dataset, which covers fact-checking related to the COVID-19 pandemic.

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). Furthermore, some studies have explored the integration of multiple task-specific LoRAs to address other 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 the methods of 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 connect 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 dataset demonstrate that LoraMap exhibits superior performance than LoraHub and also outperforms LoraConcat even with significantly fewer parameters.

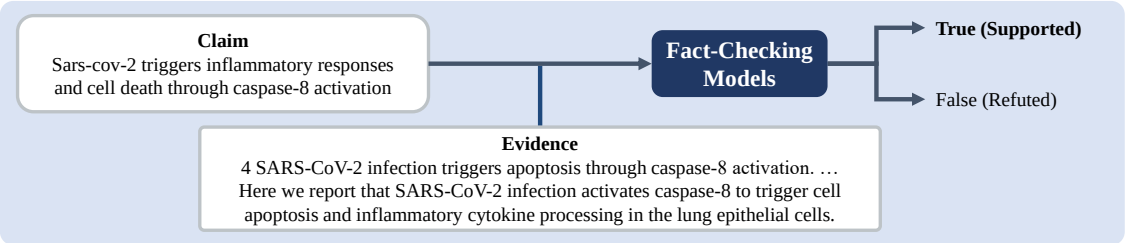

Figure 1: A fact-checking task classifies a claim as true or false based on the corresponding evidence.

2 Related Work

2.1 Biomedical Fact-Checking

Manual fact-checking has become challenging and time-consuming as biomedical literature rapidly expands. Several studies have attempted to construct biomedical fact-checking datasets and train various models. For the PubHealth dataset, the SciBERT model achieves the highest f1-score among the BERT models (Kotonya and Toni, 2020). For the SciFact (Wadden et al., 2020), the best model on the leaderboard[1] is MultiVerS (Wadden et al., 2022), a Longformer model (Beltagy et al., 2020) trained with rationale sentence selection and fact-checking label prediction. For the COVID-Fact, the RoBERTa model is fine-tuned on fact-checking and entailment inference datasets (Saakyan et al., 2021). For the HealthVer, the T5-base model performed better than the BERT models (Sarrouti et al., 2021).

2.2 Fact-Checking with LLMs

Recent studies have explored the potential of LLMs for general domain fact-checking through a zero-shot approach (Chern et al., 2023; Li et al., 2023; Wang et al., 2023b), hierarchical prompting (Zhang and Gao, 2023), multiagent debate approach (Du et al., 2023), question answering (Pan et al., 2023), and combining various language models (Min et al., 2023). The results demonstrate the effectiveness of using LLMs, but they are still room for improvement in factual reasoning (Laban et al., 2023; Wang et al., 2023a).

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 attention weight matrix into trainable low-rank matrices.

Another way to reduce computational requirements is by applying quantization to LLMs to reduce the numerical precision of model parameters. LLM.int8() (Dettmers et al., 2022) quantizes model weights to 8-bit integers through a vector-wise quantization and mixed-precision decomposition. QLoRA (Dettmers et al., 2024) employs three techniques, which are 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 where they differ. In this context, we hypothesize that identifying contrasting or common factors between the claim sentence and its corresponding evidence text can help the fact-checking model. Therefore, we customize the three reasoning tasks for fact-checking: DifferenceCoT, EntityCoT, and CorrectClaim.

DifferenceCoT is a task that generates a text that details the contextual differences between claim and evidence, such as relation, topic, and level of detail.

EntityCoT is a task that extracts synonymous biomedical entities that appear simultaneously in the claim sentence and the evidence text.

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

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

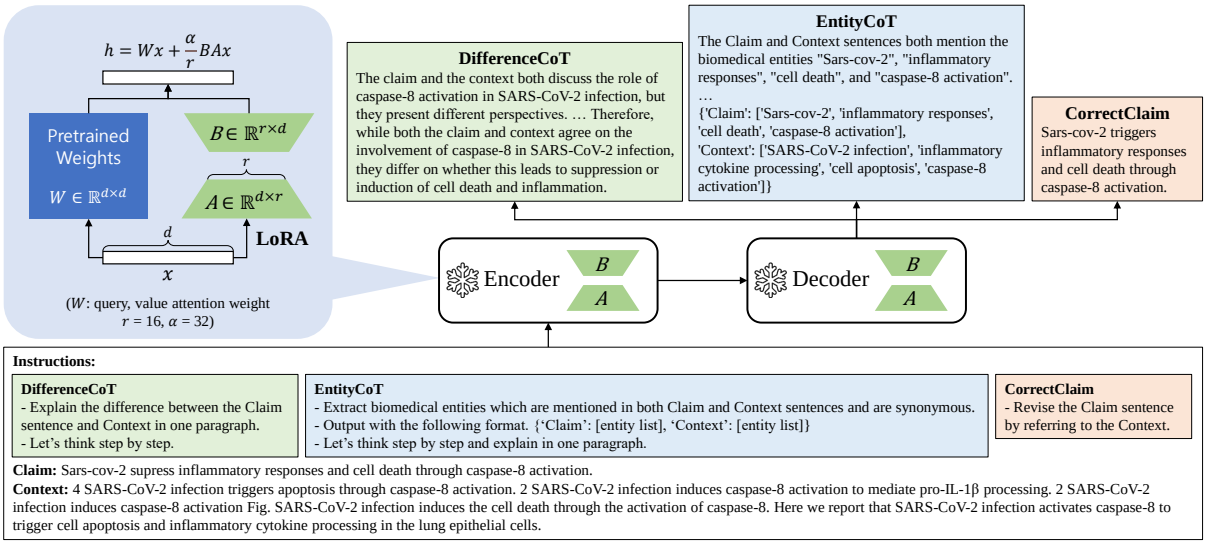

Figure 2: Example of reasoning datasets to fine-tune the base model with LoRA. The LoRA exists in the query and value parts of all transformer attention layers and consists of A and B weight matrices.

Next, we construct datasets for these three tasks. The COVID-Fact dataset contains at least one true claim and one false claim associated with each piece of evidence. First, we extract 2,550 claim-evidence pairs by randomly selecting two claims for each piece of evidence, one true and the other false, using a fixed seed of 42. Following the original dataset splits, we split them into 2,036 training instances, 258 development instances, and 256 test instances. The resulting dataset has an equal number of TRUE and FALSE classes across all splits.

For DifferenceCoT and EntityCoT, we employ Chain-of-Thought (CoT) prompting (Wei et al., 2022) with the GPT-4 API to generate output text. The prompt includes task instruction, claim, and context, which are input to GPT-4, and the ground truth output is the GPT-4 result as shown in Figure 2. On the contrary, we generate the CorrectClaim dataset from the COVID-Fact dataset by extracting claim-evidence pairs as inputs and assigning them to a true claim as the output. Whether the input claim is true or false, the ground truth output is always the true claim linked to the given evidence. Figure 2 shows an example of input and output text for a generative model, and Appendix A provides the entirely generated output text.

3.2 Fine-tuning Reasoning LoRAs

The next step is to fine-tune LoRAs for each task. We use Flan-T5 as the base model due to its range of model size options and its strong performance in zero-shot, few-shot, and CoT (Chung et al., 2022). The lightweight module LoRA exists in all transformer attention layers of the base model. Specifically, as shown in Figure 2, LoRA operates within the query and value parts of the encoder self-attention, decoder self-attention, and encoder-decoder attention layers. 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. By freezing the weights of the base model and training only the weights of LoRA, training requires much fewer parameters.

3.3 Connecting Reasoning LoRAs

The final step is to investigate methods for allocating and connecting the reasoning LoRAs, namely LoraHub, LoraConcat, and LoraMap. Figure 3 illustrates the differences among the methods.

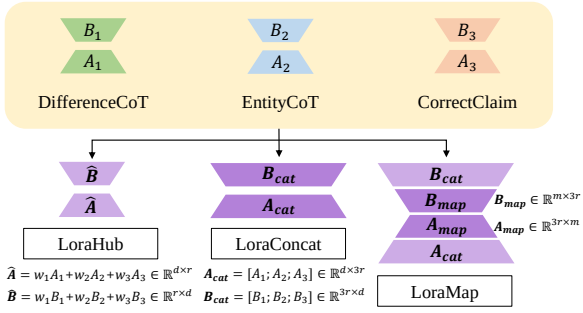

Figure 3: The comparison of LoraHub, LoraConcat, and LoraMap. Dark purple indicates trainable weights, and light purple represents fixed weights.

Base model	Reasoning LoRA	Setting	BLEU	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-Lsum	METEOR	BERTscore
Flan-T5-large (787M)	DifferenceCoT	Zero-shot	0.0023	0.2173	0.1326	0.1815	0.2011	0.1047	0.8563
		LoRA finetuning (4M)	**0.3588**	**0.6676**	**0.4206**	**0.5045**	**0.6310**	**0.5255**	**0.9275**
	EntityCoT	Zero-shot	0	0.0539	0.0201	0.0533	0.0526	0.0289	0.7997
		LoRA finetuning (4M)	**0.3885**	**0.6755**	**0.4533**	**0.5548**	**0.6397**	**0.5969**	**0.9240**
	CorrectClaim	Zero-shot	0.3636	0.6839	0.5714	0.6618	0.6636	0.6591	0.9349
		LoRA finetuning (4M)	**0.9257**	**0.9722**	**0.9437**	**0.9721**	**0.9721**	**0.9682**	**0.9944**
Flan-T5-xxl (11B)	DifferenceCoT	Zero-shot	0.0012	0.2034	0.1298	0.1718	0.1875	0.0945	0.8545
		qLoRA finetuning (18M)	**0.3764**	**0.6822**	**0.4446**	**0.5192**	**0.6444**	**0.5245**	**0.9315**
	EntityCoT	Zero-shot	0	0.0444	0.0238	0.0444	0.0442	0.0097	0.7903
		qLoRA finetuning (18M)	**0.3805**	**0.6680**	**0.4505**	**0.5500**	**0.6356**	**0.5881**	**0.9223**
	CorrectClaim	Zero-shot	0.5212	0.8102	0.7251	0.7985	0.7983	0.7821	0.9565
		qLoRA finetuning (18M)	**0.9227**	**0.9700**	**0.9389**	**0.9695**	**0.9696**	**0.9662**	**0.9943**

Table 1: The evaluation results on three reasoning test datasets. The bold text represents the best result.

LoraHub computes the weighted sum to generate $\hat{A} \in \mathbb{R}^{d\times r}$ and $\hat{B} \in \mathbb{R}^{r\times d}$. This framework freezes all A_t and B_t matrices and learns only the coefficients for each LoRA using a gradient-free approach. Our LoraHub loads three reasoning LoRAs along with the 20 LoRA modules following the original LoraHub setting[2].

LoraConcat concatenates the matrices A_t and B_t of the three reasoning LoRAs to produce $A_{cat} \in \mathbb{R}^{d\times 3r}$ and $B_{cat} \in \mathbb{R}^{3r\times d}$.

$$A_{cat} = [A_1; A_2; A_3]\ ,\ B_{cat} = [B_1; B_2; B_3]$$

We then fine-tune the A_{cat} and B_{cat} matrices targeting the COVID-Fact dataset. LoraMap not only concatenates the three reasoning LoRAs into A_{cat} and B_{cat} but also insert the trainable matrices $A_{map} \in \mathbb{R}^{3r\times m}$ and $B_{map} \in \mathbb{R}^{m\times 3r}$ between them. LoraMap freezes LoRAs that maintain specialized reasoning capabilities and learns the connection maps between them by fine-tuning only A_{map} and B_{map}. We define the mapping dimension m based on the ratio of trainable parameters to the total number of parameters in the model.

$$m = \frac{ratio \times num\ of\ total\ parameters}{3r \times num\ of\ trainable\ layers \times 100}$$

The number of trainable layers is the total number of layers of A_{map} and B_{map} in the model. The trainable parameters of this layer are $m \times 3r$.

4 Experimental Results

4.1 Reasoning LoRAs

We independently finetune DifferenceCoT LoRA, EntityCoT LoRA, and CorrectClaim LoRA inserted in the Flan-T5 models, using a fixed seed of 42 for reproducibility. The Flan-T5 model offers a range of options: small (77M), base (249M), large (787M), xl (3B), and xxl (11B). Among models below 1B, we experiment with Flan-T5-large; for models above 1B, we use Flan-T5-xxl model. The Flan-T5-large model with LoRA has a total of 787M parameters, with 4M trainable parameters. The Flan-T5-xxl model with qLoRA has 11B parameters and undergoes quantization using LLM.int8(), resulting in 18M trainable parameters.

Table 1 shows 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. 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 increases the scores of all metrics. Revising a claim is easier than capturing differences or identifying synonymous entities, so the CorrectClaim scores are considerably higher than others.

Our setup involves two RTX 3090 GPUs, and during Flan-T5-large LoRA fine-tuning, training takes 6 hours and 50 minutes for DifferenceCoT, 8 hours and 8 minutes for EntityCoT, and 3 hours and 46 minutes for CorrectClaim. For Flan-t5-xxl qLoRA fine-tuning, training takes 35 hours and 50 minutes for DifferenceCoT, 18 hours and 35 minutes for EntityCoT, and 16 hours and 9 minutes for CorrectClaim.

4.2 Connecting LoRAs for Fact-checking

We conduct experiments integrating multiple reasoning LoRAs on the COVID-Fact dataset. Given the prompt "*What is the class of the Claim by referring to the Context? Choose only from TRUE or FALSE.*" with claim and context, the output should be "*The claim is TRUE/FALSE*".

[2] https://github.com/sail-sg/lorahub

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 2: The evaluation results of the Flan-T5-large model on the COVID-Fact test dataset. In the fact-checking settings, the value in parenthesis indicates the number of trainable parameters. The bold text represents the best result. * is the size of all the training data.

4.2.1 Results of Small Language Model

The performance of the Flan-T5-large on the COVID-Fact test dataset is shown in Table 2. In the zero-shot setting, the Flan-T5-large model predominantly predicted TRUE with an f1 score of 0.5453. The key result is a comparison of connecting methods of multiple reasoning LoRAs: LoraHub, LoraConcat, and LoraMap. We experiment with various training instances, and Table 2 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. Although training with more than 100 instances is neither simple nor scalable, we aim 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 performed with ten fixed seeds (42, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384).

LoraHub achieves the highest f1-score of 0.6145 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. Notably, LoraConcat yields the best f1-score of 0.8126 at 1000-shot, and LoraMap achieves the highest f1-score of 0.8239 when using all instances. While fine-tuning the Flan-T5-large model, the mapping dimension m of LoraMap is set to 16, the same as the rank parameter of LoRA.

Figure 4 shows box plots for ten macro-f1 scores of LoraHub, LoraConcat, and LoraMap. LoraHub at 200-shot achieves an average score of 0.6145 with a median of 0.6212, and LoraConcat at 1000-shot reaches an average score of 0.8126 with a median of 0.8073. LoraMap exhibits an average of 0.8239 and a median of 0.8276 when using all training instances. LoraMap achieves statistically significant superior performance even with substantially fewer trainable parameters than LoraConcat, with a p-value of 0.03756. Additionally, comparisons between LoraHub and LoraMap and between LoraHub and LoraConcat also reveal statistically significant differences.

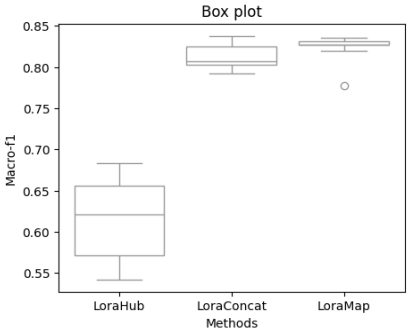

Figure 4: Box plots for macro-f1 scores of LoraHub, LoraConcat, and LoraMap.

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 containing 23 LoRAs. There are 14M trainable parameters out of 797.3M parameters when using LoraConcat and 0.22M 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.

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 3: 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 4: LoraMap results for the COVID-Fact test dataset depending on the selection of reasoning LoRAs.

4.2.2 Ablation Study

We further compare the results depending on the selection of LoRAs. Table 3 exhibits the results of LoraHub across different training instances, presenting zero-shot performance and the best result among 10-shot, 20-shot, 50-shot, and 100-shot learning. All experiments use a fixed seed of 42. The LoraHub originally uses 20 randomly selected LoRAs (base20), yielding 0.5423 under the zero-shot setting, which improves after fine-tuning 20 coefficients for each layer. When employing only three reasoning LoRAs, the zero-shot performance is higher than that of 20 random LoRAs. However, the performance does not improve while fine-tuning due to the difficulty of training only with three coefficient weights. We also experimented with three random LoRAs (base3) to verify this, and the results demonstrate the same tendency to struggle with fine-tuning. Consequently, we kept 20 random LoRAs and added three reasoning LoRAs, a setting that shows the best macro f1 score.

LoraHub outputs coefficients after training, which is the impact of each LoRA module. The coefficients for the three reasoning LoRAs are all close to 0.5, four out of the 20 base modules also exhibiting 0.5, mostly trained for question-answering, and the remaining 16 show values close to zero or negative. The coefficients confirm that our reasoning LoRAs play an important role in fact-checking.

Table 4 shows the results of the ablation study on LoraMap to show 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, which shows that the direction of identifying and correcting differences between claims and context is a more helpful task than finding synonymous entities within it.

4.2.3 Applicability to LLMs

Table 5 presents the performance of the LLMs on the COVID-Fact test dataset. In the zero-shot setting, the GPT-4 API with CoT prompting yields an f1 score of 0.6959, and the Flan-T5-xxl model exhibits an f1 score of 0.7021. The zero-shot prompt is shown in Appendix A. We compare LoraConcat and LoraMap when using all training instances for the Flan-T5-xxl. As previously mentioned, we define the mapping dimension m of A_{map} and B_{map} to scale the size of LoraMap according to the model size. Adjusting the ratio of trainable parameters to total parameters from 0.002 to 0.05 allows m to range from 16 to 400. The macro-f1 score generally improves as the size of LoraMap increases. LoraMap (4.4M) outperforms LoraConcat (56M) even with significantly fewer trainable parameters.

The total model parameters for LoraMap (4.4M) amount to 11.196B, while for LoraConcat (56M) the total is 11.191B. All experiments were performed with one RTX 3090 GPU. Training takes 17 hours and 5 minutes for LoraConcat (56M) and 15 hours and 22 minutes for LoraMap (4.4M). For each test instance, LoraConcat takes less than 3 seconds, and LoraMap takes less than 2 seconds.

4.3 Experimental Settings

When fine-tuning the three reasoning LoRAs, the experimental settings are identical, with a fixed seed 42. Depending on the dataset length, we set

Model	Reasoning LoRA	Fact-checking setting	ratio	m	Macro-precision	Macro-recall	Macro-f1
GPT-4	—	Zero-shot	—	—	0.7426	0.7070	0.6959
Flan-T5-xxl	—	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

Table 5: The evaluation results of the LLMs on the COVID-Fact test dataset. In the fact-checking settings, the value in parenthesis indicates the number of trainable parameters. The bold text represents the best result.

the maximum source length and target length as 1200 and 512, respectively. The LoRA and qLoRA configurations use 16 as the rank parameter and 32 as α. Throughout the 20 epochs of training, we employ early stopping with the patience of 3, selecting the epoch yielding the best ROUGE-Lsum score on the development set. 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. The batch size per device is 1, and the gradient accumulation step is 8. The experimental settings of fine-tuning Flan-T5-large on fact-checking are identical to fine-tuning the reasoning LoRAs, except that the gradient accumulation step is set to 4. When fine-tuning the Flan-T5-xxl model on fact-checking, the settings are also identical to finetuning the reasoning LoRAs, except that the epoch is ten.

5 Discussion

5.1 Design Motivation of LoraMap

The experimental findings highlight the significance of the connection and allocation strategies of multiple reasoning LoRAs. The main motivation for the LoraMap architecture is that the existing LoraHub linearly adds all trained LoRA weights. This linear approach can diminish the importance of matrix values due to the averaging effect, especially when weights vary significantly despite the distinct roles of LoRAs. We believe that in the human brain, training does not occur through linear addition but rather domain-specific training, enhancing the brain's functions for a specific task.

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, which preserves these matrices in their original states and learns only the connection mappings among LoRAs to facilitate decision-making from diverse reasoning perspectives. As each brain region possesses different knowledge and functionalities, establishing interconnections among them would be important. Therefore, we use the $A_{map} \in \mathbb{R}^{3r \times m}$ and $B_{map} \in \mathbb{R}^{m \times 3r}$ to train them on homogeneous functions while maintaining areas for each distinct function, which is the $A_{cat} \in \mathbb{R}^{d \times 3r}$ and $B_{cat} \in \mathbb{R}^{3r \times d}$.

Additionally, differences in the number of trainable parameters can also affect performance. When there are three LoRAs to combine, LoraHub learns only three coefficients, 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. When using fixed m, substantial parameter growth could pose a significant issue as the number of LoRAs increases. However, by specifying m of LoraMap, we can adjust the number of trainable parameters accordingly.

5.2 Applying LoraMap to Other Tasks

LoraMap can be applied to other tasks but needs some modifications. First, the reasoning LoRAs to combine 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 suitable.

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

Third, if the researcher customizes five reasoning LoRAs, the LoraMap matrix dimension changes. The original LoraMap matrices consist of A_{cat} matrix ($\mathbb{R}^{d\times 3r}$), B_{cat} matrix ($\mathbb{R}^{3r\times d}$), A_{map} matrix ($\mathbb{R}^{3r\times m}$), B_{map} matrix ($\mathbb{R}^{m\times 3r}$), and when employing five LoRAs, the dimension of $3r$ all transforms into $5r$. The more LoRA we use, the higher the computational cost will be. Therefore, selecting LoRAs relevant to the downstream task would be necessary.

5.3 Reasoning Dataset Assessment

Two graduate students manually evaluate the quality of the reasoning datasets generated by GPT-4. In order to evaluate the quality of the reasoning datasets generated by GPT-4, two graduate students perform a manual assessment. For each dataset, we randomly select 100 instances to evaluate. The DifferenceCoT dataset shows an accuracy of 0.93, and the EntityCoT dataset shows an accuracy of 0.89.

Analyzing the errors in DifferenceCoT, GPT-4 struggles to distinguish between claim and context when differing numerical values are mentioned. For example, it misses the difference when the claim states that the governor cancels school for at least two hours, but the context says it closes for at least two weeks. Likewise, it also misses when the claim mentions 1,000 people, but the context refers to at least 1 percent of the population. GPT-4 also fails due to a lack of biomedical knowledge, such as confusing bacterial viromes and human viromes as the same.

Analyzing the errors in EntityCoT, GPT-4 incorrectly identifies distinct biomedical entities as synonymous entities. For instance, it equates 'n gene of sars-cov-2' from the claim with 'N gene assay' from the context or equates 'covid-19 infection' from the claim with 'COVID-19 vaccine prospects' from the context. Additionally, it misses certain synonymous entities, such as 'sars-cov-2' in the claim and 'COVID-19' in the context. Lastly, GPT-4 shows one case of hallucination by identifying an entity that is present in the claim but absent in the context.

6 Conclusion

This paper investigates methods to establish connections among multiple reasoning LoRAs. We generate three reasoning datasets and fine-tune individual LoRAs to enable inference from different perspectives. Subsequently, we introduce LoraMap, an approach to learning the connection map between them. Our LoraMap statistically outperforms LoraHub and LoraConcat on the Flan-T5-large model. Even for the Flan-T5-xxl model, LoraMap outperforms LoraConcat even with significantly fewer parameters. We anticipate that this paper will pave the way for approaches in mapping and designing connections between LoRAs.

7 Limitations

There are true and false claims in the COVID-Fact dataset for each piece of evidence, so we automatically generate the CorrectClaim dataset. However, to apply this to other fact-checking datasets, researchers should consider the CoT prompting with GPT-4, similar to DifferenceCoT and EntityCoT. Additionally, it is essential to establish a method for assessing the quality of GPT-4 reasoning.

In real-world scenarios, the LLM outputs need to be verified. As traditional fact-checking models mostly verify a given claim, some research converts the LLM output into multiple claims. By verifying each claim and averaging its veracity, we can assess the reliability of the LLM outputs. Therefore, we focused on fact-checking stand-alone claims.

Our model is unsuitable for cases where only claims are present without evidence. In this case, appropriate evidence should be searched and provided. The COVID-Fact dataset contains evidence for each claim, so there was no need to search for evidence in this work. Making integrated judgments regarding multiple pieces of evidence is also impossible.

Finally, examining LoraConcat and LoraMap on various open-source LLMs and other fact-checking datasets in the biomedical and health domains is necessary.

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 spe-

cialized 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.

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 Zettle moyer, 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 Examples of Datasets

We constructed three reasoning datasets: DifferenceCoT, EntityCoT, and CorrectClaim. Figure 5, 6, and 7 show the input prompt and output of DifferenceCoT, EntityCoT, and CorrectClaim, respectively. For DifferenceCoT and EntityCoT, the prompt encompasses task instruction, claim, and context, which are input to GPT-4, and the ground truth output is the GPT-4 result. For CorrectClaim, the prompt contains task instructions, claim, and context, and the ground truth output is the true claim from the given evidence.

The prompt of fact-checking dataset varies depending on the settings. An example of a prompt when using zero-shot CoT with GPT-4 API is shown in Figure 8. Figure 9 shows an example prompt for fine-tuning the Flan-T5 model.

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 5: An example of DifferenceCoT dataset.

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 6: An example of EntityCoT dataset.

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 7: An example of CorrectClaim dataset.

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 prompt 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.