Rationalize and Align: Enhancing Writing Assistance with Rationale via Self-Training for Improved Alignment

Hannan Cao, Hai Ye , Hwee Tou Ng

National University of Singapore

Outline

- Motivation
- Methods
- Experimental Results
- Conclusion

Motivation

Writing Assistant (WA)

- A system that provides writing suggestions based on user instructions.
- Covers a variety of writing-related tasks, including but not limited to grammatical error correction, text simplification, and style transfer.

Motivation

State-of-the-art (SOTA) WA

- Built using Supervised Fine-tuning (SFT) on labeled instruction data.
- Text editing allows multiple valid revisions for a given input
  - Just using SFT may fail to capture the flexibility of text revision (Paulus et al., 2018).
  - However, evaluation metrics (e.g., SARI) may capture this (Paulus et al., 2018).

Romain Paulus, Caiming Xiong, and Richard Socher. *A deep reinforced model for abstractive summarization. In ICLR 2018*

Motivation

SOTA WA

- Lacks the capability to generate proper rationales (linguistic explanations) for its generated suggestions.
  - Cannot assist user in validating and learning from its suggestions.

Motivation

Want to build a WA

- ➢ Aligns better to the suggestions with higher overall quality (e.g., fluency, coherence)
- ➢ Has the capability to generate rationales.

However, there is a lack of both preference data and rationale data in writing-related tasks.

Method

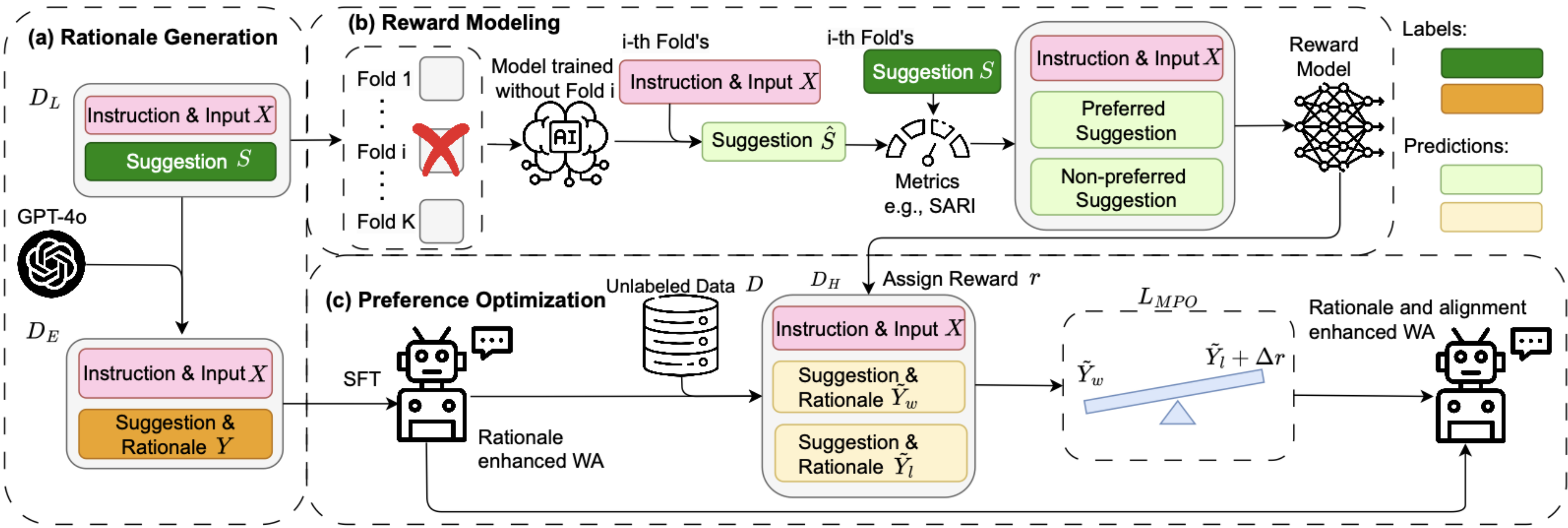
Propose a **Rationalize & Align** framework to enhance WA, consisting of:

- Rationale generation
- Self-training alignment:
  - Reward Modeling
  - Preference Optimization

Method

Overview:

a) Rationale Generation
b) Reward Modeling
c) Preference Optimization



Method

Rationale Generation:

- Extract Rationale from GPT-4o.
  - Provide the input, output, and edits.
  - Edits: modifications that transform the input to the output.

You are given a pair of English sentences along with a list of atomic edits. For each edit, the first word identifies content in the source sentence that is less appropriate, while the second word suggests a better phrase in the target sentence. `[Task Instruction]` Please generate a succinct explanation for each edit using the following template:

The word X should be deleted/inserted/replaced by Y because ...

Source sentence:
`[Input Text]`

Target sentence:
`[Output Text]`

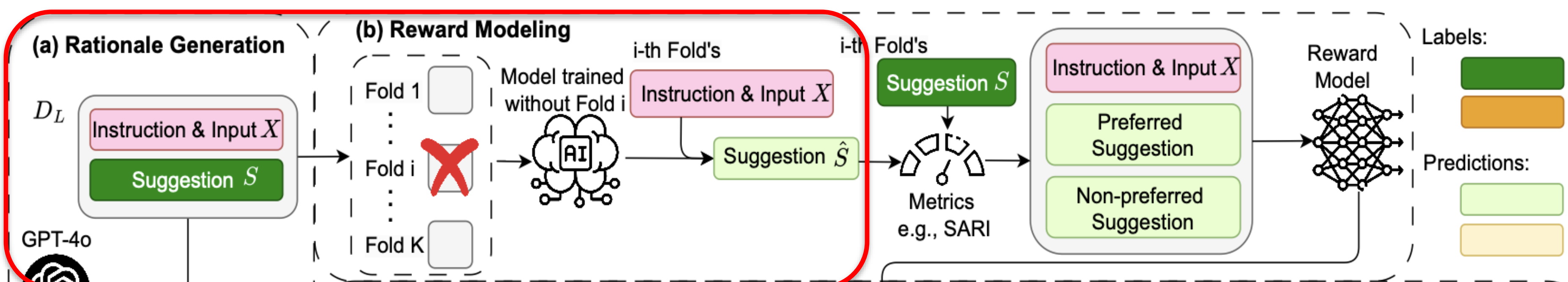
Edits:
`[Edit Content]`

Explanation:

Method

Reward Modeling

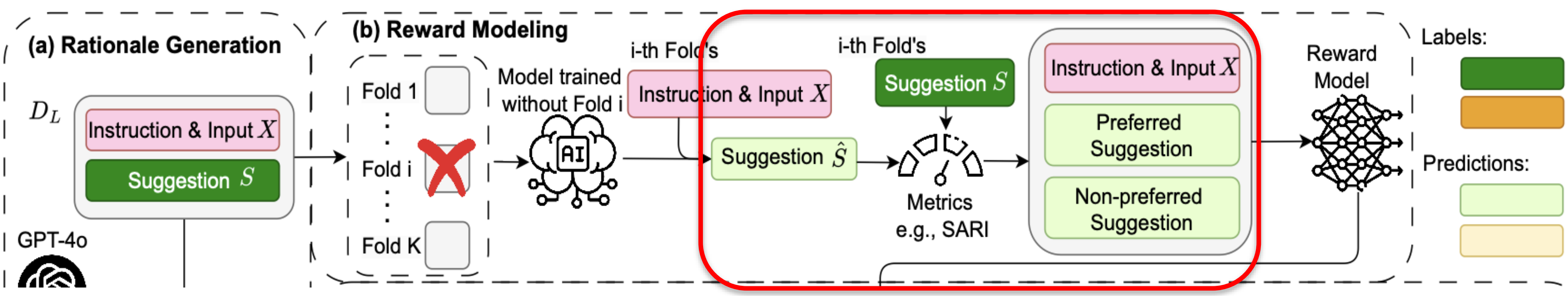
- Construct K models with labeled data and K-fold training, and make predictions on the held-out set.
- Utilizing these predictions and evaluation metrics, we can generate preference data.
- Build the reward model using the obtained preference data.



Method

Reward Modeling

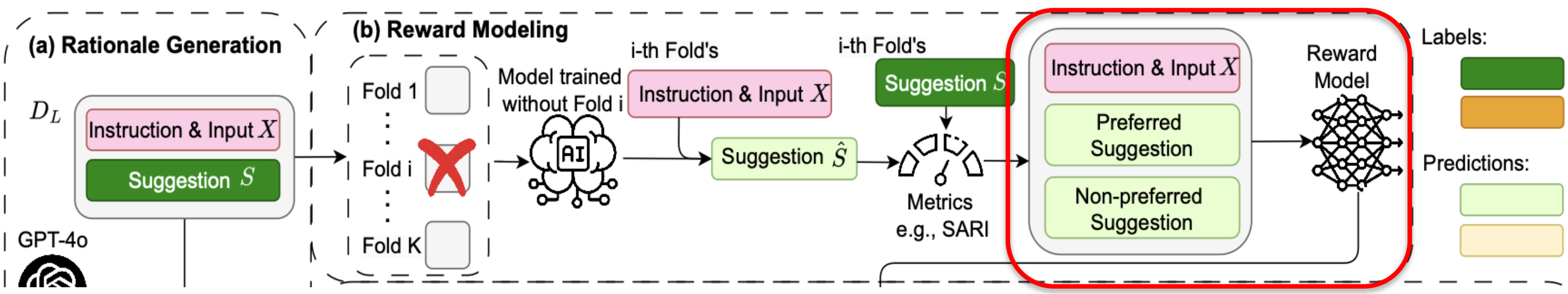
- ➢ Construct K models with labeled data and K-fold training, and make predictions on the held-out set.
- ➢ Utilizing these predictions and evaluation metrics, we can generate preference data.
- ➢ Build the reward model using the obtained preference data.



Method

Reward Modeling

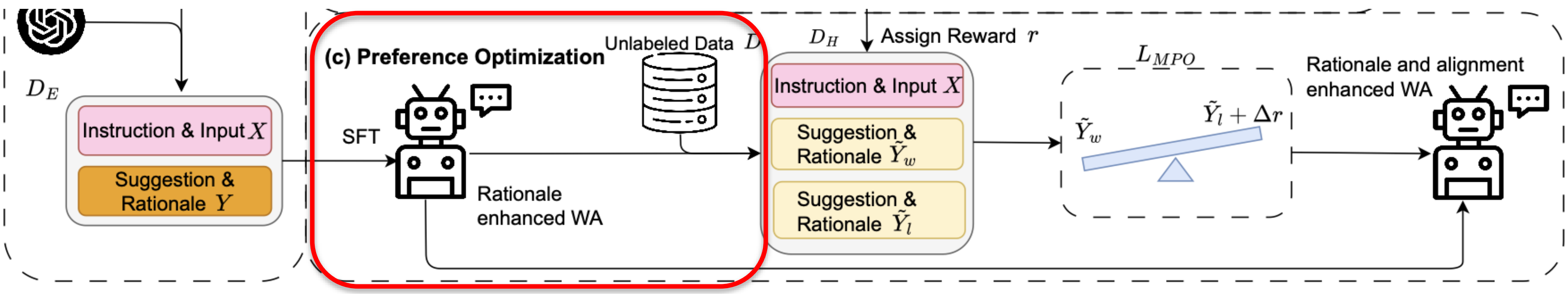
- ➢ Construct K models with labeled data and K-fold training, and make predictions on the held-out set.
- ➢ Utilizing these predictions and evaluation metrics, we can generate preference data.
- ➢ Build the reward model using the obtained preference data.



Method

Preference Optimization

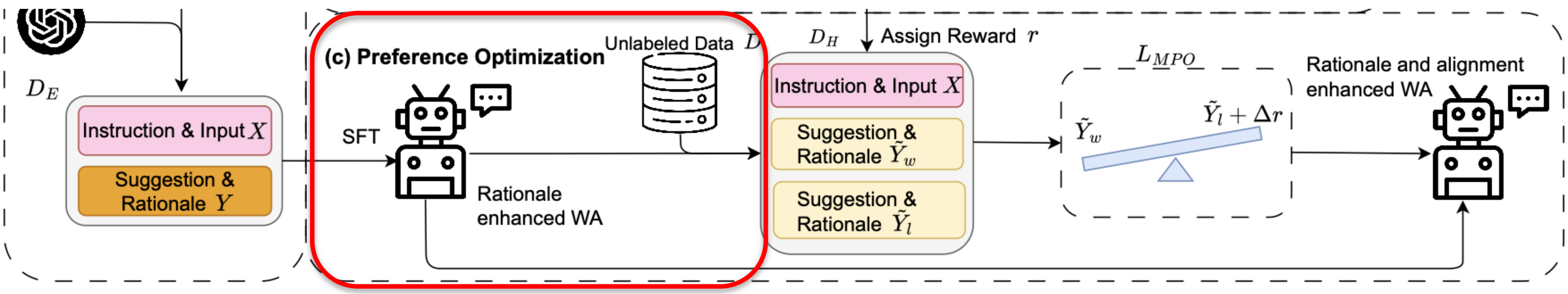
- Build a rationale enhance WA (SFT model) through SFT.
- Use SFT model to label unlabeled data.
- Use Reward model assign to rewards to SFT model's prediction.
- Generate high-quality preference pair data D_H based on the reward values.
- Optimize the WA with the L_{MPO} loss on D_H.



Method

Preference Optimization

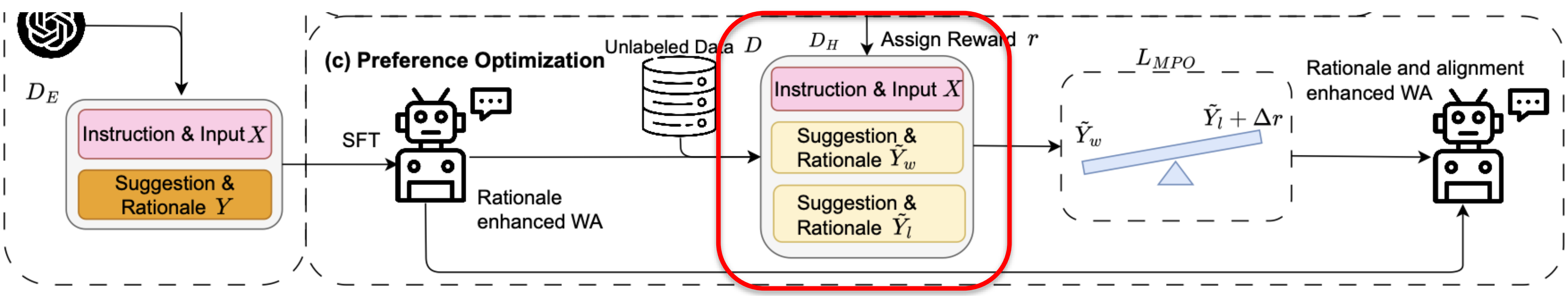
- Build a rationale enhance WA (SFT model) through SFT.
- Use SFT model to label unlabeled data.
- Use Reward model assign to rewards to SFT model's prediction.
- Generate high-quality preference pair data D_H based on the reward values.
- Optimize the WA with the L_{MPO} loss on D_H.



Method

Preference Optimization

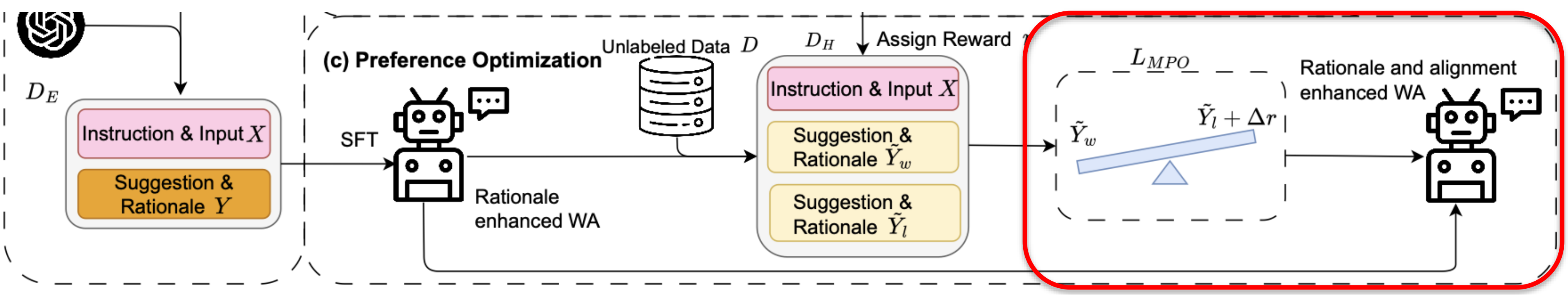
- Build a rationale enhance WA (SFT model) through SFT.
- Use SFT model to label unlabeled data.
- Use Reward model assign to rewards to SFT model's prediction.
- Generate high-quality preference pair data D_H based on the reward values.
- Optimize the WA with the L_{MPO} loss on D_H.



Method

Preference Optimization

- ➢ Build a rationale enhance WA (SFT model) through SFT.
- ➢ Use SFT model to label unlabeled data.
- ➢ Use Reward model assign to rewards to SFT model's prediction.
- ➢ Generate high-quality preference pair data D_H based on the reward values.
- ➢ Optimize the WA with the L_{MPO} loss on D_H.



Method

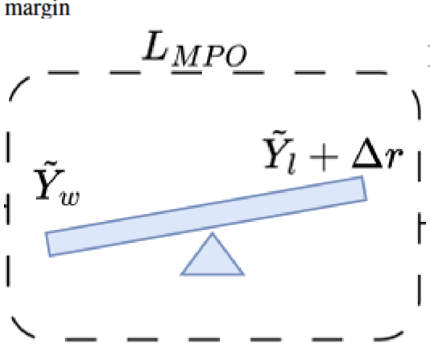
L_{MPO} loss function:

- A margin-based preference optimization loss.
  - Margin: the reward difference as determined by the reward model.

$$\mathcal{L}_{DPO} = -\mathbb{E}_{(x,\tilde{y}_w,\tilde{y}_l)\sim D_H}\left[\log\sigma\left(\beta\log\frac{\theta_{\text{W}}(\tilde{y}_w \mid x)}{\theta_{\text{SFT}}(\tilde{y}_w \mid x)} - \beta\log\frac{\theta_{\text{W}}(\tilde{y}_l \mid x)}{\theta_{\text{SFT}}(\tilde{y}_l \mid x)}\right)\right] \tag{1a}$$

$$= -\mathbb{E}_{(x,\tilde{y}_w,\tilde{y}_l)\sim D_H}\{\log\sigma[\beta(\log\theta_{\text{W}}(\tilde{y}_w \mid x) - \log\theta_{\text{W}}(\tilde{y}_l \mid x) - \underbrace{(\log\theta_{\text{SFT}}(\tilde{y}_w \mid x) - \log\theta_{\text{SFT}}(\tilde{y}_l \mid x))}_{\text{margin}})]\} \tag{1b}$$

$$\mathcal{L}_M = -\mathbb{E}_{(x,\tilde{y}_w,\tilde{y}_l)\sim D_H}\{\log\sigma[\beta(\log\theta_{\text{W}}(\tilde{y}_w \mid x) - \log\theta_{\text{W}}(\tilde{y}_l \mid x) - \gamma\underbrace{(r_w - r_l)}_{\text{margin}})]\} \quad \text{1c)}$$



$$L_{MPO} = \lambda L_M + L_{CE}(\tilde{y}_w)$$

Experiments

- Dataset:
  - ➢ 8 writing-related tasks, details shown in Table 1.
- WA model and reward model:
  - ➢ LLama3-8B, with LoRA fine-tuning.

Task	Train	Test	Metric	Abbrev.	# Sent
GEC	W&I+LOCNESS-Train	CoNLL-2014	M2	CoN	1,312
Fluency	ITERATER-V2-Train	ITERATER-fluency	SARI	ITR-F	88
Clarity	ITERATER-V2-Train	ITERATER-clarity	SARI	ITR-L	185
Coherence	ITERATER-V2-Train	ITERATER-coherence	SARI	ITR-C	35
Paraphrase	Parabank V2	STSB	SARI	STSB	97
Neutralization	WNC - Train	WNC	SARI	WNC	1,000
Simplification	TurkCorpus, NEWSELA, WikiLarge, Wiki-Auto, Parabank V2	ASSET	SARI	AST	359
FST	GYAFC-EM-Train	GYAFC-EM	BLEU, ACC	GYAFC-EM	1,416
FST	GYAFC-FR-Train	GYAFC-FR	BLEU, ACC	GYAFC-FR	1,332

Table 1: The tasks, training sets, test sets, metrics used, abbreviations used, and numbers of sentences (# Sent) in the various test sets in our evaluation benchmark. ACC represents the accuracy evaluation metric.

Experiments

- Overall Performance

	System	CoN	ITR-F	ITR-L	ITR-O	STS	WNC	AST	GYAFC		ALL
									EM	FR	
a)	Llama-3.3-70B-Instruct	55.6	46.5	31.4	31.0	34.6	31.8	46.4	59.5 / 98.1	56.2 / 98.4	43.7
a)	ChatGPT	53.3	50.9	31.5	31.0	39.9	36.3	47.0	57.7 / 99.6	60.4 / 99.5	45.3
a)	GPT-4	59.9	51.6	32.6	32.3	42.2	40.8	46.3	60.2 / 99.6	62.4 / 99.5	47.6
a)	GPT-4o	59.4	51.1	32.4	32.4	42.4	41.1	47.4	62.8 / 99.2	63.7 / 99.1	48.1
b)	Llama-3.3-70B-Instruct (R)	58.4	49.4	35.0	31.8	37.7	41.2	44.7	63.6 / 97.8	65.3 / 98.2	47.5
b)	ChatGPT (R)	56.1	51.1	30.3	28.7	40.6	36.6	45.0	63.1 / 98.7	63.5 / 98.9	46.1
b)	GPT-4 (R)	60.4	50.1	33.3	32.8	41.2	40.7	**47.6**	63.2 / 98.8	63.5 / 99.1	48.1
b)	GPT-4o (R)	60.8	51.4	32.6	32.2	43.3	40.9	46.1	64.4 / 98.4	64.6 / 98.6	48.5
c)	PEER-EDIT-11B	N.A.	52.1	32.5	32.7	28.2	54.5	29.5	N.A.	N.A.	N.A.
c)	Writing-Alpaca (7B)	55.9	**52.8**	**39.4**	37.1	44.6	64.4	44.7	N.A.	N.A.	N.A.
c)	CoEDIT-xxl (11B)	57.1*	51.6	31.8	31.5	42.9*	**71.0**	41.7	66.0 / 98.7*	68.7 / 97.9*	51.7
	Ours based on Flan-T5-xxl (11B) (RealEdit-11B)										
d)	SFT model	58.3	50.9	33.6	32.2	43.0	70.8	41.4	69.2 / 97.3	70.5 / 97.1	52.1
d)	+ Self-Training Alignment	61.4	49.3	32.8	34.7	47.0	68.9	41.1	75.3 / 96.8	78.0 / 96.3	54.3
e)	SFT model (R)	61.8	51.3	30.2	36.1	46.6	69.0	43.1	73.4 / 97.4	76.2 / 97.1	54.1
e)	+ Self-Training Alignment (R)	62.1	52.5	33.5	38.6	44.7	70.2	42.8	75.6 / 97.1	77.4 / 96.9	55.2
	Ours based on Llama 3.1 8B (RealEdit-8B)										
f)	SFT model	61.7	50.5	31.6	35.6	43.3	66.4	42.0	75.3 / 97.9	75.5 / 96.8	53.5
f)	+ Self-Training Alignment	62.5	48.7	31.9	**40.3**	**47.1**	64.6	45.2	**78.0** / 96.8	78.0 / 95.9	55.1
g)	SFT model (R)	62.7	51.1	34.0	37.5	45.1	65.8	41.3	75.9 / 98.4	76.4 / 97.5	54.4
g)	+ Self-Training Alignment (R)	**65.5**	48.7	35.4	37.6	46.5	65.9	45.2	77.3 / 97.9	**78.2** / 97.3	**55.6**

Our final model

Table 1: Performance on writing-related tasks. All results are shown in %. *: Results reproduced using the official checkpoint and scripts released by Raheja et al. (2023), due to different evaluation metrics or test sets not previously evaluated. For the GYAFC test sets, the first score is BLEU and the second is accuracy. Following Raheja et al. (2023); Zhang et al. (2023), we show the averaged result under the ALL column, and we only consider the BLEU score for the GYFAC test sets when taking the average. **a)**: zero-shot performance of LLMs. **b)**: zero-shot performance of LLMs when also prompted to generate rationales (or explanations) for their writing suggestions. **c)**: SOTA WAs. **d) & f)**: RealEdit trained without rationale. **e) & g)**: RealEdit trained with rationale.

Analysis

- Our reward model is effective in distinguishing high quality output from low quality output.

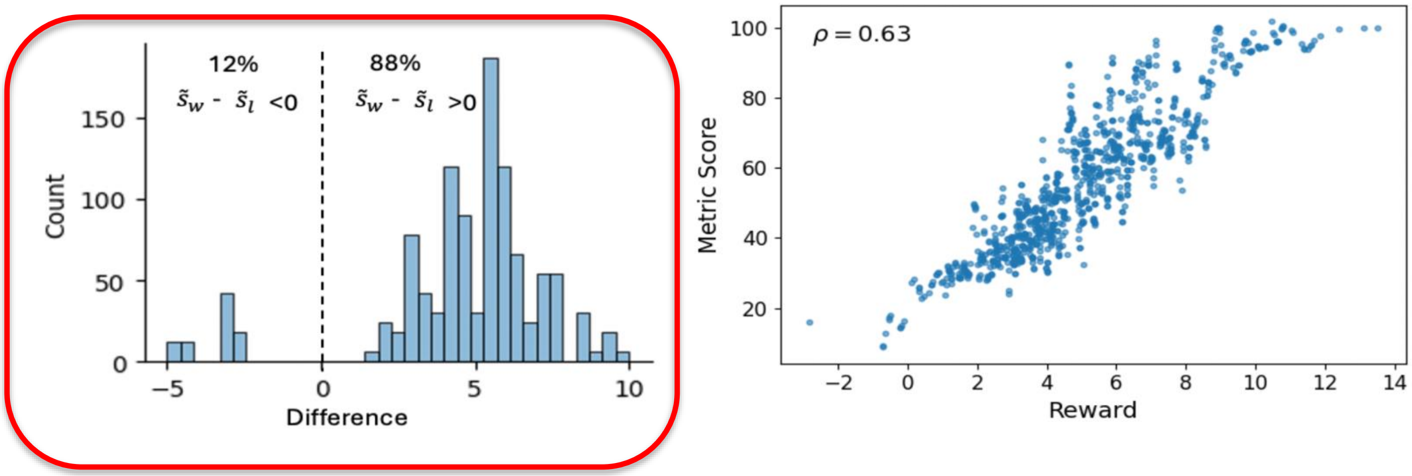


Figure 4: **Left**: The distribution of $metric(\tilde{s}_w) - metric(\tilde{s}_l)$, where $metric$ represents the task-specific evaluation metric. **Right**: Pearson correlation between our reward model and automatic evaluation metric.

Self-Training Alignment Analysis

- Our reward model exhibits strongly correlation with task-specific automatic evaluation metrics and human preferences.

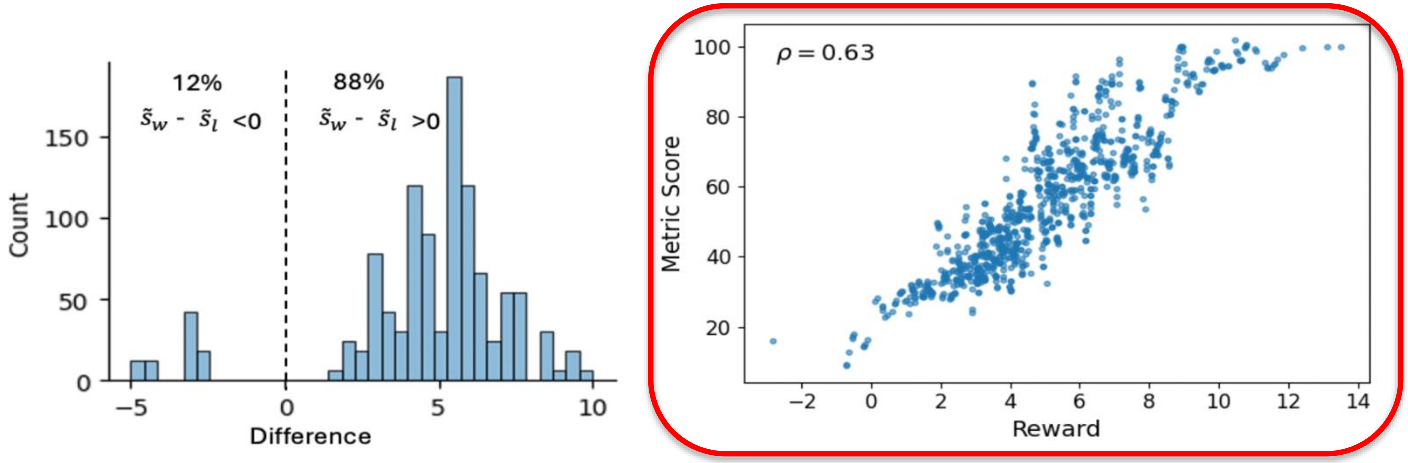


Figure 4: **Left**: The distribution of $metric(\tilde{s}_w) - metric(\tilde{s}_l)$, where $metric$ represents the task-specific evaluation metric. **Right**: Pearson correlation between our reward model and automatic evaluation metric.

Self-Training Alignment Analysis

- Using the winning response selected by the reward model could more effectively improve the WA performance.

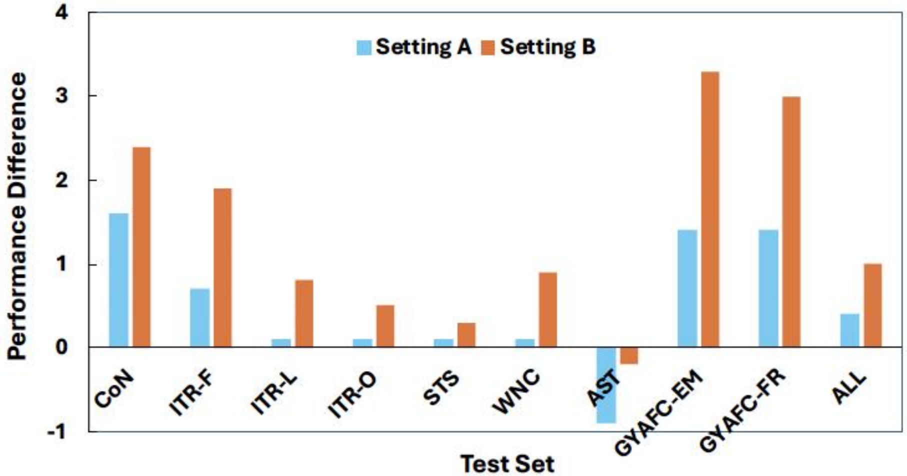


Figure 6: The performance difference (in %) between the WA (obtained under Setting A and B) and the SFT model. **Setting A**: Fine-tune the SFT model with an additional 78k labeled data (s). **Setting B**: Fine-tune the SFT model with an additional 78k $\tilde{s}_w$ from $\mathcal{D}_H$.

Self-Training Alignment Analysis

- Each of the components inside our loss function is effective.
- Our preference optimization loss function outperforms related methods.

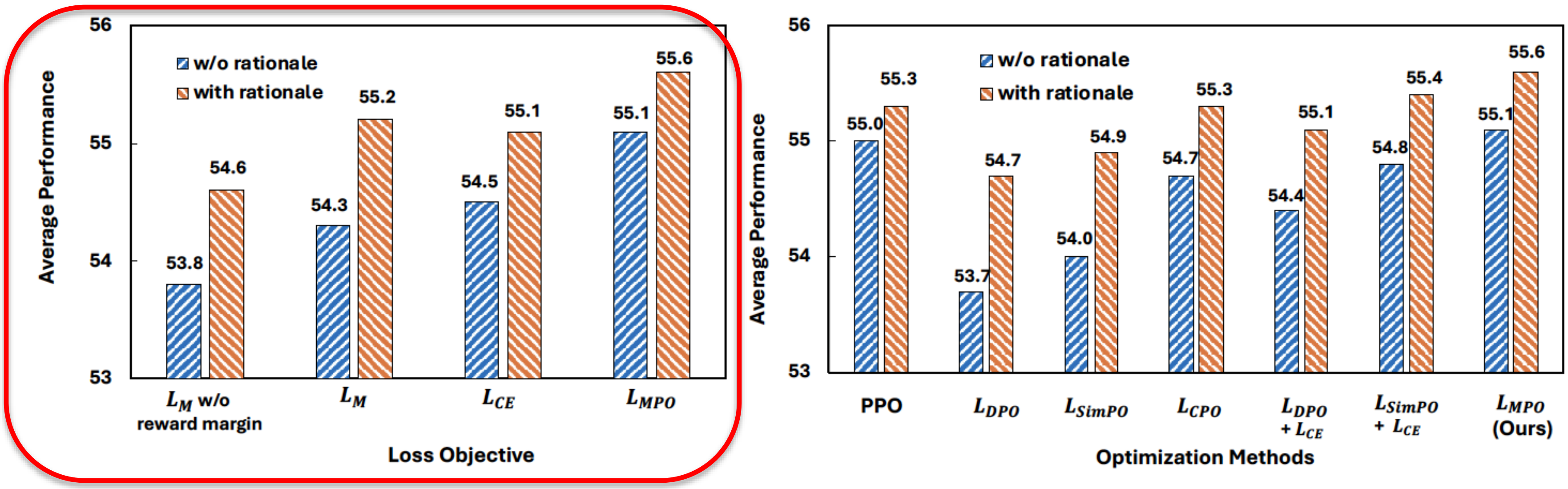


Figure 5: **Left**: Ablation study evaluating the significance of individual components in the loss function (Eq. (2)). The bars labeled 'L_M w/o reward margin' indicate setting γ to 0 in Eq. (1c). **Right**: Performance comparison against other related preference optimization methods.

Self-Training Alignment Analysis

- Each of the components inside our loss function is effective.
- Our preference optimization method outperforms related methods.

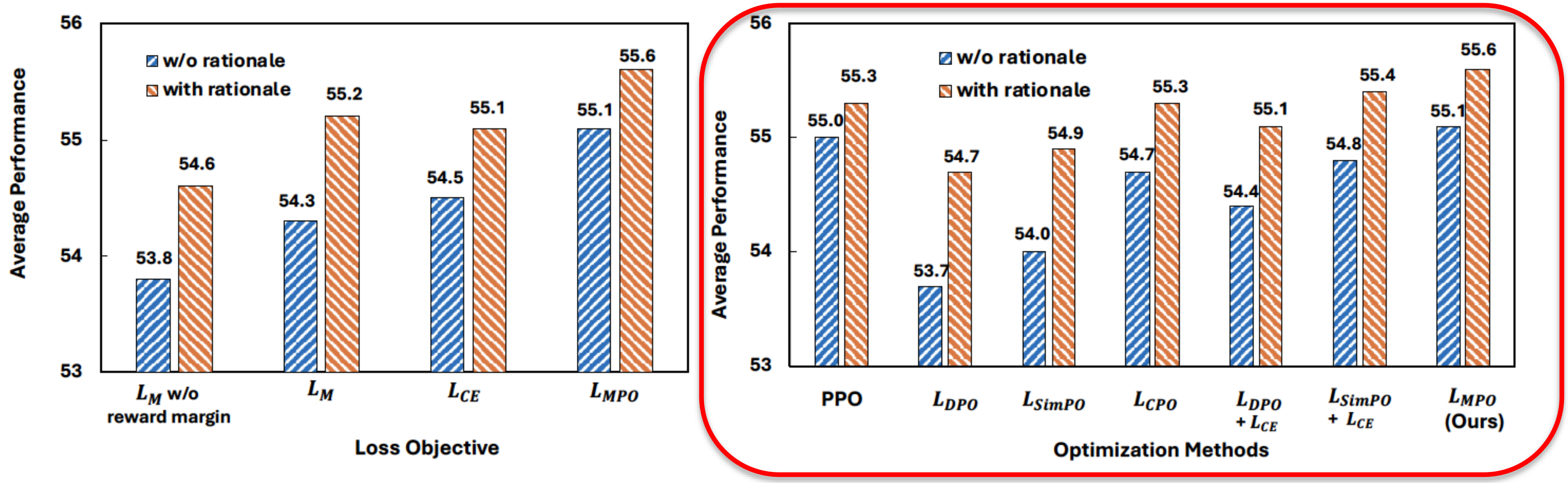


Figure 5: **Left**: Ablation study evaluating the significance of individual components in the loss function (Eq. (2)). The bars labeled 'L_M w/o reward margin' indicate setting γ to 0 in Eq. (1c). **Right**: Performance comparison against other related preference optimization methods.

Rationale Analysis

- When trained with rationales, WA become more confident and proficient in generating accurate writing suggestions.
- Training with rationales helps WA to propose more edits (less conservative) while maintaining relatively high precision.

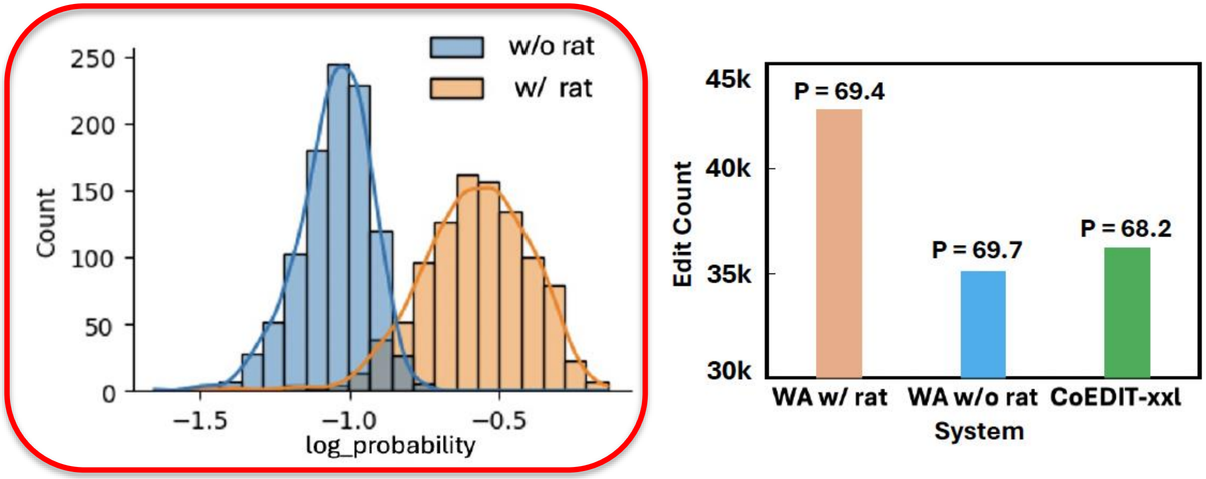
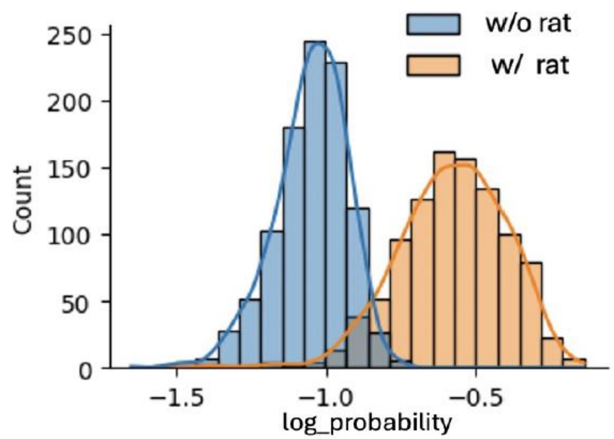
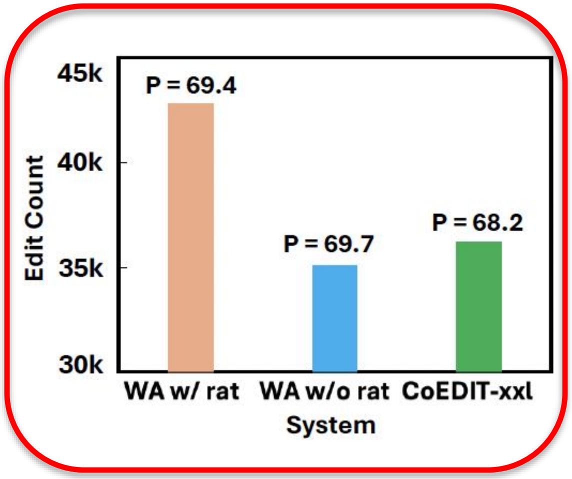


Figure 7: **Left**: The probability to generate the target sentence by WA trained with (w/ rat) and without rationale (w/o rat). **Right**: Number of edits proposed by different systems, with the precision of the edits displayed above each bar (P=*).

Rationale Analysis

- When trained with rationales, WA become more confident and proficient in generating accurate writing suggestions.





- Training with rationales helps WA to propose more edits (less conservative) while maintaining relatively high precision.

Figure 7: **Left**: The probability to generate the target sentence by WA trained with (w/ rat) and without rationale (w/o rat). **Right**: Number of edits proposed by different systems, with the precision of the edits displayed above each bar (P=*).

Conclusion

- Propose a novel Rationalize & Align framework to enhance WA.
- Our analysis discover that rationale helps WA to be more confident and less conservative.
- Our proposed margin-based preference optimization loss (MPO) surpass related preference optimization methods.
- We have developed the first open-source WA capable of generating rationales alongside its writing suggestions.

Thank You!