ate the revised responses. Each example (see Table 8 in Appendix B) in the dataset is a quadruplet (<query>, <initial response>, <critique outputs>, <revised response>).

3.2 Stylistic Rewrite

We follow RewriteLM (Shu et al., 2024) and apply chain-of-thought (CoT) prompting (Wei et al., 2022) to generate stylistic rewrite instructions for source texts from the C4 corpus (Raffel et al., 2020). The dataset is constructed by integrating source text and generated rewrite instructions using a structured template (see Table 12 in Appendix C). We then prompt LLMs to produce the revised text. Each example consists of a triplet (<source>, <instruction>, <revised text>), as exemplified by Table 10 in Appendix B.

3.3 Conversational Rewrite

We begin with large-scale natural prompts paired with raw generated emails from those prompts (see Table 7). This dataset serves as the foundation for improving clarity, tone, and personalization in conversation-based text generation.

Multi-turn instruction generation. To generate human-like rewrite instructions that focus on modifying specific details of the original conversation, we introduce a multi-turn refinement process (see Table 1) that iteratively enhances instruction specificity and naturalness. We first integrate natural prompts and raw emails, using few-shot demonstrations (see Tables 14, 15 in Appendix C) to generate raw instructions. These raw instructions, however, often lack and fail to guide nuanced rewrites (e.g., adding or removing details). For example, they might simply request specifying placeholders and benefits for customers in the raw email without providing concrete details.

We refine *raw* instructions into a more *specific* version via few-shot prompting (see Tables 16, 17 in Appendix C). For example, instantiate a concrete social media example ("TikTok") and specific benefits ("chance to win prizes in contests and giveaways"). Then, we further refine specific instructions into a *natural*, human-like style (see Tables 18, 19 in Appendix C). For example, use oral expressions ("say we're now on...", "let's tell them about the cool stuff...") to improve engagement and conversational fluency.

Rewrite generation. Finally, we generate revised emails by combining the natural prompt, raw email, and final rewrite instruction via structured template (see Table 13 in Appendix C) to prompt LLMs. Each example in CHATREWRITE consists of a quadruplet (<natural prompt>, <raw email>, <instruction>, <revised email>).

4 Framework of DR GENRÉ

4.1 Supervised Fine-Tuning (SFT)

SFT involves training a LLM on our synthetic (prompt, revisions) datasets. The prompt specify the task to be performed as well as the source text, and the responses are the generated revised outputs. This process distills the basic knowledge of reasonable text rewriting patterns from LLMs to a unified student model, and teaches the model to learn to follow diverse instructions and perform a variety of tasks. In our approach, we fine-tune a pretrained model on the combined datasets of factuality, style, and conversational rewrites. By providing explicit instructions and corresponding rewrites, the model learns to generalize across different rewriting tasks under supervised learning, acting as a reference policy π^{SFT} for the later RL stage.

4.2 Reward Modeling with LLM Preference

Preference data annotation. After SFT, the model has a basic knowledge of what high-quality revisions be like but not well-aligned with implicit preferences. For each input, we sample 10 SFT responses and compute their agreement and coherence scores by few-shot prompting LLMs. We design an agreement judging prompt for each of the three tasks (see Tables 20∼22 in Appendix D). For coherence, as it only depends on the revised response, we use consistent prompting (see Table 23 in Appendix D) across all tasks. We select a pair of revised responses with highest and lowest scores (denoted as y^+ and y^-) for each prompt x to formulate the RM training set.

RM training. We then train a generic agreement RM r_{φ_1}, and coherence RM r_{φ_2}[2], using a mixture of three preference datasets. We adopt the Bradley-Terry (BT) (Bradley and Terry, 1952) model where for each pair of revised responses $y^+, y^- \sim \pi^{\text{SFT}}(y|x)$ given a prompt x from training data $\mathcal{D}$, the preference probability is defined as

$$P(y^+ \succ y^-|x) = \frac{e^{r^*(y^+,x)}}{e^{r^*(y^+,x)} + e^{r^*(y^-,x)}}, \qquad (1)$$

[2] Conciseness reward is a rule-based edit distance metric.

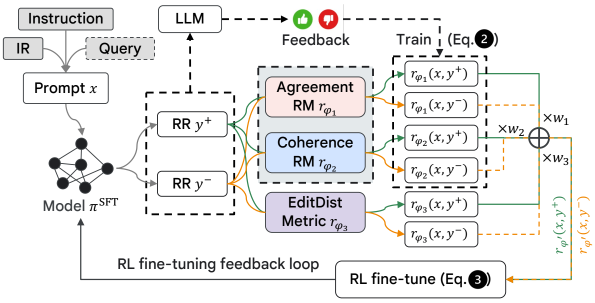

Figure 2: DR GENRÉ: RL fine-tuning with weighted decoupled rewards. Dashed lines represent workflows of reward modeling (for agreement and coherence). IR, RR denote initial and revised responses.

where r^* represents a RM (e.g., $r_{\varphi 1}$, $r_{\varphi 2}$). The BT loss function is then formulated as

$$\mathcal{L}_R = -\mathbb{E}_{(x,y^+,y^-)\sim\mathcal{D}}[\log\sigma(r^*(y^+,x)-r^*(y^-,x))], \tag{2}$$

where σ is the sigmoid function. Using this framework, we independently train the agreement and coherence reward models.

4.3 Reinforcement Learning with Decoupled Rewards

During the reward modeling phase, the SFT model takes prompts x and generates response pairs $y^+, y^- \sim \pi^{\text{SFT}(y|x)}$, which are then evaluated by each reward function to generate objective-oriented preference signals. In the RL phase, we integrate all reward functions to guide policy optimization. Specifically, we incorporate the decoupled rewards into the PPO[3] (Schulman et al., 2017) objective

$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta(y|x)}[r_{\varphi'}(x,y)] \\ - \beta \cdot \mathbb{D}_{\text{KL}}[\pi_\theta(y|x)||\pi_{\text{ref}}(y|x)], \tag{3}$$

where β is a parameter that controls the deviation from the reference policy π_{ref} (the initial SFT model π^{SFT}), and $r_{\varphi'}$ represents the aggregated decoupled reward function

$$r_{\varphi'}(x,y) = \sum_{o=1}^{O} w_o^t \cdot r_{\varphi_o}(x,y), \text{where } (x,y) \sim \mathcal{D}^t. \tag{4}$$

Here, $O = 3$ is the number of objectives, $\mathcal{D}^t$ denotes the dataset for task t (e.g., LONGFACT), and $w_o^t \in [0,1]$ is a task-specific weight for objective o-oriented reward r_{φ_o}. For example, conversational rewriting samples will be assigned a higher agreement weight w_1 than stylistic rewriting, as they involve more intricate detailed editing. An illustration of DR GENRÉ is shown in Figure 2.

5 Evaluation

5.1 Setup

We use instruction-tuned PaLM 2-L (denoted PaLM 2-L-IT) (Anil et al., 2023) as the teacher model for generating both training data (§3) and reward modeling data (§4.2). The base (student) model is a PaLM 2-S[4], which serves as the foundation for few-shot experiments and the initial checkpoint for SFT. Both agreement and coherence RMs are PaLM 2-M, fine-tuned to capture the teacher model's preferences and guide the student model during RL. For evaluation, we employ Gemini-1.5-Ultra (Team et al., 2023) as the AutoRater, which are used together with rule-based metrics to rate the quality of generated rewrites. We detail the other experimental setups in Appendix A.

Metrics. We evaluate performance across three objective-oriented metrics using AutoRaters:

- **Agreement**: Measures how well the revised text adheres to the instruction in terms of atomic requirements (e.g., correcting non-factual statements). We design task-specific agreement prompts to capture these requirements.
- **Coherence**: Judge whether the revised response is internally consistent. We use few-shot LLM prompting to assess coherence.
- **Edit Ratio** (Ristad and Yianilos, 1998): Quantifies the word-level textural difference between the original and revised texts. This is computed as the relative edit distance normalized by the length of the original text, reflecting conciseness—the proportion of text modified.

For LONGFACT, we follow Wei et al. (2024) to evaluate fact correction accuracy **F1@K**, where K is the expected number of facts per response. We consider two K values: the medium and maximum number of facts averaged in LONGFACT.

For REWRITELM, we follow Shu et al. (2024) to measure **NLI** (Bowman et al., 2015) and **Reverse NLI** score over the source-revision pairs. These scores estimate how well the rewrite retains the original information. Higher NLI and lower edit

[3] We use PPO instead of Direct Policy Optimization (DPO) (Rafailov et al., 2024) since our approach relies on explicit reward modeling rather than implicit preference scores. Additionally, PPO allows for exploration beyond the initial SFT policy, potentially discovering better rewriting strategies.

[4] We choose PaLM 2 models to be consistent with that of the prior rewrite work, i.e., RewriteLM (Shu et al., 2024).

Method	Length	F1@13↑	F1@35↑	Agreement↑	Coherence↑	Edit Ratio↓
Few-shot	287	0.7232	0.4579	0.7607	0.6367	0.0420
Surgical	369	0.8255	0.5480	**0.9238**	0.4120	**0.0108**
SFT-LongFact	348	0.7967	0.5192	0.8011	0.5141	0.0214
SFT	348	0.8209	0.5204	0.7950	0.5400	0.0210
DR GENRÉ-static	389	**0.8261**	**0.5531**	0.7926	**0.6520**	0.0583
DR GENRÉ	365	0.8091	0.5409	0.7878	0.6400	0.0270

Table 3: Performance on LONGFACT. Methods are grouped into ICL-based, SFT-based, and RL-based.

Method	Length	NLI↑	Reverse NLI↑	Agreement↑	Coherence↑	Edit Ratio↓
Few-shot	108	0.8790	0.8418	0.8235	**0.6960**	0.1168
SFT-RewriteLM	106	0.8806	0.8563	0.9524	0.6720	0.1242
SFT	102	0.8914	0.8718	0.9163	0.6840	**0.1078**
RL-CoComposer	181	**0.9256**	**0.8830**	0.9042	0.6480	0.2499
DR GENRÉ-static	107	0.9173	0.8591	0.9433	0.6800	0.1200
DR GENRÉ	101	0.8937	0.8684	**0.9641**	0.6834	0.1541

Table 4: Performance on OPENREWRITEEVAL (Shu et al., 2024). Length is the averaged output length.

ratio are desirable, as excessive edits can introduce hallucinations if the NLI scores are low.

For CHATREWRITE, we use auto Side-by-Side (**AutoSxS**) (Zheng et al., 2023) for pairwise agreement evaluation. AutoSxS compares responses generated by different models for the same prompt (see Tables 24, 25 in Appendix D). It is particularly useful for capturing nuanced differences between responses, as pointwise agreement checks often neglect implicit differences.

Baselines. We compare DR GENRÉ with three rewrite generation baselines[5] across all tasks:

- **Few-shot (ICL)**: Direct generate rewritten texts by few-shot prompting PaLM 2-S.
- **SFT**: PaLM 2-S fine-tuned on a mixture of all datasets, serving as a supervised baseline.
- **DR GENRÉ-static**: PaLM 2-S fine-tuned with RL using static weights, i.e., w_o^t becomes task-agnostic w_o, for the three reward objectives.

5.2 Results on LONGFACT

For the factuality rewriting task, we include two additional baselines:

- **Surgical**: Directly substituting non-factual contents identified in the critique outputs with the factual revisions, without further refinement.
- **SFT-LongFact**: Training the PaLM 2-S exclusively on LONGFACT. This baseline allows isolating the effect of task-specific training compared to the generic training used in SFT.

Table 3 presents the results on the sampled 250 query subset (Wei et al., 2024) of LONGFACT. DR GENRÉ-static achieves the highest factual correction performance (F1@13: 0.82, F1@35: 0.55), along with the highest coherence, while maintaining competitive edit ratio. DR GENRÉ, with dynamically adjusted reward weights, balances factuality and edit preservation, yielding a lower edit ratio and strong coherence while maintaining factual accuracy (F1@13: 0.81, F1@35: 0.54).

SFT outperforms SFT-LongFact, confirming that multi-task learning (Vilalta and Drissi, 2002) benefits factuality rewriting. However, all SFT-based methods experience a significant drop in coherence (around 10%) compared to few-shot prompting, highlighting *the challenge of maintaining internal consistency while improving instruction adherence in factuality rewriting*. Such limitation is further exemplified by the surgical baseline, which, despite achieving the highest agreement score, exhibits poor coherence due to its lack of refinement.

DR GENRÉ mitigates this trade-off by dynamically adjusting reward weights based on task properties. For example, by assigning higher edit distance weights to tasks requiring more substantial revisions (e.g., factuality vs. stylistic), it ensures robustness and no severe deviation (e.g., lower edit ratio compared to DR GENRÉ-static) throughout the post-training process.

5.3 Results on OPENREWRITEEVAL

For the stylistic rewriting task, we investigate two additional baselines:

- **SFT-RewriteLM**: Trains the PaLM 2-S exclusively on REWRITELM, without leveraging the patterns from other tasks.
- **RL-CoComposer**: An existing RL-based method (Shu et al., 2024) that optimizes a single

[5] We discuss more selection rationale in Appendix A.