Analytical Report: Improving Generative Ad Text on Facebook using Reinforcement Learning

NLP Course Project

February 6, 2026

Abstract

This report analyzes the paper “Improving Generative Ad Text on Facebook using Reinforcement Learning” (arXiv:2507.21983v2). The paper studies a production deployment of an RL-trained large language model (LLM) for generating ad text on Facebook. It introduces Reinforcement Learning with Performance Feedback (RLPF), which uses historical ad performance as a reward signal. A large-scale A/B test on nearly 35,000 advertisers shows a statistically significant relative lift in advertiser-level click-through rate (CTR). This report explains the problem, data, method, experiment, results, limitations, and provides a small prototype plan aligned with the course requirements.

1 Problem and Motivation

Large language models (LLMs) require post-training to align them with real-world objectives. Much of the post-training literature focuses on human preferences (e.g., RLHF), but the economic impact of post-training is less understood. The paper asks whether reinforcement learning with real performance metrics can improve outcomes in a high-stakes product setting: ad text generation. In online advertising, even small CTR improvements translate into meaningful economic gains, making this domain a strong testbed for post-training impact.

2 System Inputs and Outputs

The product is a text generation feature in Meta Ads Manager. The system:

- **Input:** A human-written ad text provided by the advertiser.
- **Output:** Multiple AI-generated ad text variations that the advertiser can select and edit.

The core objective is not to produce a single best rewrite but to supply a set of high-quality variants for human selection.

3 Data

The method depends on historical ad performance data where only the text varies while other ad components (image, targeting, budget) are held constant. This “multitext” setup allows performance differences to be attributed to text changes.

Key details:

- **Type:** Historical ad text pairs with different CTR values.
- **Source:** Meta/Facebook ad platform data.
- **Scale:** A/B test over approximately 35,000 advertisers and 640,000 ad variations during a 10-week period.

4 Method: Reinforcement Learning with Performance Feedback (RLPF)

RLPF combines reward modeling with policy optimization:

1. **Reward Model.** Use historical multitext data to construct preference pairs: given two ad texts with different CTRs, the higher-CTR text is labeled as preferred. Train a reward model to assign higher scores to texts with better performance.
2. **RL Fine-tuning.** Use the reward model as a proxy environment. Fine-tune the LLM using Proximal Policy Optimization (PPO) to maximize reward while staying close to a reference model (via KL penalty). A length penalty discourages overly long ad text.

   A simplified objective is:

$$\max_{\pi_\phi} \mathbb{E}_{x \sim D,\, y \sim \pi_\phi(y|x)} \left[r_\theta(x, y) - \beta \mathrm{KL}(\pi_\phi || \pi_{ref}) - \alpha \cdot \mathrm{length}(y) \right] \tag{1}$$

5 Baselines and Model Variants

The paper compares:

- **Imitation LLM v1:** SFT on synthetic rewrites.
- **Imitation LLM v2:** SFT on synthetic + human-written rewrites.
- **AdLlama (proposed):** Imitation v2 further trained with RLPF.

All models are based on Llama 2 Chat 7B.

6 Experiment Design

A randomized A/B test evaluates the impact of RLPF:

- **Period:** February 16, 2024 to April 25, 2024 (10 weeks).
- **Population:** 34,849 US advertisers.
- **Assignment:** Advertiser-level randomization to control (Imitation v2) or treatment (AdLlama).
- **Metrics:** Advertiser-level CTR, clicks, impressions, number of ads, number of ad variations.

The main analysis uses log-binomial regression with covariates to estimate CTR lift.

7 Results

The primary result is a statistically significant improvement in advertiser-level CTR:

- **Relative CTR lift:** +6.7% for AdLlama vs. Imitation v2 ($p = 0.0296$).
- **Absolute CTR:** approximately 3.1% to 3.3%.
- **Ad variations:** +18.5% increase in the number of text variants created.
- **Number of ads:** no statistically significant change.

These gains are meaningful in a mature advertising platform where large CTR gains are difficult to achieve.

8 Limitations

The authors highlight several constraints:

- **Offline-only RL:** The reward model is trained on historical data; there is no real-time learning loop.
- **Single objective:** CTR is optimized without explicit constraints on tone, creativity, or advertiser preference.
- **Human selection not modeled:** The reward does not consider which AI suggestions advertisers actually choose.
- **Platform-level effects:** Broader impacts (ad diversity, user experience) are not evaluated.

9 Course Prototype (Planned Implementation)

The paper does not release code or public data. For this course project, we propose a prototype that approximates the RLPF pipeline:

- Use the Meta Ad Library API to collect real ad text (political/issue ads). When API access is unavailable, use synthetic ad text and CTR proxies.
- Train a reward model on text-performance pairs (TF-IDF + Ridge baseline).
- Generate multiple text variants using template-based transformations.
- Select the best variant by reward score and report improvements.

This prototype does not replace online RL but demonstrates the core logic of performance-driven post-training.

10 Prototype Outputs and Interpretation

The prototype produces several artifacts to make results transparent:

- **Notebook outputs (demo/):** The notebook loads data (real or synthetic), trains a reward model, simulates RLPF-style selection, and plots a histogram of reward improvements. The histogram shows how often the selected variation improves the reward score over the original text, and the distribution of improvement magnitudes.
- **Run summary JSON (logs/summary_*.json):** A compact aggregate report with:
  - `count`: number of evaluated texts.
  - `mean_original`: average reward of the original texts.
  - `mean_best`: average reward of the best selected variation.
  - `mean_delta`: mean improvement (`best - original`).
  - `median_delta`: median improvement.
  - `improvement_rate`: fraction of cases where the selected text scores higher than the original.
- **Agent trace JSON (logs/agent_runs_*.json):** Two example runs showing the original text, generated variations, reward scores, and the selected best text. This is a qualitative check that the selection behavior is sensible.
- **Ads CSV (logs/ads_*.csv):** A snapshot of the processed dataset with text, proxy scores, and metadata. This can be used for further analysis or auditing.

These outputs provide both quantitative and qualitative evidence of the prototype's behavior.

11 Conclusion

This paper provides one of the largest real-world evaluations of RL-based post-training for LLMs. By aligning a model with performance feedback, the authors demonstrate a measurable CTR lift and increased advertiser adoption. The work suggests that aggregate performance metrics can serve as scalable rewards for LLM alignment, bridging the gap between model capability and business impact.