MAIN: Mutual Alignment Is Necessary for instruction tuning

Anonymous ACL submission

Abstract

Instruction tuning has enabled large language models (LLMs) to achieve remarkable performance, but its success heavily depends on the availability of large-scale, high-quality instruction–response pairs. However, current methods for scaling up data generation often overlook a crucial aspect: the alignment between instructions and responses. We hypothesize that high-quality instruction–response pairs are not defined by the individual quality of each component, but by the extent of their alignment with each other. To address this, we propose a Mutual Alignment Framework (MAIN) that ensures coherence between the instruction and response through mutual constraints. Experiments demonstrate that models such as LLaMA and Mistral, fine-tuned within this framework, outperform traditional methods across multiple benchmarks. This approach underscores the critical role of instruction-response alignment in enabling scalable and high-quality instruction tuning for LLMs.

1 Introduction

Large Language Models (LLMs) have demonstrated unprecedented capabilities in comprehending human intent and performing cross-task generalization through contextual learning(Brown et al., 2020). A key breakthrough in aligning model behaviors with human expectations is primarily attributed to instruction tuning, a supervised learning paradigm that bridges the gap between pretrained models' latent knowledge and explicit task requirements (Ouyang et al., 2022). Through multitask training on (instruction, response) pairs, this approach enables systematic knowledge elicitation while maintaining task-agnostic generalization (Chung et al., 2024). The effectiveness of this process is significantly influenced by the availability of high-quality instruction-response pairs at scale. In essence, the quality of data used in instruction tuning is critical to determining the performance and overall effectiveness of the model.

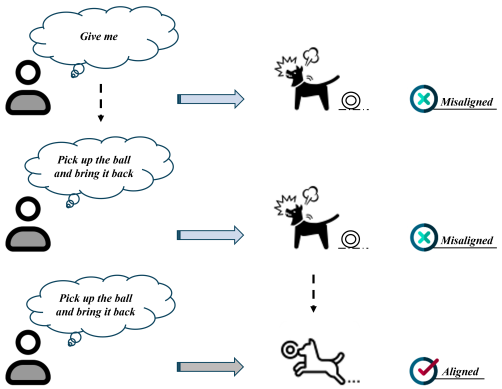

Figure 1: This figure illustrates a common interaction where a person and a dog adjust their behaviors to align instruction with response, evolving through repeated interactions to achieve mutual understanding.

Instruction-tuning methods currently follow two primary approaches. The first involves engaging domain experts (Köpf et al., 2024; Conover et al., 2023; Bach et al., 2022) to manually create instructions for specific tasks, ensuring high precision but facing challenges related to scalability and cost. The second approach (Wang et al., 2022a; Peng et al., 2023) leverages LLMs to generate responses based on given prompts. Although this approach is more scalable, it risks introducing inaccuracies or hallucinations (Zhang et al., 2023). Recent research has explored an alternative: leveraging human-written documents as typical responses and using LLMs to infer user instructions (Köksal et al., 2023; Li et al., 2023a; Chen et al., 2024; Nguyen et al., 2024), a process known as instruction back-translation. These approaches primarily focused on making the generated data resemble human data, without considering the inherent relationship between the instruction and the response. We contend that the alignment between the instruction and the

response is also essential.

As shown in Figure 1, the interaction between a person and a dog illustrates the bidirectional nature of training. Both the person and the dog adjust their behaviors to achieve mutual alignment. Similar to how a good command to a dog is one that elicits a proper response, generating an instruction-response pair must be aligned for optimal effectiveness. The quality of the instruction is validated by the response it triggers, and the same logic applies in reverse. Generating a high-quality pair requires careful alignment through mutual interaction. The instruction must clearly guide the response, while the response should accurately reflect the instruction, ensuring that both are mutually reinforcing.

The interdependence between instructions and responses introduces a dual-variable optimization problem, where optimizing one requires simultaneous consideration of the other, as neither can be fully optimized in isolation. Drawing inspiration from the alternating update strategy used in Expectation-Maximization (EM) algorithms (Moon, 1996), we propose a method to synthesize high-quality data, called **MAIN**. This framework iteratively optimizes both instructions and responses, enhancing their mutual alignment. Through this co-adaptive process, the alignment of instruction-response pairs improves progressively. We believe that the aligned pairs can significantly benefit model's capabilities. Furthermore, we propose a simple but effective filtering strategy, **mutual filter**, which selects pairs with superior alignment, ultimately boosting the quality of the fine-tuning dataset.

To validate the effectiveness of our proposed MAIN, we test model with our generated instruction-tuning data on multiple benchmark tasks. Our experiments demonstrate significant improvements in output quality, instruction-following, and reasoning ability. Specifically, for the LLaMA-2-7B model, our generated data yield a 5.85% improvement in output quality compared to the most competitive baseline, Dog Instruct (Chen et al., 2024), and a 3.60% increase in instruction following ability over Better Alignment (Nguyen et al., 2024). Experimental results show that our method substantially boosts the model's capabilities in various areas, surpassing traditional approaches. In summary, our main contributions are as follows:

- We emphasize the critical importance of mutual alignment between instructions and responses in synthesizing high-quality instruction-tuning data.
- Propose a mutual alignment framework that reinforces the inner connection between instructions and responses, and develop an efficient data filtering method based on this framework.
- Extensive experiments on multiple benchmark datasets show that our approach outperforms existing methods in enhancing instruction tuning effectiveness.

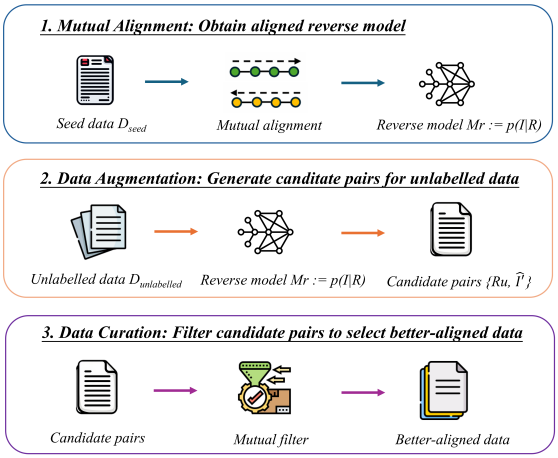

Figure 2: An overview of the data synthesis process, including mutual alignment, data augmentation, and data curation steps, aimed at creating high-quality, well-aligned instruction-response pairs from both seed and unlabelled data.

2 Methodology

In this section, we present our proposed Mutual Alignment Framework, designed to enhance instruction tuning performance by establishing and strengthening the intrinsic alignment between instructions and responses.

2.1 Preliminary

Data The framework utilizes two primary datasets: a limited set of high-quality, human-annotated instruction-response pairs seed data $D_{\text{seed}} = \{(I, R)\}$ and a larger collection of unlabelled responses $D_{\text{unlabelled}} = \{R_u\}$, extracted from web corpus.

Models The forward model $M_f := p(R|I)$ is designed to follow instructions, generating responses given instructions, while the reverse model $M_r :=$

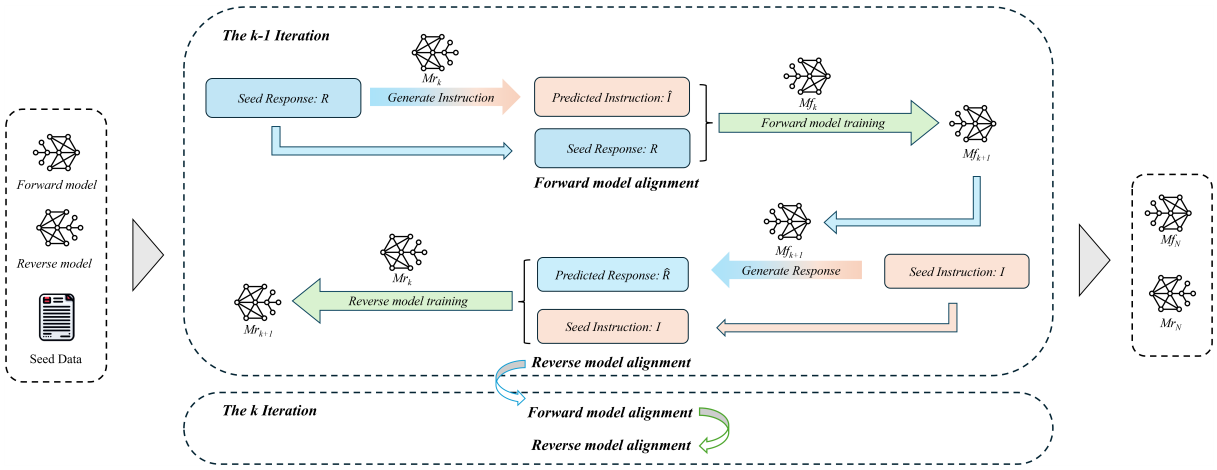

Figure 3: An overview of our method for iteratively aligning instructions and responses through mutual optimization.

$p(I|R)$ learns to generate instructions given responses.

2.2 Data Synthesis Framework: MAIN

We present our data synthesis process as shown in the Figure 2. Our Approach assumes access to a base language model, a small set of high-quality seed data, and a large collection of unlabelled responses. The overall preocess performs three core steps: **Mutual Alignment**, **Data Augmentation**, **Data Curation**.

- Mutual Alignment: This step is to obtain a reverse model $M_r := p(I|R)$ from the seed data D_{seed} based on the base model M_{base}. This step would align the internal relationship between instruction and response.

- Data Augmentation: This step is to generate candidate pairs from unlabelled data, which leverages the trained reverse model $M_r := p(I|R)$ to generate predicted instructions based on the unlabelled data $\mathcal{D}_{\text{unlabelled}}$, obtaining the augmented data pairs $\mathcal{D}_{\text{aug}} = \{R_u, \hat{I}'\}$.

- Data Curation: This step involves data filtering to obtain the curated, well-aligned pairs. In detail, we leverage our proposed **mutual-filter** mechanism to select high-quality samples from the augmented pairs $\mathcal{D}_{\text{aug}}$. These retained samples are used to form the final training dataset with seed data $\mathcal{D}_{\text{filter}} = \left[\text{filter}(\mathcal{D}_{\text{aug}}), \mathcal{D}_{\text{seed}}\right]$ for fine-tuning the base model.

2.3 Mutual Alignment

Achieving strong alignment between instructions and responses is critical for effective instruction tuning. However, establishing a robust relationship between these two components presents a challenging dual-variable problem, as neither direction can be optimized in isolation. Inspired by the iterative principles of the Expectation-Maximization algorithm, we propose mutual alignment that treats instruction-to-response and response-to-instruction generation as complementary tasks, modeled as a forward generation process and a reverse generation process, respectively. By alternately optimizing one direction while regulating the other, our method iteratively minimizes discrepancies until convergence is reached, ultimately yielding a model that produces highly aligned instruction–response pairs.

An overview of our approach is provided in Figure 3, and Algorithm 1 details the iterative optimization process.

Forward Model Alignment. To capture the alignment from responses to instructions, we let the forward model learn the distribution that the reverse model simulates. Specifically, at each iteration k, the reverse model generates synthetic instructions $\hat{I}$ for the responses, forming a target distribution that reflects how instructions should ideally relate to responses. The forward model is then trained to approximate this distribution.

$$\hat{I} = M_r^k(R), \quad \forall R \in \mathcal{D}_{\text{seed}}. \tag{1}$$

These synthetic pairs $(\hat{I}, R)$ are merged with the original seed data (I, R) to form the training set.

Algorithm 1 Mutual Alignment

Input: Seed data $\mathcal{D}_{\text{seed}} = \{(I, R)\}$,
Unlabelled data $\mathcal{D}_{\text{unlabelled}} = \{R_u\}$,
Base model M_{base},
Number of iterations N
Output: Final reverse model M_r^N, forward model M_f^N

Initialize forward model $M_f^0 \leftarrow M_{\text{base}}$;
Initialize reverse model $M_r^0 \leftarrow M_{\text{base}}$;
for $k = 0$ **to** N **do**
  Use M_r^k to generate $\hat{I}$ from R in $\mathcal{D}_{\text{seed}}$;
  Construct $\mathcal{D}_f = \{(\hat{I}, R)\} \cup \mathcal{D}_{\text{seed}}$;
  Update M_f^k on $\mathcal{D}_f$ by minimizing the loss $\mathcal{L}_f$ (Equation (2)) to obtain M_f^{k+1};
  Use M_f^{k+1} to generate $\hat{R}$ from I in $\mathcal{D}_{\text{seed}}$;
  Construct $\mathcal{D}_r = \{(\hat{R}, I)\} \cup \mathcal{D}_{\text{seed}}$;
  Update M_r^k on $\mathcal{D}_r$ by minimizing the loss $\mathcal{L}_r$ (Equation (4)) to obtain M_r^{k+1};
end for
Return M_r^N and M_f^N

The forward model is then updated to M_f^{k+1} by optimizing a weighted loss function:

$$\mathcal{L}_f = \alpha \cdot \mathcal{L}(\hat{I}, R) + (1 - \alpha) \cdot \mathcal{L}(I, R). \quad (2)$$

The first loss term $\mathcal{L}(\hat{I}, R)$ aligns the forward model with the synthetic instructions generated by the reverse model, ensuring that the forward model learns how responses correspond to instructions as modeled by the reverse model. The second loss term $\mathcal{L}(I, R)$ maintains consistency with the original human-annotated instructions, thereby preventing the forward model from overfitting to synthetic data. The parameter α controls the balance between synthetic and human-annotated instructions, with its dynamic adjustment described in Dynamic Weighting. This process encourages the forward model to adapt to the instruction distribution induced by the reverse model.

Reverse Model Alignment. Similarity, the reverse model is trained to capture the alignment from instruction to response as guided by the forward model. The reverse model now is updated based on the latest forward model M_f^{k+1} that generates synthetic responses $\hat{R}$ conditioned on the seed instructions:

$$\hat{R} = M_f^{k+1}(I), \quad \forall I \in \mathcal{D}_{\text{seed}}. \quad (3)$$

And it is optimized using similar weighted loss function:

$$\mathcal{L}_r = \alpha \cdot \mathcal{L}(\hat{R}, I) + (1 - \alpha) \cdot \mathcal{L}(R, I). \quad (4)$$

Dynamic Weighting The objective of dynamic weighting is to maintain a balanced contribution of synthetic and seed data, thereby improving the alignment between the forward and reverse models. The weighting parameter $\alpha \in [0, 1]$ is critical in this process, as it determines the relative influence of synthetic and seed data during fine-tuning. A static weighting scheme may result in suboptimal alignment: excessive reliance on synthetic data introduces noise, whereas an overemphasis on seed data can limit the model's generalization ability. To mitigate this, we employ a dynamic adjustment mechanism that updates α based on the relative loss contributions of synthetic and seed data at each step. Specifically, for forward model alignment, α is updated as: Å

$$\alpha = \frac{\mathcal{L}(\hat{I}, R)}{\mathcal{L}(\hat{I}, R) + \mathcal{L}(I, R)}. \quad (5)$$

This formulation adaptively adjusts the weight of synthetic data based on its relative loss, ensuring a balance between learning from synthetic examples and maintaining stability with seed data. By dynamically adjusting α, the model effectively integrates new information from synthetic pairs while maintaining consistency with the high-quality seed data.

2.4 Data Augmentation

After optimizing the mutual alignment between instructions and responses, we expand our training data by generating synthetic instructions for unlabelled responses. Specifically, for each unlabelled response $R_u \in \mathcal{D}_{\text{unlabelled}}$, the reverse model produces a corresponding synthetic instruction $\hat{I}'$. This yields candidate instruction–response pairs of the form $\{R_u, \hat{I}'\}$, which serve as approximations of how users might naturally formulate instructions for the given responses. However, not all candidate pairs are of high quality, so further curation is necessary.

2.5 Data Curation

To further improve data alignment, we introduce a simple yet effective filtering mechanism. We assume that high-quality instruction-response pairs

should be well-aligned, where the predicted instruction generated by the reverse model can be decoded by the forward model to recover the response, which should closely resemble the original. This process is akin to the interaction between an encoder and a decoder (Cho et al., 2014). Thus, we select the most well-aligned pairs. Using the candidate pairs $\{R_u, \hat{I}'\}$ from the Augmentation stage, we then employ the forward model to generate synthetic responses $\hat{R}'$ based on $\hat{I}'$:

$$\hat{R}' = M_f^N(\hat{I}'). \tag{6}$$

We compute the Cross-Entropy between the synthetic responses $\hat{R}'$ and the original unlabelled responses R_u:

$$\mathcal{L}_{\text{CE}}(\hat{R}', R_u) = -\sum_i \log p(\hat{R}' \mid \hat{I}', R_u). \tag{7}$$

Candidate pairs are sorted in ascending order by their values, and only those with the smallest values—indicating the highest degree of mutual alignment are retained.:

$$\mathcal{D}_{\text{filter}} = \left[\text{filter}(\mathcal{D}_{\text{aug}}), \mathcal{D}_{\text{seed}}\right] \tag{8}$$

This straightforward mechanism, relying solely on our mutual alignment model, effectively curates a high-quality subset of data for fine-tuning.

3 Experiment

3.1 Experimental Setup

Data. The seed data consists of 3,200 human-annotated (instruction, response) examples from the Open Assistant dataset (Köpf et al., 2024), serving as a reliable baseline for fine-tuning. The unlabelled data is Falcon RefinedWeb (Penedo et al., 2023) that is a massive English web dataset containing raw responses without paired instructions. We sampled 502k segments.

Mutual Alignment Framework. In the MAIN, we used the LLaMA-2-7B model (Touvron et al., 2023) as the base model for mutual alignment experiments, and additionally validated the generalization of our approach using the Mistral-7B-v1 model (Jiang et al., 2023). For each iteration, both the forward and reverse models are trained for one epoch. The learning rate is set to 1×10^{-5} with a linear decay schedule. The adaptation parameter, denoted as a, dynamically adjusts based on the formula in 5. The batch size is set to 32. For data curation, we selected the top 16,800 pairs from the unlabelled data based on cross-entropy and combined them with the seed data to create the final fine-tuning dataset.

Base model & fine-tuning. We use the pre-trained LLaMA-2-7B model and Mistral-7B-v1 model as the base models for fine-tuning respectively. Detailed hyperparameter configurations are provided in Appendix A.

3.2 Evaluation

To evaluate our framework, we conduct experiments across three benchmarks that assess different aspects of model performance.

AlpacaEval. We assess output preference using 805 instructions from the AlpacaEval dataset (Li et al., 2023b). Model outputs are compared against text-davinci-003 in a pairwise setting, with GPT-4-based judgments determining win rates.

IFEval. Instruction-following ability is evaluated with IFEval (Zhou et al., 2023), which reports accuracy across four metrics: Prompt-level Strict (P-S), Instruction-level Strict (I-S), Prompt-level Loose (P-L (Zhou et al., 2023), ensuring a comprehensive assessment of instruction adherence.

OpenLLM. Reasoning ability is measured via the Open LLM Leaderboard (Beeching et al., 2023) using the Language Model Evaluation Harness (Gao et al., 2023). We evaluate on ARC (Clark et al., 2018), HellaSwag (Zellers et al., 2019), Winogrande (Sakaguchi et al., 2021), MMLU (Hendrycks et al., 2020), and TruthfulQA (Lin et al., 2021).

3.3 Baselines

We compare our framework to several baseline approaches, all evaluated on the Falcon-RefinedWeb dataset with 20K samples.

Longform. This method (Köksal et al., 2023) prompts a large language model to generate instructions for human-written texts.

Humpback. This method (Li et al., 2023a) is a two-stage curation process that filters and selects high-quality instruction–response pairs before fine-tuning.

Dog Instruct. This method (Chen et al., 2024) involves a post-processing step that refines the responses to align with standard AI-generated output.

Better Alignment. This method (Nguyen et al., 2024) first generates instructions via back-translation, then filters low-quality pairs to obtain high-quality response.

4 Experimental Results

This section presents the experimental results, including quantitative evaluations, an ablation study, and a case study, to assess the effectiveness of our approach.

4.1 Quantitative Results

We conduct experiments across three benchmarks, each assessing a different aspect of the MAIN: AlpacaEval evaluates output quality, IFEval measures instruction-following ability, and OpenLLM tests reasoning capability.

Output Quality. As shown in Table 1, our method achieves the highest win rate in AlpacaEval dataset, surpassing the leading baseline. Specifically, on Llama-2-7B, our method achieves a win rate of 58.20%, representing a 5.85% improvement over the best baseline Dog Instruct (Chen et al., 2024). On Mistral-7B, our method outperforms Better alignment (Nguyen et al., 2024) by 3.15%, reaching a win rate of 48.94% compared to 45.79%. These results confirm that our MAIN method enhances instruction-response alignment more effectively than previous approaches, leading to outputs that better align with human expectations.

Instruction Following. Table 1 illustrates the performance of our method in IFEval dataset, where it outperforms previous approaches. Compared to the best-performing baseline Better alignment (Nguyen et al., 2024), our approach achieves consistent improvements across all evaluation metrics. Specifically, on Llama-2-7B, we see an increase of 2.59% in P-S and 3.42% in I-S. For Mistral-7B, we observe a 5.12% improvement in P-S and a 4.84% improvement in I-S over Better alignment (Nguyen et al., 2024). These results highlight the crucial role of enhanced data alignment in fine-tuning, which allows our model to better interpret and respond to user instructions, thereby driving its superior performance in instruction-following tasks.

Reasoning Ability. As shown in Table 1, our method demonstrates strong improvements in reasoning and factual accuracy across multiple downstream tasks. On Llama-2-7B, our approach shows a 2.02% improvement over the best baseline, Better Alignment, on ARC-Challenge and a 1.63% improvement on TruthfulQA. In particular, ARC-Challenge benefits from our method's ability to better capture common-sense reasoning patterns, which likely leads to more accurate responses. On Mistral-7B, the most significant improvements are observed in TruthfulQA, where our method outperforms Better Alignment by 5.03%, and in MMLU, with a 2.65% increase. These benchmarks, which require accurate factual recall and complex reasoning, show how our method strengthens the model's ability to provide correct and contextually appropriate answers.

These results underline the effectiveness of our MAIN in enhancing both reasoning and factual accuracy. By refining the alignment of instruction and response pairs during fine-tuning, our model is better equipped to handle complex reasoning tasks and provide more precise, reliable outputs across various challenging benchmarks.

4.2 Ablation Study

We perform further ablation studies to analyze the effect of filtering stragety and dynamic weighting.

Filtering Stragety. Effective filtering plays a crucial role in improving alignment quality by removing noisy or misaligned instruction-response pairs. Table 2 presents a results comparing models trained with no filtering, score-based filtering, and our mutual-filter approach. Our mutual-filter method achieves the highest win rate, surpassing both no filtering and score-based filtering. It consistently improves instruction-following accuracy, demonstrating that our simple mutual-filter approach, without additional model-based scoring, effectively selects high-quality instruction-response pairs for fine-tuning.

In contrast, score-based filtering provides little benefit and even underperforms compared to unfiltered data. This is due to the score-based approach, used in Humpback (Li et al., 2023a) and Better Alignment (Nguyen et al., 2024), relying on a ranking model fine-tuned on seed data rather than a dedicated scoring model. Without a clear optimization objective for instruction-response alignment, it struggles to identify high-quality pairs, leading to suboptimal fine-tuning.

Our mutual filter, by leveraging mutual-alignment models, directly favors instruction-response pairs with strong semantic coherence. By

Base Model	Method	Output quality	Instruction following				Reasoning ability				
		AlpacaEval	IFEval				ARC_C	MMLU	HellaSwag	Winogrande	TruthfulQA
			P-S	I-S	P-L	I-L					
Llama-2-7B	Humpback (Li et al., 2023a)	41.02	15.46	18.39	26.42	29.91	55.90	44.91	79.42	73.32	44.48
	Longform (Köksal et al., 2023)	35.64	15.23	17.56	26.10	29.29	55.72	45.02	78.98	73.21	45.07
	Dog Instruct(Chen et al., 2024)	52.35	15.52	19.40	28.17	32.01	56.06	45.62	79.89	74.13	45.77
	Better Alignment (Nguyen et al., 2024)	50.37	16.82	19.69	27.70	31.52	55.92	**45.84**	80.33	74.12	45.30
	MAIN	**58.20**	**20.22**	**23.36**	**31.17**	**35.37**	**57.08**	45.47	**81.22**	**74.51**	**47.40**
	Δ over Best Result	+5.85	+3.40	+3.67	+3.00	+3.36	+1.02	-0.37	+0.89	+0.38	+1.63
Mistral-7B	Humpback (Li et al., 2023a)	40.48	17.19	28.05	20.88	32.37	54.01	49.26	79.12	**73.24**	45.48
	Longform (Köksal et al., 2023)	37.62	16.98	27.89	20.75	32.10	53.98	48.12	78.25	71.60	44.88
	Dog Instruct(Chen et al., 2024)	45.34	18.23	28.48	21.32	33.47	53.15	49.10	79.07	73.21	45.98
	Better Alignment(Nguyen et al., 2024)	45.79	18.35	29.45	21.49	34.10	54.20	50.28	78.37	71.48	44.73
	MAIN	**48.94**	**23.47**	**34.29**	**26.60**	**38.84**	**55.12**	**52.93**	**79.38**	72.38	**49.76**
	Δ over Best Result	+3.15	+5.12	+4.84	+5.11	+4.74	+0.92	+2.65	+0.31	-0.86	+3.78

Table 1: Benchmarking results of different instruction tuning methods on Llama-2-7B and Mistral-7B using Falcon RefinedWeb dataset given same data quantity. Δ over Best Result quantify improvements relative to the strongest baseline method across evaluation categories.

eliminating misaligned samples without requiring additional supervision, it ensures a more effective training dataset, resulting in improved instruction-following and generalization capabilities.

Filtering Method	Win Rate	P-S	I-S	P-L	I-L
Ours w/o filtering	56.40	19.41	23.11	29.74	34.17
Ours + score-Based filtering	55.26	17.63	20.21	29.11	33.72
Ours + mutual filter	58.20	20.22	23.36	31.17	35.37

Table 2: Performance evaluation of LLaMA-2-7B fine-tuned on the Falcon-RefinedWeb dataset (20K samples) under three filtering conditions: no filtering, score-based filtering, and our proposed mutual filter. All conditions operate on instruction-response pairs generated by the same reverse model.

Dynamic Weighting. Balancing the contribution of aligned instruction-response pairs is crucial for achieving both strong alignment and robust generalization. The weighting parameter α controls this balance during training by adjusting the relative influence of synthetic and seed data. To evaluate its effectiveness, we compare fixed values ($\alpha = 0.3, 0.5, 0.7, 0.8, 1.0$) with our adaptive approach ($\alpha =$ Dynamic), which continuously updates α throughout training.

As shown in Figure 4, increasing α generally improves instruction-following ability and output quality by emphasizing well-aligned pairs. However, excessively high α makes the model overly reliant on generated instruction-response pairs, leading to unstable training and degraded performance.

To mitigate this, our dynamic weighting strategy adaptively balances aligned and seed data, preventing instability while maintaining strong alignment. The results show that this approach significantly improves output quality and instruction-following.

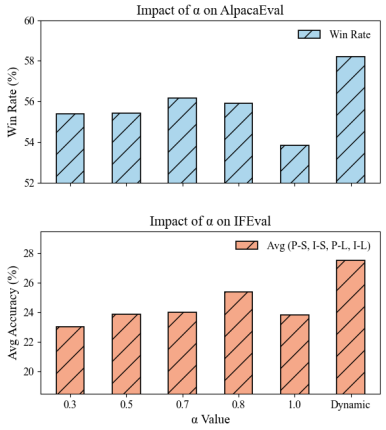

Figure 4: Evaluation of dynamic weighting strategies on LLaMA-2-7B training, comparing fixed and adaptive α values using the Falcon-RefinedWeb dataset, with performance assessed on AlpacaEval and IFEval.

4.3 Case Study

As shown in Figure 5, the two examples, both extracted from unlabeled data, illustrate the effectiveness of our approach. In the first case, the MAIN instruction explicitly requests specific details about the victim, the shooter, and the events surrounding the shooting, providing clear guidance for the response. In contrast, the baseline instruction is more general, only asking for a brief article about the shooting without specifying key details. In the second case, the MAIN instruction emphasizes a critical event: a wave of car burglaries in the suburbs, while the baseline instruction remains vague, simply requesting a summary of events in suburban areas.

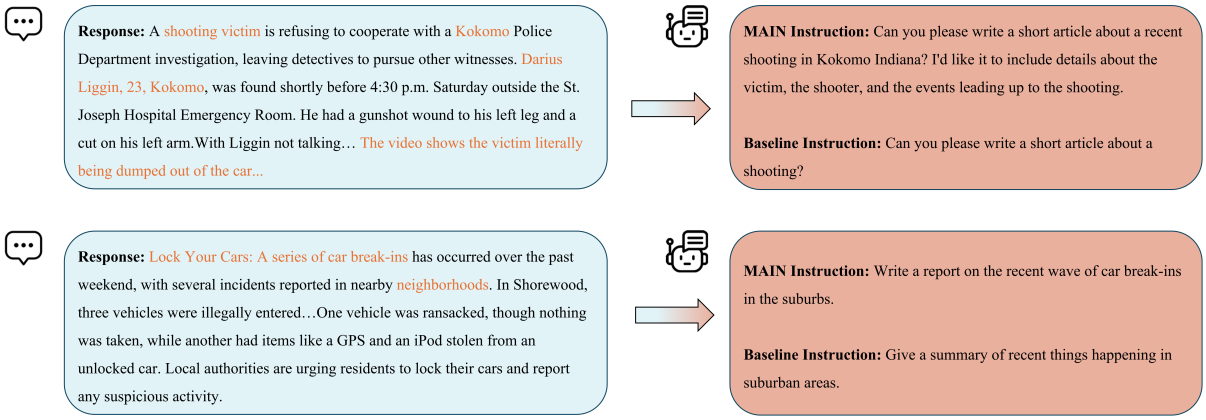

Figure 5: Method Comparison for Instruction Generation: A Case Study on the Effectiveness of Reverse Model Approaches in Aligning Instructions with Responses

In both cases, the reverse model's MAIN instructions are more focused and specific, resulting in responses that are better aligned with the intended context. In contrast, the baseline instructions are more general. These examples demonstrate that our method generates instructions that are more closely aligned with the responses, improving instruction-response coherence and enhancing the overall quality of the generated content.

5 Related Work

5.1 Instruction Tuning

Instruction tuning fine-tunes pre-trained LLMs on instruction-response pairs, enabling models to generalize across tasks without task-specific fine-tuning (Wei et al., 2021; Mishra et al., 2021; Wang et al., 2022b). Subsequent work (Mishra et al., 2021; Sanh et al., 2021) focused on cross-task generalization through diverse inputs.

5.2 Data Generation

Effective instruction tuning relies on large-scale, high-quality datasets, typically generated in two ways: human-crafted or model-generated.

Human-Crafted Data Datasets curated by domain experts, like OpenAssistant Conversations (Köpf et al., 2024) and Databricks Dolly-15k (Conover et al., 2023), are high quality but costly. Crowdsourced platforms like ShareGPT (Chiang et al., 2023) also contribute valuable data, especially user-uploaded conversations.

Model-Generated Data To reduce manual annotation costs, methods like Self-Instruct (Wang et al., 2022a) and Alpaca-GPT4 (Peng et al., 2023) generate instruction-response pairs automatically. However, issues like hallucinations (Zhang et al., 2023) persist. New approaches, such as Better Alignment (Nguyen et al., 2024) and Dog-Instruct(Chen et al., 2024), pair human responses with inferred instructions to reduce hallucinations and improve scalability. Our proposed MAIN builds on this by iteratively optimizing instruction-response alignment to ensure high-quality data.

6 Conclusion

In this paper, we redefine the alignment between instructions and responses, emphasizing its crucial role in optimizing instruction tuning for LLMs. By introducing the Mutual Alignment Framework, we present an innovative approach that iteratively optimizes both instructions and responses to enhance their alignment. Additionally, we propose mutual filter, a simple yet effective method for selecting instruction-response pairs with superior alignment. Our experimental results demonstrate that models fine-tuned within this framework outperform baselines across multiple evaluation benchmarks, including AlpacaEval, IFEval, and OpenLLM. These findings highlight the importance of mutual alignment in instruction tuning and how it can optimize fine-tuning data to enhance model performance. In conclusion, this work offers valuable insights for refining instruction-response pairs, enabling more efficient and scalable instruction tuning in future LLM developments.

Limitations

The mutual alignment approach requires more computational resources and time compared to traditional supervised fine-tuning methods. Specifically, the iterative optimization process can be computationally intensive, particularly for very large models. Due to these resource demands, we have evaluated the effectiveness of our method only on the LLaMA-2-7B and Mistral-7B models. In future work, we will explore the scalability of our approach across larger models and a wider range of datasets to assess its generalizability and applicability. Additionally, efforts will be made to develop more efficient algorithms for optimizing instruction-response alignment and adapting the framework to various tasks and domains.

References

Stephen H Bach, Victor Sanh, Zheng-Xin Yong, Albert Webson, Colin Raffel, Nihal V Nayak, Abheesht Sharma, Taewoon Kim, M Saiful Bari, Thibault Fevry, et al. 2022. Promptsource: An integrated development environment and repository for natural language prompts. *arXiv preprint arXiv:2202.01279*.

Edward Beeching, Clémentine Fourrier, Nathan Habib, Sheon Han, Nathan Lambert, Nazneen Rajani, Omar Sanseviero, Lewis Tunstall, and Thomas Wolf. 2023. Open llm leaderboard (2023-2024). https://huggingface.co/spaces/open-llm-leaderboard-old/open_llm_leaderboard.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Yongrui Chen, Haiyun Jiang, Xinting Huang, Shuming Shi, and Guilin Qi. 2024. Dog-instruct: Towards premium instruction-tuning data via text-grounded instruction wrapping. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4125–4135.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2(3):6.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. Free dolly: Introducing the world's first truly open instruction-tuned llm. *Company Blog of Databricks*.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2023. A framework for few-shot language model evaluation.

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

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Abdullatif Köksal, Timo Schick, Anna Korhonen, and Hinrich Schütze. 2023. Longform: Effective instruction tuning with reverse instructions. *arXiv preprint arXiv:2304.08460*.

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi Rui Tam, Keith Stevens, Abdullah Barhoum, Duc Nguyen, Oliver Stanley, Richárd Nagyfi, et al. 2024. Openassistant conversations-democratizing large language model alignment. *Advances in Neural Information Processing Systems*, 36.

Xian Li, Ping Yu, Chunting Zhou, Timo Schick, Omer Levy, Luke Zettlemoyer, Jason Weston, and Mike Lewis. 2023a. Self-alignment with instruction backtranslation. *arXiv preprint arXiv:2308.06259*.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023b. Alpacaeval: An automatic evaluator of instruction-following models.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2021. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2021. Cross-task generalization via natural language crowdsourcing instructions. *arXiv preprint arXiv:2104.08773*.

Todd K Moon. 1996. The expectation-maximization algorithm. *IEEE Signal processing magazine*, 13(6):47–60.

Thao Nguyen, Jeffrey Li, Sewoong Oh, Ludwig Schmidt, Jason Weston, Luke Zettlemoyer, and Xian Li. 2024. Better alignment with instruction back-and-forth translation. *arXiv preprint arXiv:2408.04614*.

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

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The refinedweb dataset for falcon llm: outperforming curated corpora with web data, and web data only. *arXiv preprint arXiv:2306.01116*.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. 2021. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*.

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

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022a. Self-instruct: Aligning language models with self-generated instructions. *arXiv preprint arXiv:2212.10560*.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, et al. 2022b. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. *arXiv preprint arXiv:2204.07705*.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*.

Muru Zhang, Ofir Press, William Merrill, Alisa Liu, and Noah A Smith. 2023. How language model hallucinations can snowball. *arXiv preprint arXiv:2305.13534*.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2024. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*.

A Training Details

Hyperparameter	Assignment
Computing Infrastructure	8 A100-80GB GPUs
Number of epochs	2
Batch size per GPU	64
Maximum sequence length	1024
Maximum learning rate	2e-5
Optimizer	Adam
Adam epsilon	1e-8
Adam beta weights	0.9, 0.999
Learning rate scheduler	warmup linear
Weight decay	0.1
Warmup steps	100
Learning rate decay	linear

Table 3: Hyperparameters used in the experiments.

Training is conducted with hyperparameters aligned to established supervised fine-tuning (SFT) practices (Zhou et al., 2024; Touvron et al., 2023). The learning rate is set to 2×10^{-5}, with a weight decay of 0.1, a batch size of 64, and a dropout rate of 0.1. Additionally, each iterative phase of training is limited to one epoch. For text generation,

we apply nucleus sampling (Holtzman et al., 2019) with a temperature (T) of 0.7 and a top-p value of 0.9. These settings balance diversity and relevance in the generated outputs. More hyperparameters listed in Table 3

B Ablation of Iteration.

To analyze the impact of iteration count N, we vary N from 1 to 20 and evaluate its effect on AlpacaEval and IFEval dataset. As shown in Table 4, increasing N initially improves performance, as iterative refinement enables the forward and reverse models to progressively align their outputs, enhancing instruction-response consistency.

However, beyond a certain point, performance begins to decline. Excessive iterations reinforce suboptimal patterns leading to overfitting. This underscores the necessity of selecting an optimal N that balances refinement and generalization. Our results emphasize the importance of properly tuning N to maximize the benefits of mutual alignment.

Iterations N	**Win Rate**
$N = 1$	50.11
$N = 2$	55.72
$N = 3$	58.20
$N = 4$	55.89
$N = 5$	55.60
$N = 10$	54.41
$N = 15$	54.29
$N = 20$	54.52

Table 4: Ablation study on the effect of iteration count N. We analyze the influence of varying the number of training iterations ($N = 1, 2, 3, 4, 5, 10, 15, 20$) on Llama-2-7B fine-tuned on Falcon-RefinedWeb.