Rewrite-to-Rank: Optimizing Ad Visibility via Retrieval-Aware Text Rewriting

Chloe Ho **Ishneet Sukhvinder Singh**[1] **Diya Sharma**[*] **Tanvi Reddy Anumandla**[*] **Michael Lu**[2] **Vasu Sharma**[3] **Kevin Zhu**[4]

Abstract

Search algorithms and user query relevance have given LLMs the ability to return relevant information, but the effect of content phrasing on ad visibility remains underexplored. We investigate how LLM-based rewriting of advertisements can improve their ranking in retrieval systems and inclusion in generated LLM responses, without modifying the retrieval model itself. We introduce a supervised fine-tuning framework with a custom loss balancing semantic relevance and content fidelity. To evaluate effectiveness, we propose two metrics: ΔMRR@K (ranking improvement) and ΔDIR@K (inclusion frequency improvement). Our approach presents a scalable method to optimize ad phrasing, enhancing visibility in retrieval-based LLM workflows. Experiments across both instruction-based and few-shot prompting demonstrate that PPO trained models outperform both prompt engineering and supervised fine-tuning in most cases, achieving up to a 2.79 ΔDIR@5 and 0.0073 ΔMRR@5 in instruction-based prompting. These results highlight the importance of how the ad is written before retrieval and prompt format and reinforcement learning in effective ad rewriting for LLM integrated retrieval systems.

1. Introduction

Large language models (LLMs) have demonstrated impressive capabilities across a wide range of natural language tasks such as question answering and summarization (Nijkamp et al., 2023; Lewis et al., 2021b). To overcome the limitations of internal knowledge, many modern LLM systems are designed to use Retrieval-Augmented Generation (RAG) systems (Lewis et al., 2021b), where a retriever selects relevant documents from an external corpus which is then passed to a generator. As LLMs continue to be used in industries like digital advertising, whose global revenue reached USD 259 billion in 2024, up 15% from 2023 (Interactive Advertising Bureau (IAB) & PwC, 2025), their ability to retrieve external documents is often limited by the quality of current retrieval systems (Alinejad et al., 2024). Current RAG pipelines focus on query-side optimization, often relying on black-box retrievers that compute similarity scores between user queries and a static document corpus to identify the most relevant documents (Liu et al., 2024), yet they ignore the producer side, i.e. how documents themselves can be optimized for retrieval. This oversight leads to relevant advertisements being missed due to minor differences in phrasing, formatting, or keyword choice (Dash et al., 2024). Suboptimal ranking in the retrieval process can significantly reduce ad visibility, which hurts revenue potential. Optimizing to fill the producer-side gap via integrating advertisements directly into LLM retrieval workflows presents a promising solution for relevant sponsored content and sustainable revenue models for AI applications.

In this paper, we introduce a novel approach to improving the ranking of advertisements by rewriting ad content rather than modifying the retrieval system itself. Specifically, we propose using a fine-tuned language model with a custom-designed loss function to encourage rewritten advertisements to: (1) align rewrites with user queries and (2) preserve the ad's original meaning. The contributions of this paper are as follows:

- We propose a new approach that improves document ranking and likelihood of inclusion in LLM output by rewriting ad content, rather than modifying the retrieval system, introducing a complementary angle to existing re-ranking and retriever-based methods.
- We develop a tailored loss function that balances maximizing relevance to user queries with preservation of the original intent of the ad, enabling effective fine-tuning for retrieval-sensitive content rewriting.
- We show how Proximal Policy Optimization (PPO) can be applied to train LLMs to optimize ad visibility, demonstrating a new use case for RL in language model fine-tuning.
- We contribute an evaluation setup using ΔMRR@K

[*] Equal contribution [1] University of Oxford, Oxford, England [2] University of Berkeley, Berkeley, CA, USA [3] Carnegie Mellon University, Pittsburgh, PA, USA [4] Algoverse AI Reserach, Remote. Correspondence to: Chloe Ho <chloe.ho.0830@gmail.com>.

and ΔDIR@K and highlight a scalable, potentially monetizable application of LLMs in ad retrieval, offering a framework others can adopt for commercial or academic use.

2. Related Works

Our work builds upon recent progress in Retrieval-Augmented Generation (RAG) systems and the broader literature on optimizing language model outputs for downstream utility.

Document Expansion vs. Ad Phrasing. Traditional document expansion techniques, such as term importance prediction (MacAvaney et al., 2020) and BERT-based re-ranking (Nogueira & Cho, 2020), often assume neutral objectives that can be generalized to a broad document corpus via optimizing for query relevance. Ad phrasing, conversely, holds even subtle linguistic changes to a higher standard insofar as influencing user engagement. Rather than uniformly enriching content, we target strategic micro-edits that align with the retrieval objectives of black-box models.

Retrieval-Augmented Generation. RAG has emerged as a powerful framework for enhancing language models with external knowledge sources (Lewis et al., 2021b), where a retriever selects documents passed to a generator. While prior work has largely focused on improving retrievers (Liu et al., 2024) or combining retriever-generator feedback loops (Alinejad et al., 2024), our approach complements these efforts by rewriting the documents themselves to enhance retrievability—without modifying the retriever.

Ad Content Optimization. In the context of digital advertising, existing work such as (Dash et al., 2024; Haans et al., 2013) has highlighted the limitations of sponsored content ranking, showing that minor changes in phrasing can drastically affect ad visibility. However, most methods address this through ranking adjustments or changes on the platform side. In contrast, we propose content-side optimization via LLM-driven rewriting, allowing advertisers to improve discoverability in black-box systems.

Prompting and Language Model Steering. Our use of Chain-of-Thought prompting (Wei et al., 2023b) and few-shot learning techniques (Brown et al., 2020) follows recent trends in LLM control. While prior work focuses on task accuracy or reasoning depth, we apply these techniques to guide ad rewriting for improved ranking and inclusion in LLM responses.

Reinforcement Learning with LLMs. We introduce a novel application of Proximal Policy Optimization (PPO) to optimize ad visibility based on a composite reward that balances ranking relevance and content fidelity. Although PPO has been successfully applied to instruction tuning and retrieval optimization (Ouyang et al., 2022; Kulkarni et al., 2024), its application to retrieval-centric rewriting remains underexplored.

Our framework fills a gap between search ranking systems and LLM generation, providing a scalable method for advertisers to increase content discoverability in AI-driven platforms.

3. Methods

Our objective is to rewrite advertisement documents such that they rank higher in retrieval systems and are more likely to be included in the LLM's response, while preserving the semantic content of the original ad.

In order to train the model for document rewriting, we employ a custom loss function, which 1) maximizes the similarity between the rewritten document and the query, 2) ensures that the rewritten document is more similar to the query than randomly selected top k documents, and 3) the rewritten document remains semantically faithful to the original document.

To interpret the overall performance of the model for each training iteration, we implemented two evaluation metrics MRR@K Improvement (ΔMRR@K) and document inclusion rate @ K improvement (ΔDIR@K), which measure the change in rank a document experiences after a rewriting strategy is applied and the change in frequency in which the document is included in the LLM output, respectively.

To access inclusion, we prompt for a black-box LLM, Gemini 1.5 Pro, with the original and rewritten ad documents. The prompt is provided in the Appendix and visible in the code base and is written to be intentionally minimal to avoid prompt-induced bias and reflect practical scenarios.

3.1. Data Construction

First, we randomly sampled 11,000 advertisements from the original dataset, CommercialAdsDataset by Microsoft, with 10,000 ads for the train data and 1,000 ads as our test data to construct our initial advertisement dataset.

Each ad is associated with a domain and subdomain label using a large language model (Gemini 1.5 Pro), which were verified through human annotators with 85% accuracy. We create a query data set by generating ten queries for each pair of domain-subdomains. For each query, we retrieve the top k relevant ads from the main advertisement dataset and generate LLM responses conditioned on those ads. The retrieval rankings produced for each query are then retained for later evaluation in rewriting strategies.

3.2. Retriever Configuration

Embedding model. We embed every advertisement *and* every query with the public Sentence-Transformers checkpoint `all-MiniLM-L6-v2`[1] Vectors are ℓ_2-normalised so cosine similarity reduces to inner-product distance.

Index type. All 11 000 ad embeddings (10 k train + 1 k test) are stored in a `faiss::IndexFlatIP` (exact inner-product search) built with FAISS. Exact search avoids ANN hyper-parameters and guarantees deterministic recall. All 11 000 ads are embedded in mini-batches of 64 and then added to the index in one call (`index.add(embs)`). At search time we encode one query at a time, so the FAISS call sees a batch of 1.

LLM for inclusion. Document-inclusion is evaluated by `gemini-1.5-pro-001` (Gemini 1.5 Pro, February 2024 release); the exact prompt appears in Appendix A.

3.3. Loss Function Construction and Implementation

The loss function guides the model during training by quantifying how well a rewritten ad balances three goals in a 1:1:1 ratio:

- Maximize similarity to the user query
- Maximize difference from top-k documents
- Preserve semantic meaning from the original ad

Let the loss function be composed of the three components:

- $L_{\text{rel\_gain}}$: Relevance gain loss
- L_{triplet}: Triplet sampling loss
- L_{fidelity}: Content fidelity loss

By minimizing loss, the model learns to generate rewrites that aim to improve retrievability while staying loyal to the original content. The overall loss function is a summation:

$L_{\text{total}} = L_{\text{rel\_gain}} + L_{\text{triplet}} + L_{\text{fidelity}}$

Let:

d = the document we are tracking

Q_d = set of user queries relevant to d

3.3.1. Relevance Gain Loss ($L_{\text{REL\_GAIN}}$)

This term encourages the rewritten ad to be more relevant to the query than the original version. Let:

d_{before}: original advertisement

d_{after}: rewritten advertisement

$q \in Q_{\text{d}}$: a query relevant to the ad d

sim(a,b): cosine similarity between the embeddings of a and b

Then:

$L_{\text{rel\_gain}}(q, d) = -(\text{sim}(q, d_{\text{after}})$ - $\text{sim}(q, d_{\text{before}}))$

3.3.2. Triplet Sampling Loss (L_{TRIPLET})

This loss term encourages the rewritten ad to be more relevant to the query than competing documents retrieved for the same query. We sample a set of 3 documents from the top-k retrieved list for a given query and compute their average similarity to the query. This average is then compared with the similarity between the rewritten ad and the query. This loss penalizes rewritten documents whose relevance to the query falls below the average of a sample of the top-k retrieved documents.

$N_{q,d}$: set of 3 documents for query q

Define the average similarity of the distractors to the query as:

$s_{\neg}(q) = \frac{1}{3}\sum_{n \in N_{q,d}} sim(n, q)$

Then the triple sampling loss is:

$L_{\text{triplet}}(q, d) = -\left[sim(q, d_{\text{after}}) - sim(q, s_{\neg}(q))\right]$

3.3.3. Content Fidelity (L_{FIDELITY})

We preserve content fidelity between the rewritten and original documents, maximizing the similarity between the original and rewritten documents.

$L_{\text{fidelity}}(q, d) = 1 - sim(d_{\text{after}}, d_{\text{before}})$

Our overall loss function for each document is as follows:

$L_{\text{total}} = L_{\text{rel\_gain}} + L_{\text{triplet}} + L_{\text{fidelity}}$

3.4. Prompting Strategies

To guide the language model in rewriting advertisements effectively, we designed two prompts based on a technique called Chain-of-Thought (CoT) reasoning. This approach encourages the model to ”think out loud” by walking through the reasoning process before generating a final rewrite.

Using two prompting strategies grounded in popular, established prompting methods (Wei et al., 2023b; Brown et al., 2020), our goal was to help the model focus on improving how relevant an ad sounds to a user's query, without losing the original meaning. The specific prompts are provided in the Appendix.

[1] 384-dimensional output; sentence-transformers v2.6.1.

3.4.1. Strategy A: Instruction-based Prompting with CoT (Zero-Shot):

In this approach, the model is given a high-level instruction to rewrite an ad in a way that improves its retrieval ranking and likelihood of inclusion and is told to reason and think before responding. The prompt encourages semantic alignment with user intent and explicit reasoning before rewriting.

3.4.2. Strategy B: Few-Shot Prompting with CoT:

This setup provides the LLM with two examples of ad rewrites, including the original ad, rationale, and improved version. After observing the examples, the model is prompted to perform a similar rewrite on a new ad input.

4. Experiments

4.1. Evaluation Metrics

Let:

K = rank cut-off we impose (e.g. 10)

$v \in \{\text{before}, \text{after}\} \rightarrow$ document version

rank$_v(d, q)$ = rank of d for query q

$X_v^{(d,q)} = 1$ if the LLM actually uses d in its final answer, else 0

4.1.1. Metric 1: MMR@K Improvement (ΔMRR@K)

We evaluate whether a rewriting strategy is effective in improving a document's ranking by measuring the change in mean reciprocal ranking prior and post imposition. We compute:

$$\Delta\text{MRR@}K = \text{MRR@}K_{\text{after}} - \text{MRR@}K_{\text{before}}$$

$$\Delta\text{MRR@}K = \frac{1}{|Q_d|} \sum_{q \in Q_d} \left(RR_K^{\text{after}}(d,q) - RR_K^{\text{before}}(d,q) \right)$$

where

$$RR_K^v(d,q) = \begin{cases} \frac{1}{\text{rank}_v(d,q)}, & \text{if rank}_v(d,q) \leq K \\ 0, & \text{otherwise} \end{cases}$$

4.1.2. Metric 2: Document-Inclusion Rate @ K Improvement (ΔDIR@K)

We track the inclusion rate of a document in the LLM's response across relevant queries and compute the change between the inclusion rate of a document remaining in the top-K set before and after rewriting. This metric represents the absolute change (percentage-point lift) and formally, we define it as follows:

$$\Delta DIR@K = DIR@K_{\text{after}} - DIR@K_{\text{before}}$$

Below is a formal definition of the base metric, $DIR@K_v$:

First, restrict the query set to include only queries where the document rank is within the top-k cut-off in both versions:

$$Q_d^{(K)} = \{q \in Q_d \mid \text{rank}_{\text{before}}(d,q) \leq K \wedge \text{rank}_{\text{after}}(d,q) \leq K\}$$

Then the Document-Inclusion Rate @ K for version v of any document is:

$$DIR@K_{v \in \{\text{before},\text{after}\}} = \frac{1}{|Q_d^{(K)}|} \sum_{q \in Q_d^{(K)}} X_v^{(d,q)}$$

4.2. Baselines

4.2.1. Zero-Rewrite Control

We establish a baseline using the original, unmodified ads. For each query, we evaluate the responses retrieved and generated using the original ad set, without any LLM rewriting.

4.2.2. Prompt Engineering

Each of the two prompting strategies in Section 3.3 are used to generate rewrites for the 1,000 test ads. For each rewritten version, we compute ΔMRR@K and ΔDIR@K relative to the original retrieval and inclusion statistics for each ad.

4.2.3. Supervised Fine-Tuning (SFT)

We fine-tune a pretrained language model (LLaMA-3.1-8B) using the original ads as inputs with rewritten advertisements as the ground truths. The rewritten versions are generated through a general prompt, which is provided in the Appendix, that is known to boost metric scores.

The objective is to teach the model to rewrite an original ad similarly to an improved version which is more likely to rank higher in retrieval and be included in LLM output.

4.3. Proximal Policy Optimization (PPO)

To further enhance the retrievability of rewritten ads, we apply Proximal Policy Optimization (PPO) on top of our supervised fine-tuned LLaMA-3.1-8B model. PPO enables the model to optimize directly for our task-specific objective by learning from a custom reward signal.

The reward is defined as the negative of the total loss introduced in Section 3.2. This loss function balances three priorities: improving query relevance, outperforming distractor documents, and preserving content fidelity.

For each ad, we perform the following steps to compute the reward:

- Identify up to three queries that share the same domain and subdomain as the ad.
- For each query, retrieve the top-k documents from the original dataset.
- Compute the reward using the loss function defined in Section 3.2.
- Average the loss across all selected queries to obtain the final reward signal.

This reinforcement learning stage allows the model to go beyond pattern imitation and optimize rewrites that are more likely to rank higher in retrieval and be included in downstream LLM outputs.

We implement LoRA fine-tuning with rank 8 on `all` attention and MLP projections, enable gradient checkpointing, train in `bfloat16`, and employ DeepSpeed ZeRO-2 with CPU optimizer offloading (`zero_stage=2`). All PPO hyper-parameters can be found in the Appendix.

5. Results

5.1. Comparing the Models

Tables 1, 2, and 3 present the ΔMRR@5 and ΔDIR@5 for the original ad baseline, prompt engineering, supervised fine-tuning, and PPO training across both prompting strategies, instruction-based and few-shot, for the latter three models.

Model	ΔMRR@5	ΔDIR@5
Zero-Rewrite	0.00	-0.0807

Table 1. Values of ΔMRR and ΔDIR for the zero-rewrite baseline for k = 5

Model	ΔMRR@5	ΔDIR@5
Prompt Eng.	0.0051	0.9061
SFT	0.0022	1.6382
PPO	0.0073	2.7944

Table 2. Comparison of ΔMRR and ΔDIR values across different models for k = 5 for instruction-based prompting

Model	ΔMRR@5	ΔDIR@5
Prompt Eng.	0.0067	1.9133
SFT	-0.0017	1.4199
PPO	-0.0008	1.9267

Table 3. Comparison of ΔMRR and ΔDIR values across different models for k = 5 for few-shot prompting

In our evaluation, ΔMRR@5 is 0 and ΔDIR@5 is -0.0807 for the zero-rewrite baseline. As expected, both are closer to zero than the metrics when the ads are rewritten.

Under instruction-based prompting, PPO consistently outperforms both prompt engineering and SFT in both ΔMRR and ΔDIR, achieving the highest ΔMRR value of 0.0073 and highest ΔDIR value of 2.7944, both of which are statistically significant under a 95% confidence interval. SFT achieves the second best ΔDIR score of 1.6382 but does have a low value for ΔMRR of 0.0022.

For few-shot prompting, PPO maintains a strong ΔDIR of 1.9267 which is relatively similar to prompt engineering's ΔDIR value of 1.9133, but does exhibit a negative ΔMRR value. SFT performs worst here, with both lower ΔMRR and ΔDIR values.

5.2. Ablation Studies

5.2.1. Varying K

We conduct ablation studies by varying the value of k to examine how the amount of ads retrieved affects the metrics ΔMRR@k and ΔDIR@k across models.

k	ΔMRR@k	ΔDIR@k
1	0.00	0.6293
3	0.00	-0.558
5	0.00	-0.0807
10	0.00	-0.2844
20	0.00	0.2288
30	0.00	0.1246

Table 4. ΔMRR and ΔDIR values across varying k values for the zero-rewrite baseline

The ΔMRR for the zero-rewrite control is always 0.00 since the ad content remains unchanged and therefore has identical similarity to the query. The ΔDIR, while expected to be near zero, may have slight fluctuations as the nondeterministic nature of LLM generated responses can include different ads across runs even when the input remains the same.

k	**Prompt Eng.**	**SFT**	**PPO**
1	0.0033	0.0035	0.0072
3	0.0051	0.0026	0.0072
5	0.0051	0.0022	0.0072
10	0.0055	0.0036	0.0085
20	0.0065	0.0038	0.0082
30	0.0064	0.0036	0.0082

Table 5. ΔMRR values for different k across models with instruction-based prompting

k	**Prompt Eng.**	**SFT**	**PPO**
1	0.4619	0.7773	1.8217
3	0.8898	1.3500	2.2337
5	0.9061	1.6382	2.7944
10	0.4598	1.0754	2.2342
20	-0.8517	-0.7896	-0.5477
30	-0.4629	-0.7658	-0.5215

Table 7. ΔDIR values for different k across models with instruction-based prompting

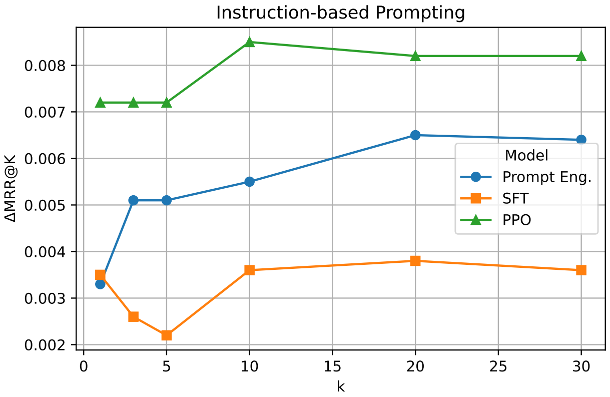

Figure 1. ΔMRR metric for instruction-based prompting across k values.

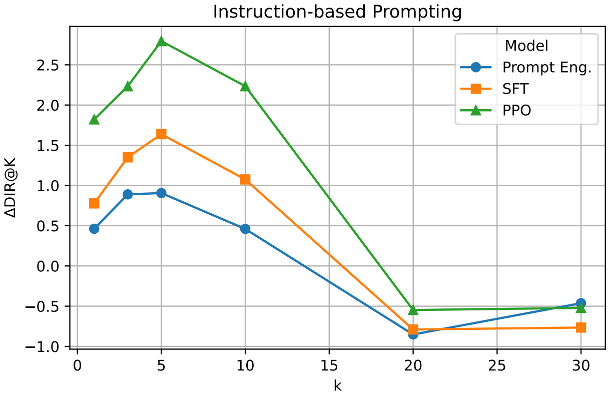

Figure 3. ΔDIR metric for instruction-based prompting across k values.

k	**Prompt Eng.**	**SFT**	**PPO**
1	0.0062	-0.0003	-0.0000
3	0.0068	-0.0017	-0.0006
5	0.0067	-0.0017	-0.0008
10	0.0081	-0.0015	-0.0005
20	0.0088	-0.0014	-0.0008
30	0.0090	-0.0017	-0.0010

Table 6. ΔMRR values for different k across models with few-shot prompting

k	**Prompt Eng.**	**SFT**	**PPO**
1	0.0139	0.8113	1.5303
3	0.6524	0.8892	2.0755
5	1.9133	1.4199	1.9267
10	0.5740	0.9773	1.4795
20	-0.9622	-0.9881	-0.9081
30	-0.4483	-0.4321	-0.6708

Table 8. ΔDIR values for different k across models with few-shot prompting

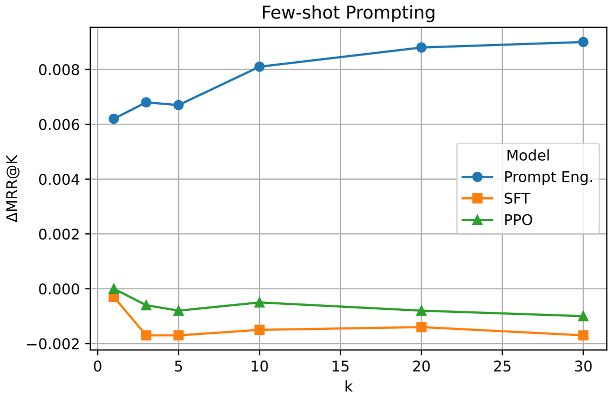

Figure 2. ΔMRR metric for few-shot prompting across k values.

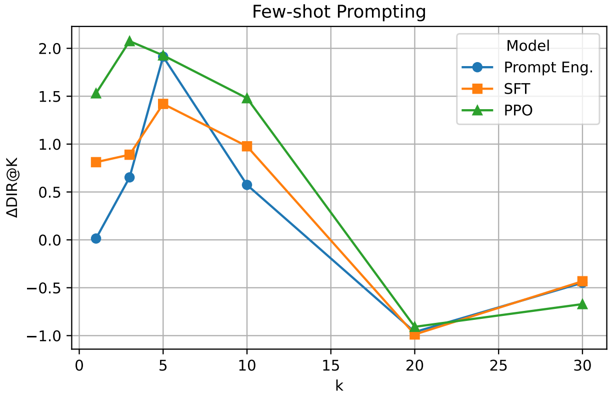

Figure 4. ΔDIR metric for few-shot prompting across k values.

ΔMRR Trends (Tables 5, 6):

- For instruction-based prompting, PPO maintains consistent performance across all values of k, with ΔMRR peaking at k = 10 (0.0085), surpassing the values of prompt engineering and SFT.
- For few-shot prompting, only prompt engineering has a positive value of ΔMRR. Both SFT and PPO have negative or near zero ΔMRR values suggesting that few-shot rewriting may decrease the ad's relevance to the query.

ΔDIR Trends (Tables 7, 8):

- Under instruction-based prompting, PPO has the highest values of ΔDIR, peaking at k = 5 (2.7944), surpassing both prompt engineering and SFT. Beyond k = 20, all three models exhibited negative values of DIR.
- For few-shot prompting, ΔDIR is the highest for PPO at k = 3 (2.0755), but becomes negative beyond k = 20.

5.2.2. Loss Function

To understand the impact of each component in our loss function, we conduct an ablation study across six different weightings of the composite loss used during PPO for instruction-based and few-shot prompting:

$L_{\text{total}} = \alpha L_{\text{rel\_gain}} + \beta L_{\text{triplet}} + \gamma L_{\text{fidelity}}$

The weightings tested are:

- **Equal weighting** (1:1:1) – baseline
- 35:45:20 – emphasizing triplet loss
- 45:35:20 – emphasizing relevance gain
- 45:20:35 – emphasizing content fidelity
- 20:45:35 – balanced triplet and fidelity
- 30:60:10 – heavily emphasizing triplet loss

k	ΔMRR in.	ΔDIR in.	ΔMRR fe.	ΔDIR fe.
1	0.0072	1.8217	0.0000	1.5303
3	0.0072	2.2337	-0.0006	2.0755
5	0.0072	2.7944	-0.0008	1.9267
10	0.0085	2.2342	-0.0005	1.4795
20	0.0082	-0.5477	-0.0008	-0.9081
30	0.0082	-0.5215	-0.0010	-0.6708

Table 9. ΔMRR@k and ΔDIR@k values at $\alpha = 1$, $\beta = 1$, and $\gamma = 1$

k	ΔMRR in.	ΔDIR in.	ΔMRR fe.	ΔDIR fe.
1	-0.0061	0.7782	-0.0031	1.531
3	0.0070	2.6131	-0.0053	1.3385
5	0.0058	2.7644	-0.0053	2.5133
10	0.0058	2.3594	-0.0058	3.0537
20	0.0061	0.3193	-0.0058	0.6826
30	0.0050	0.5871	-0.0061	0.7782

Table 10. ΔMRR@k and ΔDIR@k values at $\alpha = 0.35$, $\beta = 0.45$, and $\gamma = 0.20$

k	ΔMRR in.	ΔDIR in.	ΔMRR fe.	ΔDIR fe.
1	-0.0051	2.0068	-0.0045	2.5359
3	-0.0048	2.3990	-0.0074	2.1676
5	-0.0047	2.9368	-0.0087	2.1994
10	-0.0037	2.5265	-0.0090	2.0288
20	-0.0033	0.6943	-0.0089	0.9471
30	-0.0033	0.7806	-0.009	0.7327

Table 11. ΔMRR@k and ΔDIR@k values at $\alpha = 0.45$, $\beta = 0.35$, and $\gamma = 0.20$

k	ΔMRR in.	ΔDIR in.	ΔMRR fe.	ΔDIR fe.
1	0.0046	2.2036	0.0002	1.7101
3	0.0042	2.3052	-0.0021	1.8560
5	0.0045	2.5554	-0.0034	2.5287
10	0.0056	2.2332	-0.0043	2.177
20	0.0059	0.9995	-0.0043	0.9982
30	0.0058	0.9697	-0.0045	0.7180

Table 12. ΔMRR@k and ΔDIR@k values at $\alpha = 0.45$, $\beta = 0.20$, and $\gamma = 0.35$

k	ΔMRR in.	ΔDIR in.	ΔMRR fe.	ΔDIR fe.
1	0.0028	1.1920	-0.0014	2.2464
3	0.0031	1.2890	-0.0025	2.4326
5	0.0036	2.6791	-0.0040	2.8285
10	0.0036	2.6109	-0.0048	2.0644
20	0.0039	1.0337	-0.0050	1.5736
30	0.0038	0.9374	-0.0053	0.9407

Table 13. ΔMRR@k and ΔDIR@k values at $\alpha = 0.20$, $\beta = 0.45$, and $\gamma = 0.35$

k	ΔMRR in.	ΔDIR in.	ΔMRR fe.	ΔDIR fe.
1	0.0077	1.7394	-0.0015	2.0998
3	0.0078	2.4325	-0.0023	2.1454
5	0.0066	2.6691	-0.0044	2.9004
10	0.0068	2.2473	-0.0045	2.2760
20	0.0068	0.8375	-0.0046	1.7480
30	0.0068	0.6761	-0.0046	1.7158

Table 14. ΔMRR@k and ΔDIR@k values at $\alpha = 0.30$, $\beta = 0.60$, and $\gamma = 0.10$

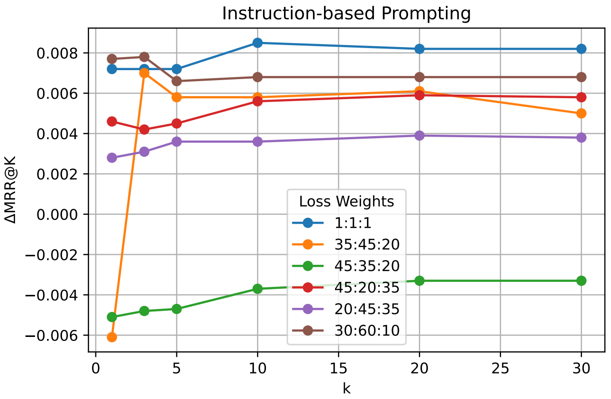

Figure 5. ΔMRR@k metric for instruction-based prompting across varying loss weights.

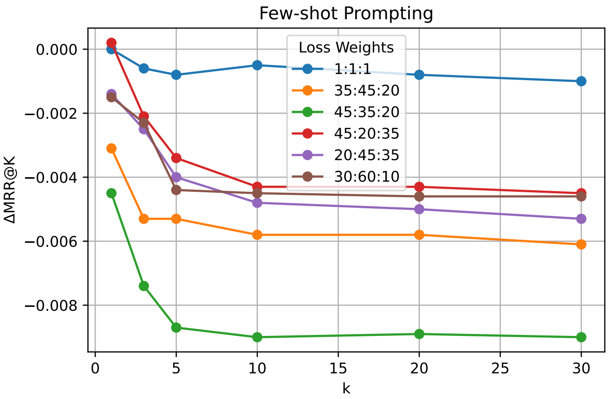

Figure 6. ΔMRR@k metric for few-shot prompting across varying loss weights.

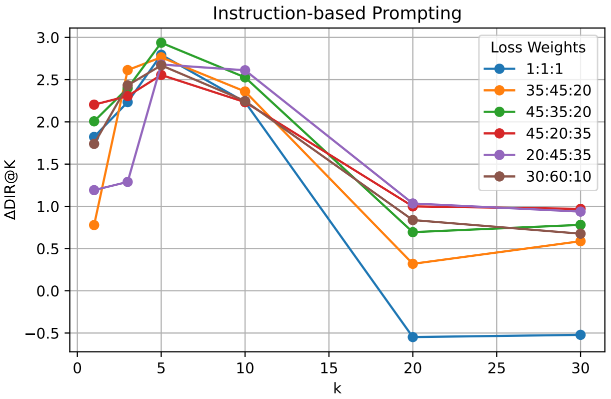

Figure 7. ΔDIR@k metric for instruction-based prompting across varying loss weights..

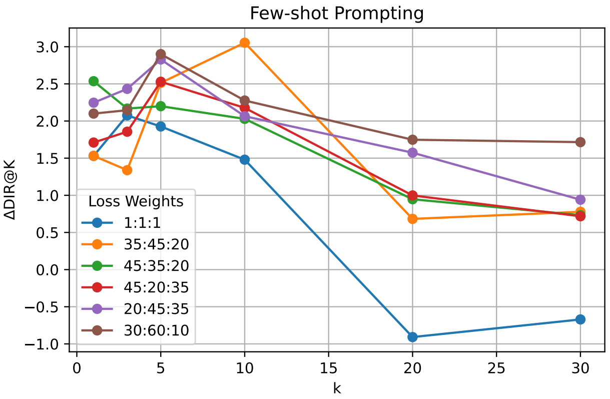

Figure 8. ΔMRR@k metric for few-shot prompting across varying loss weights.

ΔMRR Trends (Tables 9 to 14):

- For *instruction-based* prompting, the equal weighting (1:1:1) achieves the most consistent positive ΔMRR@5 performance (0.0072), outperforming most alternative weightings. The 45:35:20 weighting shows negative ΔMRR values (-0.0047 at k=5) despite achieving the highest ΔDIR performance.

- For *few-shot* prompting, most weightings result in negative ΔMRR values, with the equal weighting (1:1:1) maintaining the least negative impact (ΔMRR@5 = -0.0008). This suggests that few-shot prompting may inherently trade ranking performance for inclusion improvements.

ΔDIR Trends (Tables 9 to 14):

- Under *instruction-based* prompting, the 45:35:20 weighting achieves peak ΔDIR@5 performance (2.9368), surpassing the equal-weighting baseline (2.7944). The 35:45:20 weighting also shows strong inclusion performance (2.7644 at k=5).

- For *few-shot* prompting, the 30:60:10 weighting performs well on inclusion metrics, achieving ΔDIR@5 = 2.9004 and the 35:45:20 weighting achieves strong performance at k=10 (ΔDIR@10 = 3.0537).

6. Discussion

Main comparison at k=5 Our main claim is that *PPO simultaneously boosts rank and inclusion under instruction prompting*. Table 2 confirms this: **PPO lifts** ΔMRR@5 from 0.0051 (prompt engineering) to 0.0073 and increases ΔDIR@5 from 0.9061 to 2.7944. Supervised fine-tuning reduces rank to 0.0022 yet still improves inclusion to 1.6382, indicating that content-fidelity supervision alone cannot fully overcome retriever bias but that lower MRR does not correlate with higher inclusion.

For the *few-shot* prompting (Table 3), PPO continues to increase ΔDIR (+1.93) but lowers ΔMRR (–0.0008), suggesting that rewriting improves the chance of an ad being integrated by an LLM but not its relevance to a query. The rank dip persists for all k values in Table 6, whereas ΔDIR remains positive until k=20, matching typical RAG firstpage windows (Lewis et al., 2021a).

Sensitivity to cut-off k Under instruction prompting (Tables 5, 7), the metrics for PPO usually peak at around k = 5 to k = 10, decreasing beyond k = 20. This underscores that rewriting may become less effective in retrieval ranking and inclusion as less relevant ads are being ranked and the LLM has more ads to choose from.

Ablation: k value Optimal retrieval occurs at moderate values of k = 5 to k = 10 because large values of k introduce noise and diminish relevance. Removing PPO and retaining only SFT drops ΔMRR by 60–70% yet still improves ΔDIR, confirming that *content-fidelity supervision alone* cannot reliably raise rank (Dong et al., 2024).

Ablation: Loss weighting The equal weighting (1:1:1) provides the most balanced performance across both ranking and inclusion metrics, suggesting that all three components contribute meaningfully to effective ad rewriting (Huq & Pervin, 2022; Chen et al., 2018). $L_{\text{rel\_gain}}$ directly optimizes query–document similarity, explaining why higher α values can improve inclusion rates (Ha & Blanz, 2021). L_{triplet} helps ads outperform competing documents, which is crucial in competitive retrieval scenarios, as shown by the high inclusion (few-shot) and relatively high ΔMRR (instruction-based) in Tables 10 and 14. L_{fidelity} prevents semantic drift, but can limit the model's ability to make strategic keyword choices (Chaturvedi et al., 2024; Matton et al., 2025).

Why does PPO help? PPO optimizes a composite reward balancing query similarity, distractor margin, and fidelity, leveraging its stability for LLM alignment (Schulman et al., 2017; Ram et al., 2023). This surfaces high-IDF keywords valued by the retriever while preserving semantics, hence the joint lift in ΔMRR and ΔDIR (Ziegler et al., 2020).

Inclusion vs. ranking. An important consideration for practical deployment is that inclusion in LLM responses (ΔDIR) may be more commercially valuable than ranking position (ΔMRR). Recent RAG-evaluation studies emphasise that the text actually surfaced in the answer, rather than its raw retrieval rank, drives user value and system quality (Yu et al., 2025; Brehme et al., 2025). Because users never see the internal ranking scores but do see which passages are quoted, optimizing ΔDIR is often more aligned with real-world objectives (Es et al., 2025). Our own results echo this: configurations such as 45:35:20 (instruction) and 35:45:20 (few-shot) push inclusion strongly even when ranking gains are modest or negative.

Strategy-specific Instruction prompting typically supplies richer task signals than few-shot demonstrations, which explains why our instruction-based run with α=0.45, β=0.35, γ=0.20 achieves the highest ΔDIR@5 (Wei et al., 2023a; Wang et al., 2023a). In contrast, few-shot exemplars anchor the model to brevity and marketing tone; the fidelity weight over-penalizes lexical deviation, reducing keyword coverage, and hurting rank, a pattern also observed in ranking work on few-shot prompting (Sinhababu et al., 2024) and lexical overlap analyses of dense retrievers (Ram et al., 2023). Few-shot setups benefit from emphasizing hard-negative discrimination (β) as in the 35:45:20 setting, mirroring findings that chain-of-thought exemplars can over-anchor lexical style and hurt precision unless balanced by discriminative losses (Wang et al., 2023b).

- **Instruction-based prompting:** $\alpha = 0.45, \beta = 0.35, \gamma = 0.20$ achieves peak ΔDIR@5 = 2.9368.
- **Few-shot prompting:** $\alpha = 0.35$, $\beta = 0.45$, $\gamma = 0.20$ achieves peak ΔDIR@10 = 3.0537.

The results demonstrate a clear **trade-off between ranking and inclusion performance**. Most alternative weightings improve inclusion at the expense of ranking performance, with only the equal weighting maintaining positive ranking improvements while achieving competitive inclusion rates. Practitioners should therefore pick weights according to business goals: equal weighting for balanced improvements, or inclusion-focused weightings when ad visibility in LLM responses is the primary concern.

7. Conclusion

In this study, we introduced a novel approach to improve ad visibility in LLM retrieval systems through content rewriting, without modifying the retriever itself. Our framework leverages Proximal Policy Optimization to improve retrieval ranking and inclusion in the LLM response. By creating a custom composite loss that maximizes semantic relevance and content fidelity, we demonstrate how language models can be fine-tuned to make commercially effective rewrites. Empirically, we show that our PPO enhanced models generally outperform both supervised fine-tuning and prompt engineering baselines on two key metrics, ΔMRR@k and ΔDIR@k, over multiple k values and two prompting strategies, suggesting that LLM rewriting and optimization is an effective approach to improve ad visibility in language model pipelines. As LLMs continue to power search, recommendation, and generation pipelines, our results indicate that phrasing optimization can improve retrieval output for black-box systems. Our hope is that this method can serve as a foundation for future work on retrieval-aware and LLM alignment, especially in domains where visibility directly influences utility or monetization.

References

Alinejad, A., Kumar, K., and Vahdat, A. Evaluating the retrieval component in llm-based question answering systems, 2024. URL https://arxiv.org/abs/2406.06458.

Brehme, L., Ströhle, T., and Breu, R. Can llms be trusted for evaluating rag systems? a survey of methods and datasets, 2025. URL https://arxiv.org/abs/2504.20119.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners, 2020. URL https://arxiv.org/abs/2005.14165.

Chaturvedi, A., Bhar, S., Saha, S., Garain, U., and Asher, N. Analyzing semantic faithfulness of language models via input intervention on question answering. *Computational Linguistics*, 50(1):119–155, 2024. ISSN 1530-9312. doi: 10.1162/coli_a_00493. URL http://dx.doi.org/10.1162/coli_a_00493.

Chen, Z., Badrinarayanan, V., Lee, C.-Y., and Rabinovich, A. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks, 2018. URL https://arxiv.org/abs/1711.02257.

Dash, A., Ghosh, S., Mukherjee, A., Chakraborty, A., and Gummadi, K. P. Sponsored is the new organic: Implications of sponsored results on quality of search results in the amazon marketplace, 2024. URL https://arxiv.org/abs/2407.19099.

Dong, G., Yuan, H., Lu, K., Li, C., Xue, M., Liu, D., Wang, W., Yuan, Z., Zhou, C., and Zhou, J. How abilities in large language models are affected by supervised fine-tuning data composition, 2024. URL https://arxiv.org/abs/2310.05492.

Es, S., James, J., Espinosa-Anke, L., and Schockaert, S. Ragas: Automated evaluation of retrieval augmented generation, 2025. URL https://arxiv.org/abs/2309.15217.

Ha, M. L. and Blanz, V. Deep ranking with adaptive margin triplet loss, 2021. URL https://arxiv.org/abs/2107.06187.

Haans, H., Raassens, N., and Hout, R. Search engine advertisements: The impact of advertising statements on click-through and conversion rates. *Marketing Letters*, 24, 06 2013. doi: 10.1007/s11002-013-9226-5.

Huq, A. and Pervin, M. T. Adaptive weight assignment scheme for multi-task learning. *IAES International Journal of Artificial Intelligence (IJ-AI)*, 11(1):173, March 2022. ISSN 2089-4872. doi: 10.11591/ijai.v11.i1.pp173-178. URL http://dx.doi.org/10.11591/ijai.v11.i1.pp173-178.

Interactive Advertising Bureau (IAB) and PwC. IAB/PwC Internet Advertising Revenue Report: Full Year 2024. https://www.iab.com/wp-content/uploads/2025/04/IAB_PwC-Internet-Ad-Revenue-Report-Full-Year-2024.pdf, 2025. Accessed: 2025-05-18.

Kulkarni, M., Tangarajan, P., Kim, K., and Trivedi, A. Reinforcement learning for optimizing rag for domain chatbots, 2024. URL https://arxiv.org/abs/2401.06800.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., tau Yih, W., Rocktäschel, T., Riedel, S., and Kiela, D. Retrieval-augmented generation for knowledge-intensive nlp tasks, 2021a. URL https://arxiv.org/abs/2005.11401.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., tau Yih, W., Rocktäschel, T., Riedel, S., and Kiela, D. Retrieval-augmented generation for knowledge-intensive nlp tasks, 2021b. URL https://arxiv.org/abs/2005.11401.

Liu, Y., Hu, X., Zhang, S., Chen, J., Wu, F., and Wu, F. Fine-grained guidance for retrievers: Leveraging llms' feedback in retrieval-augmented generation, 2024. URL https://arxiv.org/abs/2411.03957.

MacAvaney, S., Nardini, F. M., Perego, R., Tonellotto, N., Goharian, N., and Frieder, O. Expansion via prediction of importance with contextualization. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, pp. 1573–1576. ACM, July 2020. doi: 10.1145/3397271.3401262. URL http://dx.doi.org/10.1145/3397271.3401262.

Matton, K., Ness, R. O., Guttag, J., and Kıcıman, E. Walk the talk? measuring the faithfulness of large language model explanations, 2025. URL https://arxiv.org/abs/2504.14150.

Nijkamp, E., Pang, B., Hayashi, H., Tu, L., Wang, H., Zhou, Y., Savarese, S., and Xiong, C. Codegen: An open large language model for code with multi-turn program synthesis, 2023. URL https://arxiv.org/abs/2203.13474.

Nogueira, R. and Cho, K. Passage re-ranking with bert, 2020. URL https://arxiv.org/abs/1901.04085.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J., and Lowe, R. Training language models to follow instructions with human feedback, 2022. URL https://arxiv.org/abs/2203.02155.

Ram, O., Bezalel, L., Zicher, A., Belinkov, Y., Berant, J., and Globerson, A. What are you token about? dense retrieval as distributions over the vocabulary, 2023. URL https://arxiv.org/abs/2212.10380.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

Sinhababu, N., Parry, A., Ganguly, D., Samanta, D., and Mitra, P. Few-shot prompting for pairwise ranking: An effective non-parametric retrieval model, 2024. URL https://arxiv.org/abs/2409.17745.

Wang, B., Min, S., Deng, X., Shen, J., Wu, Y., Zettlemoyer, L., and Sun, H. Towards understanding chain-of-thought prompting: An empirical study of what matters, 2023a. URL https://arxiv.org/abs/2212.10001.

Wang, L., Xu, W., Lan, Y., Hu, Z., Lan, Y., Lee, R. K.-W., and Lim, E.-P. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models, 2023b. URL https://arxiv.org/abs/2305.04091.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., and Zhou, D. Chain-of-thought prompting elicits reasoning in large language models, 2023a. URL https://arxiv.org/abs/2201.11903.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., and Zhou, D. Chain-of-thought prompting elicits reasoning in large language models, 2023b. URL https://arxiv.org/abs/2201.11903.

Yu, H., Gan, A., Zhang, K., Tong, S., Liu, Q., and Liu, Z. *Evaluation of Retrieval-Augmented Generation: A Survey*, pp. 102–120. Springer Nature Singapore, 2025. ISBN 9789819610242. doi: 10.1007/978-981-96-1024-2_8. URL http://dx.doi.org/10.1007/978-981-96-1024-2_8.

Ziegler, D. M., Stiennon, N., Wu, J., Brown, T. B., Radford, A., Amodei, D., Christiano, P., and Irving, G. Fine-tuning language models from human preferences, 2020. URL https://arxiv.org/abs/1909.08593.

A. Appendix

A.1. Prompt Design

Black-Box LLM that Integrates the Ads Prompt

You are a helpful and knowledgeable assistant. Below is a list of product advertisements. Your task is to respond to the user's query in a natural and informative way, promoting one or two relevant products. Include brand and url into your response without revealing that you have access to product ads.

At the end of your response (not visible to the user), list which documents you included clearly only using the format: 'id: ..., id: ..., ...'.

USER QUERY: query
RETRIEVED COMMERCIAL ADS: context

Please provide a helpful, informative response directed to the user based on the above information.

General prompting strategy

You are given an advertisement. Your task is to rewrite it so that its ranking in retrieval and inclusion in LLM response improves. Focus on semantic relevance and matching the user's likely search intent.

Original Ad: ad

Respond with the improved version:

Title: ...

Description: ...

Instruction-based Prompting with CoT (Zero-Shot)

You are given an advertisement. Your task is to rewrite the ad so that its ranking in retrieval and inclusion in LLM responses improves. Focus on semantic relevance and matching the user's likely search intent.

Original Ad: ad

Think step by step first, then provide the improved version.

Respond with the improved version at the end of your response **only** in the following format:

Thinking: ...
Title: ...
Description: ...

Few-Shot Prompting with CoT

Rewrite the advertisement so that it ranks better in retrieval and its inclusion in LLM responses improves. Here are two examples:

Example 1

Original Ad: Title: Yoga Pants Description: Affordable yoga pants for women, available in multiple colors.

Reasoning: The phrase ”affordable yoga pants” is generic. Adding activity-specific and quality-based terms may help.

Rewritten Ad: Title: High-performance women’s yoga leggings Description: great yoga pants for training and Pilates – breathable, colorful, and comfortable.

Example 2

Original Ad: Title: mugs Description: Buy custom mugs with your name.

Reasoning: This lacks variety and emotional appeal. Including gifting context and materials can help retrieval.

Rewritten Ad: Title: Personalized ceramic mugs Description: perfect gifts with names, photos, or messages.

Your turn

Original Ad: ad

Reasoning:

Rewritten Ad:

A.2. PPO Hyper-Parameters

Table 15 lists every setting required to reproduce our RL stage.

Category	**Symbol / Key**	**Value**
Clip range	ϵ	0.2
KL penalty	β	0.1
PPO epochs / batch	—	4
Reward discount	γ	1.0
Micro-batch size	m	4
Grad-accumulation	g	4
Effective batch	$m \times g$	16
Learning rate	η	1×10^{-5}
Scheduler	—	Cosine
Warm-up ratio	—	0.10
Train epochs	—	3
Dataset size	N	10 000 ads
Total PPO updates†	$\lceil 3N/(16 \times N_{\text{GPU}}) \rceil$	1 875 (1 GPU)

Table 15. PPO settings for LLaMA-3.1-8B. †Divide by N_{GPU} to scale.

A.3. Code and Data Availability

To foster reproducibility while preserving anonymity, we release our implementation in two static repositories:

- https://anonymous.4open.science/r/ad-ppo-lora-2706
- https://anonymous.4open.science/r/ad-doc-reranker-57C6