ADVERSARIAL LEARNING FOR FAIR LLM FINE-TUNING FROM IMBALANCED TRAINING SIGNALS

Anonymous authors
Paper under double-blind review

ABSTRACT

Fine-tuning large language models (LLMs) typically optimizes average utility, but such improvements can mask severe regressions for underrepresented or intrinsically difficult prompt groups. We observe this failure mode in personalized recipe generation, where naive fine-tuning substantially degrades performance for less frequent cuisines and prompts including rare ingredients. To address this problem, we propose an adversarial policy learning framework for LLM fine-tuning that directly targets worst-group performance. Our method alternates between adaptively reweighting training samples and updating the policy to maximize the resulting weighted reward. The framework supports multiple fairness objectives within a unified optimization scheme, including protecting prespecified groups, improving robustness without predefined group labels, and enforcing uniform relative improvements over a reference policy. Experiments on recipe generation and content summarization show that our method consistently improves worst-group performance over naive and inverse-weighting baselines, with little degradation in average performance, thereby mitigating severe group-level regressions under imbalanced data. These results demonstrate the importance of moving beyond average-reward optimization when fine-tuning LLMs from heterogeneous data.

1 INTRODUCTION

Fine-tuning large language models (LLMs) is a key mechanism for adapting model behavior to human preferences, downstream tasks, and domain-specific objectives (Christiano et al., 2017; Ziegler et al., 2019; Rafailov et al., 2023). LLM fine-tuning can often benefit frequent or easier prompt regions while providing smaller gains, or even causing regressions, for underrepresented or more difficult ones. The practical question is thus not only whether model fine-tuning improves average utility, but whether it improves the model *robustly* across heterogeneous regions of the prompt space.

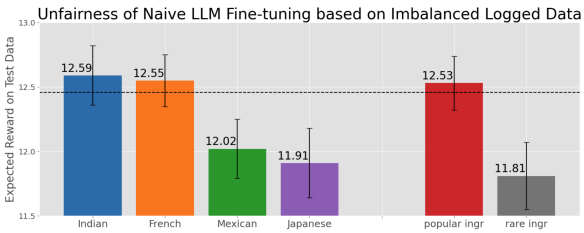


Figure 1: Group-specific performance degradation in recipe generation after fine-tuning on imbalanced logged data. The dotted line shows the global average test reward. The model performs better on overrepresented Indian and French recipes than on underrepresented Mexican and Japanese recipes, and also performs worse on recipes containing rare ingredients than on those with common ingredients, highlighting fairness issues for underrepresented and difficult groups.

Average improvement can conceal substantial regressions for particular subsets of prompts. Training feedback in real-world environments is rarely balanced across languages, domains, user segments, or levels of input difficulty (Mehrabi et al., 2021). Some groups may also be systematically harder than others, exhibiting lower rewards or greater reward variability even when their frequency is comparable. Under such heterogeneity, expected-reward optimization can concentrate learning on frequent or easier prompts while sacrificing underrepresented or difficult groups (Cui et al., 2019; Das et al., 2024; Doan et al., 2024). This failure mode is especially concerning when the performance of the weakest group matters independently of overall average utility.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Figure 1 illustrates this issue in personalized recipe generation. We construct a training dataset in which prompts vary by cuisine and ingredient constraints, and evaluate generated recipes using an LLM-based evaluator along four dimensions: clarity, completeness, constraint adherence, and practicality. The training is imbalanced, containing substantially more feedback for Indian and French recipes than for Mexican and Japanese recipes. After naively fine-tuning the base LLM to maximize expected reward, performance becomes strongly group dependent; Japanese recipes score approximately 5.4% lower than high-frequency Indian recipes, while recipes requiring rare ingredients score approximately 5.8% lower than those using common ingredients. Thus, fine-tuning that appears beneficial in aggregate can still leave particular prompt groups substantially worse served. Related performance disparities have been studied in recommender systems, information retrieval, and LLMs (Singh & Joachims, 2019; Morik et al., 2020; Wang et al., 2023; Kearns et al., 2018; Doan et al., 2024; Liang et al., 2022); Appendix A provides an extended discussion on related work.

A natural response to group imbalance is *inverse weighting*, which scales each training sample by the inverse empirical frequency of its group (Elkan, 2001; Cui et al., 2019; Precup et al., 2000). While this compensates for unequal representation, frequency alone does not determine which groups require additional optimization. A frequent group may still be difficult, whereas a rare group may already perform well. Inverse weighting also requires predefined group labels and can produce high-variance model updates when group frequencies are highly skewed. More generally, static reweighting cannot adapt as the identity of the worst-performing group changes during fine-tuning.

We therefore propose an *adversarial and adaptive policy learning framework for fair LLM fine-tuning*. Rather than fixing sample weights in advance, our method alternates between two coupled updates; an adversary adjusts the importance assigned to training samples according to current policy performance, and the LLM policy is updated to maximize the resulting weighted reward. This adaptive interaction during fine-tuning directs optimization pressure toward poorly performing regions of the prompt distribution instead of allocating weight solely according to their frequency.

Importantly, our method accommodates three complementary notions of fairness.[1] *Group-aware fairness* explicitly protects designated groups when relevant labels are available, such as particular languages, jurisdictions, or customer segments. *Group-agnostic fairness* learns context-dependent adversarial weights without requiring prespecified group labels, providing robustness to heterogeneous prompt slices that may not be known in advance. Finally, *relative fairness* seeks balanced improvements over a given reference policy rather than equal absolute performance, which is particularly relevant in production settings where fine-tuning should avoid regressions for specific groups.

Our contributions are threefold. First, we formulate fair LLM fine-tuning from imbalanced data as a policy learning problem that explicitly targets worst-group performance rather than average reward alone. Second, we develop an adversarial and adaptive reweighting framework that unifies group-aware, group-agnostic, and relative fairness objectives within a common alternating optimization procedure. Finally, we evaluate the proposed algorithm on recipe generation and content summarization, demonstrating improved worst-group performance and more uniform relative improvements than naive and inverse-weighting baselines under substantial data imbalance.

2 PRELIMINARIES

We consider LLM fine-tuning from logged training data, including historical instructions x, responses a generated by an already deployed LLM π_0, and scalar rewards r associated with those observed responses (Swaminathan & Joachims, 2015; Kiyohara et al., 2025). Let $\mathcal{X}$ denote the instruction space and $\mathcal{A}$ the output space. We are given $\mathcal{D} = \{(x_i, a_i, r_i)\}_{i=1}^n$, where $x_i \sim \mathcal{P}_X$, $a_i \sim \pi_0(\cdot \mid x_i)$ is generated by an old model π_0, and $r_i \in \mathbb{R}$ is the observed reward. Note that $\pi(a \mid x)$ denotes the probability assigned by the LLM policy π to the output sequence a given input x. Rewards may represent explicit ratings, clicks, engagement signals, or scores produced by an LLM-as-a-Judge; we assume only that larger rewards indicate better outputs.

For stating robustness goals, let $q(x, a) = \mathbb{E}[r \mid x, a]$ denote the unknown expected reward and define the population value of a target model π_θ as $V(\pi_\theta) = \mathbb{E}_{x\sim\mathcal{P}_X,\, a\sim\pi_\theta(\cdot|x)}[q(x, a)]$. These population quantities characterize the aspects of model behavior that we would like fine-tuning to

[1] Throughout this work, we use fairness to refer to disparities in task performance or improvement across prompt groups or regions of the prompt space, rather than demographic parity or representational fairness.

improve. The practical algorithm in Section 3 uses a reward-weighted likelihood objective over the logged data and requires no additional reward annotations beyond those included in $\mathcal{D}$.

Instruction tuning as a special case. Standard instruction tuning is a simple instance of the LLM fine-tuning problem. If each observed output a_i is a demonstration regarded as desirable and assigned a uniform reward $r_i = 1$, the reward-weighted objective reduces to maximizing demonstration likelihood, $\max_\theta \sum_{i=1}^n \log \pi_\theta(a_i \mid x_i)$. This lets the same robust reweighting mechanism be used in both heterogeneous-reward fine-tuning and uniform-reward instruction tuning.

2.1 Fairness in Fine-Tuning

The instruction distribution $\mathcal{P}_X$ can be highly heterogeneous; some instruction groups may be substantially more frequent than others, while some may be intrinsically more difficult. As illustrated by the recipe-generation example in Section 1, optimizing overall average reward can therefore conceal severe degradation for particular groups.

Let $\mathcal{G}$ denote a set of instruction groups, such as groups defined by domain, topic, input difficulty, or other task-specific attributes (e.g., cuisine type or ingredient rarity in recipe generation), and let $g(x) \in \mathcal{G}$ denote the group associated with instruction x when such labels are available. We define the value of LLM policy π_θ on group g as

$$V_g(\pi_\theta) = \mathbb{E}_{x\sim\mathcal{P}_X(\cdot|g),\ a\sim\pi_\theta(\cdot|x)}[q(x,a)].$$

We consider three complementary notions of fairness stated below.

Group-aware fairness. When the relevant groups are known, we seek to improve the least-performing group as

$$\max_{\pi_\theta} \min_{g\in\mathcal{G}} V_g(\pi_\theta).$$

This objective is appropriate when explicit guarantees are desired for predefined cohorts, such as languages, domains, or customer segments.

Group-agnostic (universal) fairness. When meaningful group labels are unavailable, incomplete, or undesirable to use, we consider the more conservative instruction-level objective as follows.

$$\max_{\pi_\theta} \inf_{x\in\mathrm{supp}(\mathcal{P}_X)} \mathbb{E}_{a\sim\pi_\theta(\cdot|x)}[q(x,a)].$$

Rather than specifying protected groups in advance, this objective protects poorly performing regions of the instruction space and can therefore provide robustness to arbitrary post-hoc slices induced by prompt properties such as topic, length, or difficulty.

Relative fairness. Absolute group values may differ substantially because of intrinsic differences in task difficulty. When a reference policy π_{ref} is available, such as the currently deployed LLM, we can instead consider the worst group-level improvement as

$$\max_{\pi_\theta} \min_{g\in\mathcal{G}} \left\{V_g(\pi_\theta) - V_g(\pi_{\mathrm{ref}})\right\}.$$

This objective seeks to distribute improvements more uniformly across groups and directly discourages regressions relative to a given reference policy. It is particularly relevant to LLM fine-tuning in production environments, where preserving performance for all groups can be more meaningful than equalizing their absolute reward levels.

2.2 Baseline Methods and Their Limitations

A simple baseline is *naive fine-tuning*, which ignores group structure and applies reward-weighted updates uniformly across the training data. We write its empirical surrogate objective as $\widehat{J}_{\mathrm{naive}}(\theta) = \frac{1}{n}\sum_{i=1}^n r_i \log \pi_\theta(a_i|x_i)$, with gradient $\nabla_\theta \widehat{J}_{\mathrm{naive}}(\theta) = \frac{1}{n}\sum_{i=1}^n r_i \nabla_\theta \log \pi_\theta(a_i|x_i)$. Because every logged training sample enters symmetrically, frequent groups exert greater aggregate influence on

the update. Consequently, high average performance can coexist with substantial degradation for underrepresented or difficult groups, as already illustrated in Figure 1.

A second baseline is *inverse weighting*. Let g_i denote the group of x_i and $\hat{p}_{g_i}$ its empirical frequency in the training data. The objective weights each example inversely to its group frequency, i.e., $\widehat{J}_{\mathrm{IW}}(\theta) = \frac{1}{n}\sum_{i=1}^{n} \frac{r_i}{\hat{p}_{g_i}} \log \pi_\theta(a_i|x_i)$, yielding $\nabla_\theta \widehat{J}_{\mathrm{IW}}(\theta) = \frac{1}{n}\sum_{i=1}^{n} \frac{r_i}{\hat{p}_{g_i}} \nabla_\theta \log \pi_\theta(a_i|x_i)$. Inverse weighting compensates for unequal observation frequencies, but frequency is not necessarily aligned with performance; a frequent group may remain difficult, while a rare group may already perform well. The method also requires predefined group labels and can yield high-variance updates when $\hat{p}_g$ is small. Most importantly, its weights are fixed before optimization and thus cannot adapt as the identity of the worst-performing group changes during fine-tuning.

These limitations motivate our adversarial and adaptive approach, which learns the weighting distribution jointly with the LLM policy.

3 The Proposed Adversarial Fine-tuning Method

We now introduce our adversarial fine-tuning method for realizing the three fairness objectives defined in Section 2. The key idea is to replace fixed sample weighting with an adaptive adversary that assigns greater importance to groups or instructions on which the current LLM performs poorly.

Adversarially Weighted LLM Fine-tuning. Let $w_\phi(g_i)$ denote the weight assigned by an adversary with parameters ϕ to the prompt group associated with training sample i. Larger weights give the corresponding samples greater influence on the LLM policy update, allowing the adversary to adaptively emphasize poorly performing groups. Given the training dataset $\mathcal{D} = \{(x_i, a_i, r_i)\}_{i=1}^{n}$, we define the adversarially weighted policy objective as

$$\widehat{J}(\theta, \phi) = \frac{1}{n}\sum_{i=1}^{n} w_\phi(g_i) r_i \log \pi_\theta(a_i \mid x_i).$$

The LLM policy is updated by gradient ascent, $\theta_{t+1} \leftarrow \theta_t + \eta \nabla_\theta \widehat{J}(\theta_t, \phi_t)$, where $\nabla_\theta \widehat{J}(\theta, \phi) = \frac{1}{n}\sum_{i=1}^{n} w_\phi(g_i) r_i \nabla_\theta \log \pi_\theta(a_i \mid x_i)$. Thus, samples assigned larger adversarial weights exert greater influence on the LLM policy update. We parameterize the weights in the above objective as

$$w_\phi(g_i) = b + n\frac{f_\phi(g_i)}{\sum_{j=1}^{n} f_\phi(g_j)}, \tag{1}$$

where f_ϕ is the adversary and $b > 0$ is a stability parameter that prevents any sample or group from receiving zero weight. The normalized component allows the adversary to redistribute optimization emphasis while b preserves a minimum contribution from every example. We use $b = 1$ by default and study its sensitivity empirically in Section 4.

Adversary Update. The adversary is trained in the opposite direction from the LLM policy; it seeks to place greater weight on samples associated with lower current policy value. Specifically, for the group-aware objective, we minimize

$$\widehat{L}_{\mathrm{adv}}(\phi; \theta) = \frac{1}{n}\sum_{i=1}^{n} w_\phi(g_i) \pi_\theta(a_i \mid x_i) r_i. \tag{2}$$

Because ϕ is optimized to minimize this weighted value, the adversary is encouraged to concentrate weight on groups for which the current policy parameterized by θ performs poorly. The resulting policy update then places correspondingly greater emphasis on those groups.

The same mechanism supports the three fairness notions introduced in Section 2. For **group-aware fairness**, the adversary receives the known group identifier and produces $w_\phi(g_i)$, thereby adaptively emphasizing poorly performing designated groups. For **group-agnostic (universal) fairness**, we instead use prompt- or instruction-dependent weights $w_\phi(x_i)$, allowing the adversary to identify poorly performing regions of the input space without requiring predefined group labels.

Algorithm 1 Adversarial Fair LLM Fine-tuning

Initialize LLM policy parameters θ and adversary parameters ϕ
Choose weighting mode (GROUPAWARE or UNIVERSAL) and fairness objective (ABSOLUTE or RELATIVE)
Set LLM policy learning rate η, adversary learning rate γ, and stability parameter b
for each training step **do**
  Sample minibatch $\mathcal{B} = \{(x_i, a_i, r_i, g_i)\}$
  Set $k_i \leftarrow g_i$ for GROUPAWARE, or $k_i \leftarrow x_i$ for UNIVERSAL
  Compute adversarial scores $s_i \leftarrow f_\phi(k_i)$
  Normalize weights: $w_i \leftarrow b + |\mathcal{B}| \frac{s_i}{\sum_{j \in \mathcal{B}} s_j}$
  Update the policy: $\theta \leftarrow \theta + \eta \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} w_i r_i \nabla_\theta \log \pi_\theta(a_i \mid x_i)$
  if RELATIVE **then**
    $\tilde{y}_i \leftarrow (\pi_\theta(a_i \mid x_i) - \pi_{\text{ref}}(a_i \mid x_i))\, r_i$
  else
    $\tilde{y}_i \leftarrow \pi_\theta(a_i \mid x_i) r_i$
  end if
  Update the adversary: $\phi \leftarrow \phi - \gamma \nabla_\phi \left[\frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} w_\phi(k_i) \tilde{y}_i + \Omega(\phi) \right]$
end for

For **relative fairness**, we replace absolute performance with improvement over a given reference policy π_{ref}. In this case, the adversary objective becomes

$$\widehat{L}^{\text{rel}}_{\text{adv}}(\phi; \theta, \pi_{\text{ref}}) = \frac{1}{n} \sum_{i=1}^{n} w_\phi(g_i) \left(\pi_\theta(a_i \mid x_i) - \pi_{\text{ref}}(a_i \mid x_i)\right) r_i. \tag{3}$$

Minimizing this objective directs adversarial mass toward groups exhibiting the smallest improvement relative to the reference LLM policy π_{ref}. The subsequent policy update therefore prioritizes these groups, encouraging improvements to be distributed more uniformly rather than concentrating gains on groups that already benefit substantially from fine-tuning. The same relative formulation can be combined with instruction-dependent weights when explicit group labels are unavailable.

Alternating Optimization. The complete procedure of our method is summarized in Algorithm 1. At each iteration, the adversary determines which groups or instructions currently require greater optimization pressure, and the LLM policy is updated using the resulting weighted gradient. The adversary is then updated using the performance of the new LLM policy, allowing the weighting distribution to evolve throughout fine-tuning rather than remaining fixed according to group frequency.

This alternating optimization distinguishes our method from static reweighting approaches such as inverse weighting. Whereas inverse weighting determines sample importance solely from group frequency before fine-tuning, the proposed adversary continuously adapts its weights according to the current group-wise performance. This enables the same optimization framework to target predefined vulnerable groups, operate without predefined group labels, or prioritize the groups experiencing the smallest gains relative to a reference policy. The method introduces only a lightweight adversary in addition to standard LLM fine-tuning, resulting in modest computational overhead.

Theoretical Analysis. Appendix G provides a theoretical analysis of this adversarial reweighting principle. In particular, we show that an idealized expressive adversary exactly recovers the worst-group objective, and that a smooth regularized alternating scheme approaches a stationary point under standard smoothness and adversary-tracking assumptions. We further show that the stability parameter b interpolates between worst-group optimization and preservation of average group performance, thereby controlling how strongly the adversary can concentrate optimization pressure.

4 EMPIRICAL EVALUATION

This section provides an empirical study of fair LLM fine-tuning from imbalanced data, with particular emphasis on worst-group performance and group-level regressions.

4.1 RECIPE GENERATION TASK

Setup. We construct a robust LLM fine-tuning benchmark for *personalized recipe generation*, where each instruction specifies a cuisine together with user-specific ingredient constraints. To model realistic distributional skew arising from heterogeneous user populations and preferences, cuisines are sampled from an imbalanced mixture consisting of Indian (50%), French (30%), Mexican (15%), and Japanese (5%). Each instruction further specifies two *must-use* ingredients and one *must-not-use* ingredient. Must-use ingredients are sampled from cuisine-specific candidate sets, with 10% probability from *rare* ingredients and 90% from *frequent* ingredients, while the must-not-use ingredient is sampled uniformly from the frequent set. The cuisine-specific ingredient spaces are detailed in Appendix C.

Each (x, a) pair in the training data is evaluated by an automatic culinary evaluator based on Qwen2.5-7B along four dimensions, *clarity*, *completeness*, *constraint adherence*, and *practicality*, each scored on a 1–5 scale. The scalar reward $r \in [4, 20]$ is the sum of these scores, yielding the training dataset $\mathcal{D} = \{(x_i, a_i, r_i)\}_{i=1}^n$. The exact evaluation prompt is provided in Appendix D.2. In a deployed system, such rewards could instead arise from user feedback such as ratings, clicks, or preference signals; here, the LLM-based evaluator serves as a controlled proxy for constructing such rewards and evaluating held-out outputs. Our method requires only scalar rewards and does not otherwise depend on how those rewards are obtained. All recipe-generation results are averaged over 10 independent random seeds. For each seed, we independently regenerate the training, validation, and test logs, each containing $10{,}000$ samples. We report mean $\pm$ standard deviation across seeds, capturing variation due to both training-data and test-set sampling.

Compared methods. We compare three different approaches. **Naive** applies the standard reward-weighted policy update. **Inverse Weighting (IW)** scales each sample by $1/p_g$, where p_g denotes the empirical frequency of its group. **Adversarial (Ours)** alternates one adversary update with one policy update as described in Algorithm 1. We evaluate group-aware variants based on *Cuisine* (**Adversarial (Cui)**) and *Ingredient Rarity* (**Adversarial (Rarity)**), together with the group-agnostic *Universal* variant (**Adversarial (Univ)**). Hyperparameter tuning, adversary architectures, and other implementation details are provided in Appendix E.

Evaluation metrics. Our primary evaluation focuses on absolute worst-group performance and relative fairness under distributional imbalance. For a grouping $\mathcal{G}$, either the four cuisine types or the two ingredient-rarity groups, we define the worst-group value as $V_{\min}(\pi) = \min_{g \in \mathcal{G}} V_g(\pi)$, where $V_g(\pi)$ is estimated on the test set using the automatic evaluator. This metric directly measures the performance of the least-served group rather than aggregate utility.

To assess the trade-off between robustness and overall utility, we also report the average performance $V_{avg}(\pi)$ estimated over the full held-out test set. Reporting V_{avg} together with $V_{\min}$ allows us to verify whether improvements in worst-group performance are obtained at the expense of substantial average-performance degradation.

We additionally evaluate performance relative to a reference policy π_{ref} through the worst-group relative improvement $\Delta V_{\min}(\pi) = \min_{g \in \mathcal{G}} \{V_g(\pi) - V_g(\pi_{\text{ref}})\}$. This quantity captures the smallest group-level gain, or largest regression, relative to the reference policy. In our experiments, π_{ref} is a fixed 50/50 mixture of the data generating LLM and the Naive fine-tuned policy. This provides an intermediate reference between the original logging behavior and standard expected-reward fine-tuning, allowing us to assess whether a method yields uniform improvements beyond this midpoint. The reference policy is fixed throughout training. A positive $\Delta V_{\min}$ means that every evaluated group improves over this reference, whereas a negative value identifies at least one regressing group.

Worst-group robustness. Table 1(a) reports the worst-group value $V_{\min}$ across cuisine types, while Figure 2 shows the corresponding per-group performance.[2] Naive fine-tuning attains a worst-group value of 11.91, with Japanese cuisine emerging as the lowest-performing group. Inverse Weighting improves this value only modestly to 12.02, suggesting that correcting group frequency alone is insufficient to address the observed performance disparity. By contrast, Adversarial (Cui) achieves the highest worst-group performance, 12.25, corresponding to improvements of 2.9% over

[2]The Appendix provides the full per-group results for both V_g and ΔV_g across cuisine and ingredient-rarity.

Table 1: Average performance V_{avg}, worst-group performance $V_{\min}$, and relative worst-group improvement $\Delta V_{\min}$ on the recipe generation task. Worst-group metrics are reported for cuisine type and ingredient rarity. Higher is better for all metrics. **Bold green** indicates the best result in each column, and underlining indicates the second best.

		(a) Cuisine groups		(b) Ingredient rarity	
Method	V_{avg}	$V_{\min}$	$\Delta V_{\min}$	$V_{\min}$	$\Delta V_{\min}$
Naive	**12.46**	11.91 ± 0.08	-0.003 ± 0.016	11.81 ± 0.10	-0.028 ± 0.020
Inverse Weighting (Cui)	12.23	12.02 ± 0.07	-0.021 ± 0.009	12.02 ± 0.09	-0.011 ± 0.024
Adversarial (Cui; Ours)	12.42	$\mathbf{12.25 \pm 0.07}$	$\mathbf{+0.013 \pm 0.007}$	11.98 ± 0.10	-0.006 ± 0.019
Adversarial (Rarity; Ours)	12.39	12.07 ± 0.06	-0.004 ± 0.010	$\mathbf{12.28 \pm 0.07}$	$\mathbf{+0.012 \pm 0.008}$
Adversarial (Univ; Ours)	12.35	12.21 ± 0.05	$+0.010 \pm 0.012$	12.21 ± 0.08	$+0.007 \pm 0.009$

Naive and 1.9% over Inverse Weighting. Adversarial (Univ) achieves a comparable second-best value of 12.21 despite not using cuisine labels, indicating that instruction-dependent adversarial weighting can improve robustness even without specifying the evaluation groups in advance.

A similar pattern emerges when groups are defined by ingredient rarity (Table 1(b)). Naive obtains a worst-group value of 11.81, driven by lower performance on recipes including rare ingredients, while Inverse Weighting provides only a limited improvement. Adversarial (Rarity) performs best with $V_{\min} = 12.28$, showing the advantage of explicitly targeting the relevant grouping when it is known. Adversarial (Univ) again achieves the second-best result, 12.21, improving over Naive by 3.4% without access to rarity labels. In contrast, Adversarial (Cui) provides less improvement under the rarity-based evaluation, consistent with its optimization objective being aligned with cuisine rather than ingredient rarity. Together, these results suggest that **group-aware adversarial weighting is most effective when the relevant groups are known, while the universal variant provides a robust alternative when the evaluation grouping is unknown or unavailable during fine-tuning.**

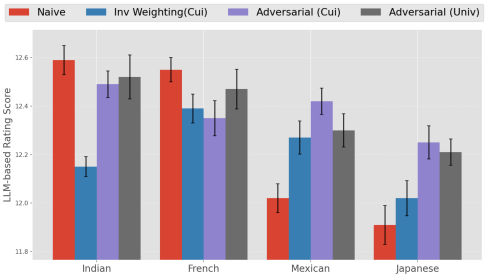


Figure 2: Recipe generation results across cuisine groups evaluated by an LLM evaluator.

Importantly, the improvements in worst-group robustness are not obtained by substantially sacrificing average utility. As shown in Table 1, Adversarial (Cui) achieves $V = 12.42$, compared with 12.46 for Naive, while improving the cuisine-based worst-group value from 11.91 to 12.25. Adversarial (Univ) similarly improves worst-group performance across both evaluated groupings while maintaining a competitive average value of 12.35, indicating a favorable trade-off between overall utility and worst-case robustness.

How do baselines fail? Qualitative inspection reveals two recurring failure modes. First, models sometimes exhibit *title spoofing*, attaching cuisine labels such as "Mexican" or "Japanese" to otherwise generic recipes without incorporating characteristic ingredients or preparation techniques, which reduces constraint adherence. Second, models can evade difficult ingredient constraints through *near-substitution*, replacing rare requested ingredients with more common alternatives (e.g., "low-fat milk" with "milk" or rare spices with common substitutes). Such substitutions may preserve fluency but violate the original instruction and are therefore penalized by the evaluator.

Relative fairness. We next evaluate the worst group-level improvement relative to the reference policy using $\Delta V_{\min}$. As shown in Table 1, Naive and Inverse Weighting exhibit negative values under both cuisine- and rarity-based evaluations. For example, Naive yields $\Delta V_{\min} = -0.003$ for cuisine and -0.028 for ingredient rarity, indicating that at least one evaluated group performs worse than under the reference policy.

The adversarial methods substantially reduce these regressions. Adversarial (Cui) achieves the largest relative improvement for cuisine groups, with $\Delta V_{\min} = +0.013$, while Adversarial (Rarity) performs best for ingredient-rarity groups, with $\Delta V_{\min} = +0.012$. Notably, Adversarial (Univ)

Table 2: Worst-group performance ($V_{\min}$) for patent summarization under LLM-based ratings and the SummaC factual consistency metric. We report $V_{\min}$ across different groupings: CPC sections and input text length (short vs. long patents). The results are mean ± standard deviation over 10 random seeds. Adversarial (Known Groups) denotes Adversarial (CPC) in panel (a) and Adversarial (Length) in panel (b). Higher values indicate better worst-group robustness. **Bold green** denotes the best worst-group performance among trained models. Human-Written Reference evaluates the reference abstracts and is included only as a reference point, not as a trained model.

	(a) CPC sections		(b) Text length (short vs. long)	
Method	LLM-based $V_{\min}$	SummaC $V_{\min}$	LLM-based $V_{\min}$	SummaC $V_{\min}$
Human-Written Reference	14.95 ± 0.61	0.681 ± 0.018	16.54 ± 0.22	0.734 ± 0.011
Naive	12.11 ± 1.10	0.591 ± 0.035	12.60 ± 0.36	0.692 ± 0.015
Inverse Weighting (IW)	12.18 ± 0.68	0.583 ± 0.029	12.45 ± 0.63	0.689 ± 0.021
Adversarial (Known Groups)	$\mathbf{12.91 \pm 0.72}$	$\mathbf{0.619 \pm 0.024}$	$\mathbf{13.19 \pm 0.42}$	$\mathbf{0.698 \pm 0.015}$
Adversarial (Universal)	12.76 ± 0.64	0.613 ± 0.026	13.05 ± 0.48	0.696 ± 0.017

achieves positive relative worst-group improvements under both groupings, with $\Delta V_{\min} = +0.010$ for cuisine and $+0.007$ for rarity, despite using neither cuisine nor rarity labels during training. Thus, consistent with the other fairness criteria, group-aware variants perform best when the relevant grouping is known in advance, while the universal variant achieves non-regression across both evaluated groupings without requiring those group definitions during optimization.

Ablation Study: Choice of b. The stability parameter b in Eq. 1 controls the minimum weight assigned to each sample and therefore the degree to which the adversary can concentrate optimization on poorly performing regions. Figure 3 evaluates its effect on worst-group performance.

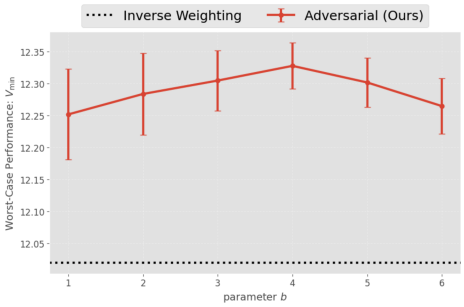


Figure 3: Ablation on the stability parameter b for recipe generation. We report the worst-group performance $V_{\min}$ for cuisine groups as b varies from 1 to 6. The horizontal dashed line denotes the Inverse Weighting baseline.

Across the tested values, adversarial reweighting consistently outperforms the Inverse Weighting baseline in terms of $V_{\min}$. The default choice $b = 1$ already captures most of the improvement, indicating that the proposed method does not require extensive tuning to obtain substantial worst-group gains. Increasing b yields modest additional improvements, with $b = 4$ achieving the highest worst-group performance among the evaluated settings. Performance then saturates and slightly decreases for larger values, suggesting that overly strong stabilization can limit the adversary's ability to focus on poorly performing groups. Overall, the results indicate that the method is robust to the choice of b, although the optimal value is task-dependent and should be selected via validation.

4.2 Content Summarization Task

We next evaluate whether the worst-group improvements observed in recipe generation generalize to a different domain and fine-tuning regime. Specifically, we consider patent summarization under standard instruction tuning, where supervision is provided by human-written target summaries.

Setup. We construct the summarization task using the BIGPATENT corpus (Sharma et al., 2019). Each example contains a long patent *Description* and a concise human-written *Abstract*; the model receives the Description x and generates an abstract-style summary a. To study distributional imbalance across technical domains, we use the standard CPC sections (A–H and Y) as groups, whose empirical frequencies are highly skewed. We also consider input-length groups to capture differences in task difficulty between shorter and longer patents. After deduplication and basic cleaning,

including removal of HTML and figure/table references, we retain examples whose Description contains between 200 and 10,000 words and whose Abstract contains between 100 and 200 words.

Following the instruction-tuning formulation in Section 2, the training data take the form $\mathcal{D} = \{(x_i, a_i, r_i)\}$, where a_i is the corresponding human-written Abstract and $r_i = 1$ for every example. We sample 10,000 examples each from the original training and validation splits using CPC-stratified sampling, while retaining the full original test set of 67,072 examples.

We fine-tune Qwen2.5-7B to generate patent-style abstracts from the Description. The generation instruction specifies a length of 100–200 words, an objective technical tone, and excludes claims-like phrasing, citations, and bullet lists. We compare **Naive**, **Inverse Weighting**, our group-aware **Adversarial** method, and the group-agnostic **Adversarial (Universal)** variant. For CPC-based evaluation, **Adversarial (CPC)** treats CPC sections as the protected groups; for length-based evaluation, **Adversarial (Length)** distinguishes short ($\leq$ 2,000 words) and long ($>$ 2,000 words) inputs. Finally, Human-Written Reference denotes evaluation of the reference abstracts themselves and is included as a reference point rather than a trained model.

Generated summaries on test data are evaluated using two complementary measures. First, an LLM-based evaluator scores each output along four dimensions, *Clarity*, *Constraint Adherence*, *Similarity to the Human Summary*, and *Factual Consistency*, each on a 1–5 scale, yielding an overall score in $[4, 20]$. Second, we report **SummaC** (Laban et al., 2022), an NLI-based factual consistency metric that measures whether statements in the generated summary are supported by the source Description. We evaluate worst-group performance under both CPC-section and input-length groupings.

Results. Table 2 reports worst-group performance $V_{\min}$ under the LLM-based evaluator and SummaC. Panel (a) evaluates robustness across CPC sections, while panel (b) evaluates robustness across short and long inputs. Adversarial (Known Groups) denotes the group-aware variant, using CPC sections as groups in panel (a) and input-length categories in panel (b). Note that we report the corresponding average performance V_{avg} in Appendix F.

Robustness across CPC sections. As shown in Table 2(a), Naive achieves a worst-group LLM-based score of 12.11, while IW provides only a modest increase to 12.18. Adversarial (CPC) improves the worst-group score to $\mathbf{12.91}$, a 6.6% improvement over Naive. Adversarial (CPC) also achieves the highest worst-group SummaC score of 0.619, compared with 0.591 for Naive and 0.583 for Inverse Weighting. These results indicate that explicitly adapting the training weights to current group performance improves robustness across imbalanced technical domains beyond what can be achieved by frequency-based reweighting alone.

Robustness across input length. Table 2(b) evaluates groups defined by input length. Naive and IW obtain worst-group LLM-based scores of 12.60 and 12.45, respectively, with long patents forming the more challenging group. Adversarial (Length) increases the worst-group score to $\mathbf{13.19}$. A consistent improvement is observed under SummaC, where Adversarial achieves the highest worst-group factual consistency of $\mathbf{0.698}$. Thus, the gains from adversarial reweighting extend beyond group-frequency imbalance and also appear when groups differ primarily in task difficulty.

5 Conclusion and Future Work

We studied fairness in LLM fine-tuning from imbalanced data, where optimizing average reward can lead to substantial regressions for underrepresented or difficult prompt groups. We proposed an adversarial fine-tuning framework that adaptively reweights training samples to directly improve worst-group performance and accommodates group-aware, group-agnostic, and relative fairness objectives within a unified procedure. Across recipe generation and content summarization, our method consistently improves worst-group performance over the baselines and the group-agnostic variant provides robustness without requiring predefined group labels.

Several directions remain for future work. An important next step is to extend the framework to richer, potentially latent or dynamically changing notions of groups and to study its behavior with noisier and more heterogeneous real-world feedback. It would also be valuable to evaluate the method in online deployments, where long-term user outcomes and distribution shifts can provide a more comprehensive assessment of the benefits and trade-offs of worst-group-aware fine-tuning.

AI USE STATEMENT

Generative AI tools were used for language editing, wording refinement, and title brainstorming. All AI-assisted edits were reviewed and verified by the authors, who take full responsibility for the final content.

REFERENCES

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

Paul F. Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9268–9277, 2019.

Saswat Das, Marco Romanelli, Cuong Tran, Zarreen Reza, Bhavya Kailkhura, and Ferdinando Fioretto. Low-rank finetuning for llms: A fairness perspective. *arXiv preprint arXiv:2405.18572*, 2024.

Thang Viet Doan, Zhibo Chu, Zichong Wang, and Wenbin Zhang. Fairness definitions in language models explained. *arXiv preprint arXiv:2407.18454*, 2024.

Charles Elkan. The foundations of cost-sensitive learning. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, pp. 973–978, 2001.

Dennis Frauen, Valentyn Melnychuk, and Stefan Feuerriegel. Fair off-policy learning from observational data. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 13943–13972, 2024.

Tatsunori B Hashimoto, Megha Srivastava, Hongseok Namkoong, and Percy Liang. Fairness without demographics in repeated loss minimization. In *International Conference on Machine Learning (ICML)*, pp. 1929–1938, 2018.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Precious Jones, Weisi Liu, I-Chan Huang, and Xiaolei Huang. Examining imbalance effects on performance and demographic fairness of clinical language models. In *2025 IEEE 13th International Conference on Healthcare Informatics (ICHI)*, pp. 58–68. IEEE, 2025.

Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2564–2572. PMLR, 2018.

Haruka Kiyohara, Daniel Yiming Cao, Yuta Saito, and Thorsten Joachims. An off-policy learning approach for steering sentence generation towards personalization. In *Proceedings of the Nineteenth ACM Conference on Recommender Systems*, pp. 41–50, 2025.

Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. SummaC: Re-visiting NLI-based models for inconsistency detection in summarization. *Transactions of the Association for Computational Linguistics*, 10:163–177, 2022. doi: 10.1162/tacl_a_00453.

Preethi Lahoti, Alex Beutel, Jilin Chen, Kang Wook Lee, Flavien Prost, Nithum Thain, Xuezhi Wang, and Ed Chi. Fairness without demographics through adversarially reweighted learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 728–740, 2020.

Jack Lanchantin, Angelica Chen, Janice Lan, Xian Li, Swarnadeep Saha, Tianlu Wang, Jing Xu, Ping Yu, Weizhe Yuan, Jason E Weston, Sainbayar Sukhbaatar, and Ilia Kulikov. Bridging offline and online reinforcement learning for llms. *arXiv preprint arXiv:2506.21495*, 2025.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*, 2022.

Bodhisattwa Prasad Majumder, Shuyang Li, Jianmo Ni, and Julian McAuley. Generating personalized recipes from historical user preferences. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 5976–5982, 2019.

Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Computing Surveys*, 54(6):1–35, 2021.

Marco Morik, Ashudeep Singh, Jessica Hong, and Thorsten Joachims. Controlling fairness and bias in dynamic learning-to-rank. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, pp. 429–438, 2020.

Yurii Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103 (1):127–152, 2005. doi: 10.1007/s10107-004-0552-5.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*, 2022.

Doina Precup, Richard S. Sutton, and Satinder Singh. Eligibility traces for off-policy policy evaluation. In *Proceedings of the 17th International Conference on Machine Learning (ICML)*, pp. 759–766, 2000.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In *International Conference on Learning Representations (ICLR)*, 2020.

Eva Sharma, Chen Li, and Lu Wang. Bigpatent: A large-scale dataset for abstractive and coherent summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 2204–2213. Association for Computational Linguistics, 2019.

Ashudeep Singh and Thorsten Joachims. Policy learning for fairness in ranking. *Advances in neural information processing systems*, 32, 2019.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems*, volume 33, pp. 3008–3021, 2020.

Adith Swaminathan and Thorsten Joachims. Counterfactual risk minimization: Learning from logged bandit feedback. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pp. 814–823, 2015.

Yifan Wang, Weizhi Ma, Min Zhang, Yiqun Liu, and Shaoping Ma. A survey on the fairness of recommender systems. *ACM Transactions on Information Systems*, 41(3):1–43, 2023.

Xiao Xiao, Jiaxu Zhao, Terry R Payne, and Meng Fang. Empirical study of social bias in medical question answering via large language models. In *International Conference on AI in Healthcare*, pp. 3–16. Springer, 2025.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

Yanghao Zhang, Tianle Zhang, Ronghui Mu, Xiaowei Huang, and Wenjie Ruan. Towards fairness-aware adversarial learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24746–24755, 2024.

Yue Zhou, Barbara Di Eugenio, and Lu Cheng. Unveiling performance challenges of large language models in low-resource healthcare: A demographic fairness perspective. pp. 7266–7278, 2025.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

A Extended Related Work

We review four strands of research, namely LLM fine-tuning, fairness in LLMs, performance disparity beyond LLMs, and fairness-aware learning. We then position our adversarial offline policy learning method as advancing *worst-group* guarantees of LLM fine-tuning under logged bandit feedback.

LLM Fine-tuning. Supervised instruction tuning and RL from human preferences (RLHF) are dominant paradigms for aligning LLMs with task- or preference-specific objectives, spanning supervised fine-tuning on demonstrations, reward modeling, and policy optimization with bandit-style feedback (Ouyang et al., 2022; Christiano et al., 2017; Stiennon et al., 2020; Ziegler et al., 2019; Lanchantin et al., 2025; Rafailov et al., 2023). While these approaches improve average utility, most assume i.i.d. data or implicitly uniform coverage over instruction groups and commonly optimize the expected reward. Our work differs by formulating LLM fine-tuning explicitly as *offline* policy optimization under group imbalance and varying difficulty. Specifically, we instantiate adversarial reweighting that directly targets worst-group performance and uniform relative improvements to a given reference policy, providing fairness-aware guarantees unavailable in standard expected-reward fine-tuning.

Fairness in LLMs: definitions and scope. Fairness for LLMs has been actively studied, but *what* is meant by fairness varies. Prior work has emphasized (a) harmful content and toxicity mitigation, (b) representational or stereotyping harms, (c) demographic parity or calibration across protected attributes, and (d) safety/abuse constraints at decoding time (Doan et al., 2024; Liang et al., 2022). In contrast, we focus on *performance-disparity* fairness: reducing gaps in task performance across groups and improving the *worst-group* outcome under data imbalance and variable difficulty. Performance disparities in LLMs have been empirically documented across demographic attributes and clinical/medical QA settings, e.g., accuracy gaps by gender and race under prompt perturbations (Xiao et al., 2025), widened minority–majority gaps under low-rank fine-tuning with imbalanced data (Das et al., 2024), and performance differences across demographic slices in low-resource healthcare benchmarks (Zhou et al., 2025; Jones et al., 2025). While across-group performance disparities have been observed in prior work, **practical fine-tuning algorithms** that address this issue remain underexplored. We target precisely this gap by instantiating adversarial reweighting for offline LLM fine-tuning using logged reward signals.

Performance disparity beyond LLMs. Outside LLMs, *performance disparity* and related exposure/coverage disparities are well established topics in recommender systems and information retrieval, including user- or item-group fairness, popularity bias, and worst-case utility guarantees (Singh & Joachims, 2019; Morik et al., 2020; Wang et al., 2023). This literature motivates our choice to operationalize fairness as *worst-group robustness* and *relative improvements* under distributional skew. We adapt these ideas to the LLM fine-tuning regime where supervision arrives as logged bandit feedback and group labels may be unavailable.

Fairness-aware Learning. Fairness-aware learning has drawn on distributionally robust optimization and worst-group (max–min) training to protect minority groups (Hashimoto et al., 2018; Sagawa et al., 2020; Frauen et al., 2024). Adversarial reweighting methods further adapt sample or group importance to close disparities without relying on perfectly specified group labels (Lahoti et al., 2020; Zhang et al., 2024). However, most prior work assumes supervised full-information losses; comparatively less addresses offline bandit settings central to LLM fine-tuning, where only logged actions and rewards are observed and counterfactuals are unavailable. Our key contribution is to apply adversarial reweighting to offline policy optimization for LLM fine-tuning. Moreover, beyond group-aware and group-agnostic fairness objectives studied previously outside the LLM fine-tuning literature, we introduce a relative-fairness objective and realize all three within a single alternating optimization scheme.

B The Generality of Our Method

While our primary focus is on maximizing the worst-group performance, the same adversarial framework can be instantiated to minimize performance gaps across groups when this is the de-

sired objective. Specifically, recall that our adversarial formulation in Eq. (2) defines a min–max optimization between the policy and an adversary. To minimize the performance gap, we instead define the adversary to maximize

$$\max_{w^+,w^-} \sum_{g\in\mathcal{G}} w^+(g)\, V_g(\pi_\theta) \;-\; \sum_{g\in\mathcal{G}} w^-(g)\, V_g(\pi_\theta),$$

where $w^+(g)$ and $w^-(g)$ are parameterized and normalized weighting functions over groups (e.g., neural networks with softmax outputs), satisfying $w^+(g), w^-(g) \geq 0$ and $\sum_g w^+(g) = \sum_g w^-(g) = 1$.

This objective is maximized when w^+ concentrates on the best-performing group and w^- on the worst-performing group, yielding $\max_{g\in\mathcal{G}} V_g(\pi_\theta) - \min_{g\in\mathcal{G}} V_g(\pi_\theta)$. The policy then minimizes this adversarial objective,

$$\min_{\pi_\theta} \max_{w^+,w^-} \left(\sum_g w^+(g) V_g(\pi_\theta) - \sum_g w^-(g) V_g(\pi_\theta) \right),$$

thereby directly optimizing gap-based fairness using the same alternating adversarial updates.

C Dataset Construction Details

C.1 Recipe Generation

Instruction schema. An instruction x specifies a cuisine label $g_{\text{cuisine}} \in \{\text{Indian}, \text{French}, \text{Mexican}, \text{Japanese}\}$ and ingredient constraints: two *must-use* items and one *must-not-use* item. Cuisines are sampled with an imbalanced mixture (Indian 50%, French 30%, Mexican 15%, Japanese 5%). The must-use set is drawn from cuisine-specific vocabularies with probability 0.9 from *frequent* items and 0.1 from *rare* items; the must-not-use item is sampled uniformly from frequent items.

Ingredient vocabularies. For each cuisine, we maintain a disjoint candidate set with frequency tags (frequent/rare) derived from public recipe corpora (Majumder et al., 2019), and we curate the top 50 frequent and rare ingredients (representative examples are listed below).

- **Indian** (frequent): cumin, coriander, turmeric, garam masala, ghee, ginger, garlic, green chile, tomato, onion. (rare): asafoetida, dried fenugreek leaf, amchur, black cardamom.
- **French** (frequent): butter, cream, thyme, rosemary, shallot, white wine, Dijon mustard. (rare): tarragon vinegar, sauce velouté, fleur de sel.
- **Mexican** (frequent): cumin, oregano, lime, cilantro, jalapeño, tomato, onion, corn tortilla. (rare): hoja santa, pasilla negro, huitlacoche.
- **Japanese** (frequent): soy sauce, mirin, sake, miso, kombu, katsuobushi, daikon. (rare): yuzu kosho, shiso seeds, kinome.

Logged data. A logging policy π_0 produces outputs a for each x by mixing two LLMs (80% GPT-2, 20% Qwen2.5-7B), both of which are pre-trained for the recipe domain using the public **Food.com Recipe & Review Data** (Majumder et al., 2019). Each (x, a) is scored by an automatic evaluator on four axes (1–5): *clarity*, *completeness*, *constraint adherence*, *practicality*; the scalar reward is $r \in [4, 20]$ as the axis-sum. Duplicates and low-quality artifacts (empty or overlong outputs) are removed.

Splits. We stratify by cuisine and rarity to create train/validation/test splits. Unless otherwise stated, training uses all available logged triples, validation is held out for hyperparameter tuning, and test is used for final reporting with 10 random seeds.

C.2 CONTENT SUMMARIZATION (BIGPATENT)

Preprocessing and filtering. We use the BIGPATENT corpus (Sharma et al., 2019) and retain examples where the *Description* length is 200–10,000 words and the *Abstract* length is 100–200 words. Basic cleaning removes HTML/boilerplate and references to figures/tables. CPC sections (A–H, Y) define primary groups and introduce substantial imbalance.

Instruction-tuning setup. We follow the original dataset splits, i.e., train = 1,207,222, validation = 67,068, and test = 67,072 examples. We keep the *test* set unchanged, and from the original train and validation splits we draw 10,000 samples each via CPC-stratified sampling. For instruction tuning, we construct $\mathcal{D}_{\text{IT}} = \{(x_i, a_i, r_i{=}1)\}$ by pairing each *Description* x_i with its human-written *Abstract* a_i (no logging policy).

D PROMPTS AND EVALUATORS

This section describes the prompts we used to generate recipes/summaries in each of the experiments and those we used for evaluation.

D.1 RECIPE GENERATOR PROMPT

```
You are an expert recipe creator.

Constraints:

- Must use: {must\_use\_str}
- Must not use: {must\_not\_use\_str}
- Cuisine: {cuisine}

Produce a single recipe with:

- Name
- Description
- Cooking Time
- Ingredients
- Steps
```

D.2 RECIPE EVALUATOR PROMPT

```
You are an expert culinary evaluator.
Your task is to assess a recipe generated by a language model based
    on the following inputs:

1. The constraints:
   - Must use: {must\_use\_str}
   - Must not use: {must\_not\_use\_str}
   - Cuisine: {cuisine}
2. The generated recipe: {recipe}

Please evaluate the recipe independently on the following four
    criteria, using a scale of 1 to 5
(where 1 = very poor and 5 = excellent). Justify each score briefly.

- Clarity: Are the instructions easy to understand, logically
    ordered, and free from ambiguity?
- Completeness: Does the recipe include all essential steps,
    ingredients,
```

```
  cooking times, and methods needed to reproduce the dish from start
    to finish?
- Constraint Adherence: Does the recipe genuinely reflect the
    specified cuisine
  (Japanese, French, Indian, or Mexican), beyond simply labeling
    itself as such?
  Does it faithfully follow the "must use" and "must not use"
    ingredient constraints?
  Penalize recipes that use superficial markers of the cuisine (e.g.,
    naming a dish
  "Mexican" without using actual Mexican culinary elements).
- Practicality: Can an average home cook realistically follow and
  execute this recipe using commonly available tools and ingredients?
  Consider the overall feasibility of preparation and clarity of
    measurements and instructions.

Note: Score higher when the recipe is well-structured, authentic, and
    feasible,
and lower when it appears broken, formulaic, or inconsistent with the
    specified cuisine or constraints.
```

D.3 Summarization Generator Prompt

```
You are a patent summarization assistant.
Your task is to summarize the following patent "Description" into a
    concise Abstract that could appear in a USPTO filing.

Requirements:

- Target length: from 100 to 200 words.
- Focus on the inventions core novelty, function, and key components.
- Use formal technical language; avoid claims-like phrasing (what is
    claimed) and references to figures/tables.
- Avoid speculative or fabricated details not grounded in the
    description.
- No bullet points; write from 1 to 3 coherent sentences.

Description: {x}
```

D.4 Summarization Evaluator Prompt

```
You are an expert patent abstract evaluator.

Your task is to rate the generated abstract using the patent
    Description
as the source of truth and the Human Reference Abstract as a target
    style/coverage guide.

Inputs: 1) Description (source of truth): {DESCRIPTION\_TEXT}
2) Generated Abstract (candidate): {CANDIDATE\_ABSTRACT}
3) Human Reference Abstract: {HUMAN\_ABSTRACT}

Criteria (rate each from 1 to 5 and give a brief justification):

- Clarity: Clear, grammatical, coherent technical writing suitable
    for a USPTO-style abstract.
- Constraint Adherence: Obeys length/style constraints; avoids
    figures/tables/claims-like phrasing; no speculation.
```

```
- Factual Consistency: All statements are supported by the
    Description (no hallucinations or contradictions).
- Similarity-to-Human summary: Semantic alignment to the Human
    Reference Abstracts meaning units (not verbatim overlap).
  - Emphasize coverage of the same key facts and relationships.
  - Do not reward copying text from the reference; reward faithful
    paraphrase capturing the same meaning.
  - If the Human Reference contains information not grounded in the
    Description,
    prioritize Factual Consistency over similarity.
```

D.5 SUMMAC CONFIGURATION

We compute SummaC (Laban et al., 2022) convergent variants between (DESCRIPTION, GENERATED ABSTRACT). Sentence-level entailment scores are pooled with contradiction penalties; the resulting 0–1 score is reported per CPC section.

D.6 SUMMARIZATION EVALUATION PROTOCOL

We evaluate each generated summary using an LLM-based evaluator and SummaC (Laban et al., 2022). The LLM-based evaluator assigns a score from 1 to 5 along four dimensions: *Clarity*, *Constraint Adherence*, *Similarity to the Human Summary*, and *Factual Consistency*. The four scores are summed to obtain an overall score in $[4, 20]$. The complete evaluator prompt is provided in Appendix D.4. SummaC measures factual consistency between each generated summary and its source patent Description and yields a score in $[0, 1]$. For both metrics, we report the worst-group value $V_{\min}$, computed separately over CPC-section and input-length groupings.

E TRAINING AND IMPLEMENTATION DETAILS

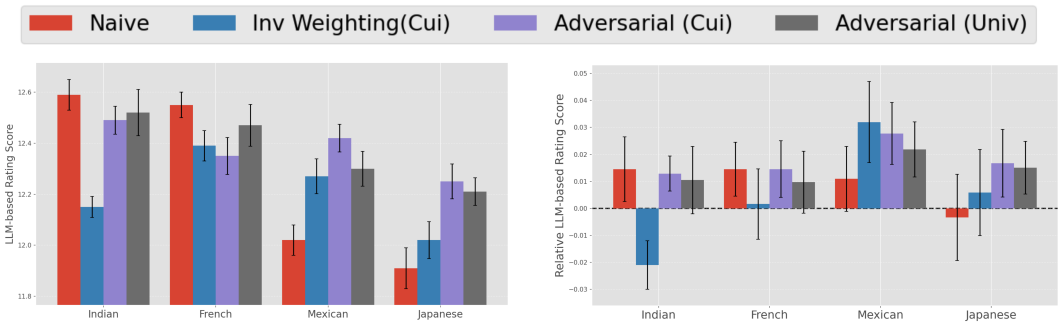
Base models and PEFT. All methods fine-tune the same **Qwen2.5-7B** base model (Yang et al., 2025) so that performance differences do not arise from differences in backbone model capacity. We adopt LoRA (Hu et al., 2022) for parameter-efficient fine-tuning with rank $= 16$, $\alpha = 32$, and targets $= \{q, k, v, o\}$. Training is conducted in bfloat16 precision with gradient clipping $= 1.0$ and a maximum sequence length of 1024. For the summarization task, we use the same backbone and configuration.

Optimization. Optimization uses **AdamW** (β_1=0.9, β_2=0.95, weight decay $= 0.1$) with a cosine schedule, 5% warm-up, and peak learning rates of 1×10^{-5} for the policy and 3×10^{-4} for the adversary, when applicable. The policy learning rate, LoRA rank, and weight decay reported above are selected using the validation-set grid search described below. Training runs for 3 epochs on 8 GPUs with a per-GPU micro-batch size of 8 sequences and gradient accumulation over 8 steps, yielding an effective global batch size of 512 sequences. The checkpoint achieving the *highest worst-group validation reward* is selected. Evaluation generation uses temperature $= 0.7$, top-p $= 0.9$, and a maximum output length of 512 tokens.

Method-specific differences are as follows.

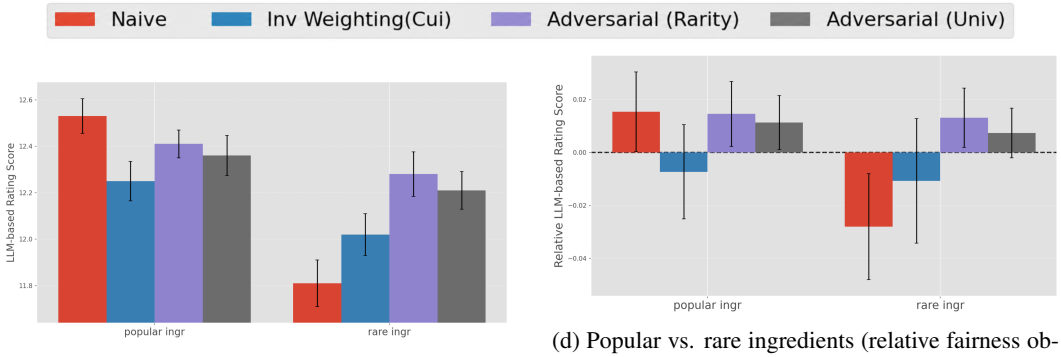
- **Naive** uses the standard reward-weighted policy gradient.
- **Inverse Weighting** multiplies per-sample gradients by $1/p_g$, where p_g is the empirical group frequency.
- **Adversarial** alternates one adversary step per policy step.

The adversary is an MLP that takes a pooled instruction representation and outputs unnormalized weights, regularized with ℓ_2 (weight penalty) and entropy (0.01) terms to prevent collapse. The adversary learning rate is 3×10^{-4}, and the stabilization floor b is selected from the range described below (default b=1).



(a) Cuisine-wise.

(b) Cuisine-wise (relative fairness objective).



(c) Popular vs. rare ingredients.

(d) Popular vs. rare ingredients (relative fairness objective).

Figure 4: Recipe generation experiment results. Each panel reports per-group absolute or relative performance measured by an LLM evaluator for the compared methods. Note that "Inv Weighting" stands for Inverse Weighting.

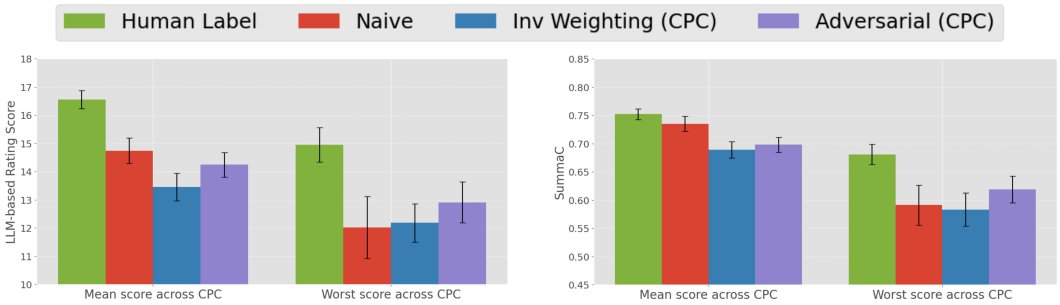
Hyperparameter search. We perform a small grid search over the learning rate $\{5 \times 10^{-6}, 1 \times 10^{-5}, 2 \times 10^{-5}\}$, LoRA rank $\{16, 32\}$, and weight decay $\{0.01, 0.1\}$. Regarding the stabilization parameter b in our adversarial method, we use $b = 1$ as the default setting, as it demonstrates stable performance across a wide range of experimental conditions. To further examine its impact, we conduct an ablation study that empirically evaluates how varying b affects worst-case performance in Figure 3.

Adversary architecture and regularization. The adversary f_ϕ is a 3-layer MLP (hidden size $= 256$, GELU activation, LayerNorm). The group-aware version uses learned embeddings per group as input, whereas the group-agnostic version uses the policy encoder's representation $h(x_i)$. We apply ℓ_2 regularization (10^{-4}) and entropy regularization (0.01) on normalized weights to prevent collapse.

Reference policy. For relative-fairness experiments, the reference π_{ref} is a 50/50 mixture of the logging policy and a naive fine-tuned policy, fixed throughout training. Relative deltas are computed on validation/test.

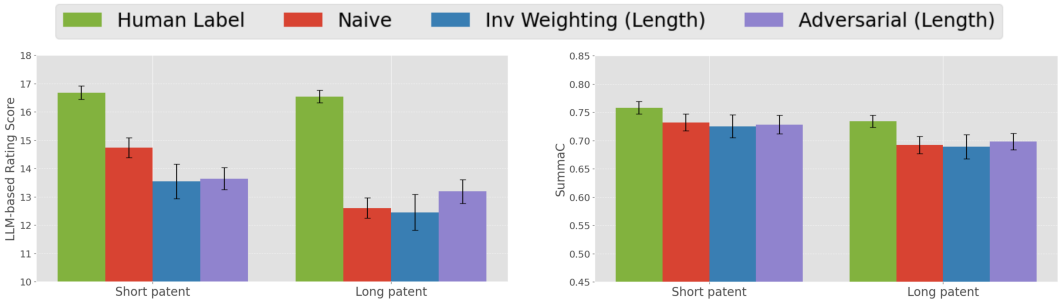
F Additional Experimental Results

This appendix provides full per-group breakdowns that complement the worst-group summaries reported in the main text.



(a) CPC group-wise, LLM evaluation.

(b) CPC group-wise, SummaC evaluation.



(c) Long vs. Short content, LLM evaluation.

(d) Long vs. Short content, SummaC evaluation.

Figure 5: Content summarization experiment results. Panels (a-b) report the mean and worst-group performance across CPC sections, whereas panels (c-d) report performance for short and long input groups. Panels (a, c) use LLM-based evaluation, and panels (b, d) use SummaC. Note that "Inv Weighting" denotes Inverse Weighting.

Table 3: Average performance (V_{avg}) for patent summarization under LLM-based ratings and the SummaC factual consistency metric. Higher values indicate better overall performance.

	(a) CPC sections		**(b) Text length**	
Method	LLM-based V_{avg}	SummaC V_{avg}	LLM-based V_{avg}	SummaC V_{avg}
Human-Written Reference	16.60 ± 0.32	0.755 ± 0.012	16.60 ± 0.32	0.755 ± 0.012
Naive	14.75 ± 0.48	0.738 ± 0.018	14.75 ± 0.48	0.738 ± 0.018
Inverse Weighting (IW)	13.50 ± 0.45	0.690 ± 0.020	13.10 ± 0.47	0.705 ± 0.020
Adversarial (Known Groups)	14.20 ± 0.46	0.700 ± 0.018	13.45 ± 0.45	0.715 ± 0.018
Adversarial (Universal)	14.05 ± 0.44	0.705 ± 0.019	14.05 ± 0.44	0.705 ± 0.019

F.1 Recipe Generation: Full Per-Group Results

Per-cuisine breakdown. Figure 4a reports the per-cuisine performance without a reference policy. While the main text focuses on worst-group statistics ($V_{\min}$ and $\Delta V_{\min}$), this figure shows that Naive fine-tuning disproportionately benefits high-frequency cuisines (Indian and French) while substantially degrading low-frequency and difficult cuisines (Mexican and Japanese). In contrast, Adversarial (Cui) and Adversarial (Univ) consistently lift the bottom cuisines without sacrificing overall quality.

Ingredient rarity breakdown. Figure 4c reports the performance for popular vs. rare ingredients without a reference policy. Naive and Inverse Weighting perform substantially worse on rare ingredients, whereas Adversarial (Rarity) and Adversarial (Univ) substantially improve the rare-ingredient group, consistent with the improvements in $V_{\min}$ reported in Table 1(b).

Relative fairness with a reference policy. Figures 4b and 4d report the relative fairness results with respect to the reference policy π_{ref}, defined as a 50/50 mixture of the logging policy and the Naive fine-tuned policy. We plot ΔV_g for each group, showing that Naive and Inverse Weighting incur regressions for at least one group, while the corresponding group-aware and universal adversarial variants achieve positive mean gains across all displayed groups. These per-group plots provide the detailed breakdown underlying the $\Delta V_{\min}$ values summarized in Table 1.

G Theoretical Analysis of Adversarial Reweighting

This section provides an idealized analysis of the optimization principle underlying our adversarial reweighting method. The goal is twofold: first, to show that an expressive adversary exactly recovers the worst-group objective; and second, to characterize the convergence of a regularized alternating optimization scheme under standard smoothness assumptions. To isolate these properties from function approximation and offline reward estimation, we analyze population group values $V_g(\pi_\theta)$. The practical algorithm replaces these quantities with logged-data surrogates and parameterizes the adversary using f_ϕ.

G.1 Adversarial Reweighting as Worst-Group Optimization

For notational simplicity, let $V_g(\theta) := V_g(\pi_\theta)$ and let $G = |\mathcal{G}|$. Consider an idealized adversary that chooses a probability vector $q \in \Delta_G$, where

$$\Delta_G = \left\{ q \in \mathbb{R}_+^G : \sum_{g=1}^{G} q_g = 1 \right\}.$$

Define the adversarially weighted value as

$$\mathcal{L}(\theta, q) = \sum_{g=1}^{G} q_g V_g(\theta).$$

Proposition 1 (Equivalence to worst-group optimization). For any fixed policy parameter θ,

$$\min_{q \in \Delta_G} \mathcal{L}(\theta, q) = \min_{g \in \mathcal{G}} V_g(\theta).$$

Consequently,

$$\max_{\theta} \min_{q \in \Delta_G} \mathcal{L}(\theta, q) = \max_{\theta} \min_{g \in \mathcal{G}} V_g(\theta).$$

Proof. Because $\mathcal{L}(\theta, q)$ is linear in q, its minimum over the probability simplex is attained at an extreme point. Let $g^\star \in \arg\min_g V_g(\theta)$. Choosing $q_{g^\star} = 1$ and $q_g = 0$ for $g \neq g^\star$ yields

$$\mathcal{L}(\theta, q) = V_{g^\star}(\theta) = \min_g V_g(\theta).$$

Conversely, every convex combination of $\{V_g(\theta)\}_{g=1}^G$ is at least their minimum, establishing the equality. □

Proposition 1 formalizes the intuition behind adversarial reweighting; an ideal adversary places all of its mass on the currently worst-performing group, and maximizing against this adversary is therefore exactly equivalent to maximizing worst-group performance. The identity of the worst-performing group may change throughout optimization, motivating adaptive rather than static reweighting.

The same argument applies directly to relative fairness. Defining

$$\Delta V_g(\theta) = V_g(\pi_\theta) - V_g(\pi_{\text{ref}}),$$

we obtain

$$\min_{q \in \Delta_G} \sum_{g=1}^{G} q_g \Delta V_g(\theta) = \min_{g \in \mathcal{G}} \Delta V_g(\theta).$$

Thus, an idealized adversary applied to relative improvements directly targets the group experiencing the smallest gain, or largest regression, relative to the reference policy.

Interpretation of the stability floor. Our practical parameterization prevents the adversary from assigning zero weight by introducing the stability parameter $b > 0$. To illustrate its role, consider an idealized group-balanced setting and define the normalized group weights

$$\widetilde{w}_g(q) = \frac{b + Gq_g}{b+1}.$$

Letting $q_g = f_\phi(g)/\sum_{g'} f_\phi(g')$, the practical group-aware weight can be written as $w_\phi(g) = b + Gq_g$. The normalized weight $\widetilde{w}_g(q) = w_\phi(g)/(b+1)$ differs only by a common positive scaling factor, chosen so that the average group weight equals one.

The average group weight is then one, and the corresponding weighted objective satisfies

$$\frac{1}{G}\sum_{g=1}^{G} \widetilde{w}_g(q) V_g(\theta) = \frac{b}{b+1} V_{\text{avg}}^{\text{grp}}(\theta) + \frac{1}{b+1}\sum_{g=1}^{G} q_g V_g(\theta),$$

where $V_{\text{avg}}^{\text{grp}}(\theta) = G^{-1}\sum_g V_g(\theta)$. Minimizing over q therefore gives

$$\min_{q\in\Delta_G} \frac{1}{G}\sum_{g=1}^{G} \widetilde{w}_g(q) V_g(\theta) = \frac{b}{b+1} V_{\text{avg}}^{\text{grp}}(\theta) + \frac{1}{b+1} V_{\min}(\theta).$$

Hence, in this idealized interpretation, the stability floor interpolates between worst-group optimization and preservation of average group performance. As $b \to 0$, the objective approaches the exact max–min objective in Proposition 1, whereas larger b prevents the adversary from concentrating all optimization pressure on a single group. This interpretation is also consistent with the empirical role of b as a stability parameter.

G.2 A Smooth Regularized Adversarial Objective

The exact worst-group objective is generally non-smooth when multiple groups attain the minimum simultaneously. To obtain a smooth optimization problem, consider the regularized adversarial objective

$$\mathcal{L}_\tau(\theta, q) = \sum_{g=1}^{G} q_g V_g(\theta) + \tau R(q), \qquad F_\tau(\theta) = \min_{q\in\Delta_G} \mathcal{L}_\tau(\theta, q),$$

where $\tau > 0$ and R is a strongly convex regularizer. This abstraction captures the role of adversary regularization in preventing degenerate weight concentration. For example, with the negative-entropy regularizer $R(q) = \sum_g q_g \log q_g$, the inner problem has the unique solution

$$q_g^\star(\theta) = \frac{\exp(-V_g(\theta)/\tau)}{\sum_{h=1}^{G}\exp(-V_h(\theta)/\tau)}.$$

Thus, groups with lower current performance receive larger adversarial weights. The resulting reduced objective is the standard soft minimum

$$F_\tau(\theta) = -\tau \log \sum_{g=1}^{G} \exp\left(-V_g(\theta)/\tau\right),$$

which satisfies the standard log-sum-exp bound (Boyd & Vandenberghe, 2004; Nesterov, 2005),

$$V_{\min}(\theta) - \tau \log G \le F_\tau(\theta) \le V_{\min}(\theta).$$

The bound follows directly from

$$\exp\left(-V_{\min}(\theta)/\tau\right) \le \sum_{g=1}^{G} \exp\left(-V_g(\theta)/\tau\right) \le G \exp\left(-V_{\min}(\theta)/\tau\right).$$

Hence, the negative-entropy regularization replaces the hard minimum with a smooth approximation whose error is at most $\tau \log G$; therefore, by choosing a sufficiently small τ, the resulting objective can be made arbitrarily close to the original worst-group objective while remaining smooth.

G.3 Stationarity of Inexact Alternating Updates

We next give a simple stationarity result for alternating policy and adversary updates. In practice, the adversary is optimized only approximately at each iteration, so the resulting adversarial weight q_t need not coincide with the exact inner optimum $q^\star(\theta_t)$. The central question is therefore whether updating the policy using $\nabla_\theta \mathcal{L}_\tau(\theta_t, q_t)$ still allows the iterates to approach a stationary point of the reduced objective F_τ.

Assume here that: (i) each $V_g(\theta)$ is continuously differentiable with bounded gradients; (ii) the regularized inner problem is strongly convex in q, so that $q^\star(\theta) = \arg\min_{q \in \Delta_G} \mathcal{L}_\tau(\theta, q)$ is unique; (iii) the reduced objective F_τ is L_F-smooth and upper bounded by $F_\tau^{\max}$; and (iv) there exists $C < \infty$ such that

$$\|\nabla_\theta \mathcal{L}_\tau(\theta, q) - \nabla_\theta \mathcal{L}_\tau(\theta, q')\| \le C\|q - q'\|$$

for all feasible (θ, q, q'). Assumption (i) ensures regular behavior of the group values and their policy gradients, while (ii) guarantees that the optimal adversary is well defined and unique. Assumption (iii) provides the smoothness needed to analyze gradient-based policy updates and an upper bound on the achievable objective value. Assumption (iv) states that a small perturbation in the adversarial weights induces only a proportionally small change in the policy gradient.

At iteration t, suppose that the adversary update returns q_t satisfying

$$\|q_t - q^\star(\theta_t)\| \le \epsilon_t,$$

after which the policy is updated as

$$\theta_{t+1} = \theta_t + \eta \nabla_\theta \mathcal{L}_\tau(\theta_t, q_t).$$

Theorem 1. (Stationarity under approximate adversary updates) Under the assumptions above, if $0 < \eta \le 1/(2L_F)$, then after T policy updates,

$$\frac{1}{T}\sum_{t=0}^{T-1} \|\nabla F_\tau(\theta_t)\|^2 \le \underbrace{\frac{2(F_\tau^{\max} - F_\tau(\theta_0))}{\eta T}}_{\text{policy optimization error}} + \underbrace{\frac{C^2}{T}\sum_{t=0}^{T-1} \epsilon_t^2}_{\text{adversary tracking error}}.$$

In particular, if the adversary is solved exactly at every iteration, $\epsilon_t = 0$, the average squared gradient norm decreases at rate $O(1/T)$. More generally, if $\frac{1}{T}\sum_{t=0}^{T-1} \epsilon_t^2 \to 0$, the alternating updates approach a stationary point of the regularized robust objective F_τ. If instead $\epsilon_t \le \epsilon$ uniformly, the iterates approach an $O(\epsilon)$ stationary neighborhood.

The bound separates two sources of error. The first term is the standard policy optimization error and decreases as $O(1/T)$, while the second term captures the adversary tracking error through the discrepancy ϵ_t between q_t and the exact inner optimum $q^\star(\theta_t)$. Thus, accurate adversary tracking allows the policy update to behave increasingly like gradient ascent on the true reduced objective.

Proof sketch. Strong convexity of the inner problem guarantees a unique $q^\star(\theta)$. By Danskin's theorem, we have $\nabla F_\tau(\theta) = \nabla_\theta \mathcal{L}_\tau(\theta, q^\star(\theta))$. Define the gradient error

$$e_t = \nabla_\theta \mathcal{L}_\tau(\theta_t, q_t) - \nabla F_\tau(\theta_t).$$

This quantity measures the difference between the policy gradient actually used in the update and the exact gradient of the reduced smooth worst-group objective. By Assumption (iv), which states that the policy gradient is Lipschitz with respect to the adversarial weights, we obtain

$$\|e_t\| \le C\|q_t - q^\star(\theta_t)\|.$$

Finally, since the approximate adversary satisfies $\|q_t - q^\star(\theta_t)\| \le \epsilon_t$, we obtain

$$\|e_t\| \le C\|q_t - q^\star(\theta_t)\| \le C\epsilon_t.$$

Hence, the error in the policy gradient grows at most linearly with the adversary tracking error; if q_t closely tracks the optimal adversary $q^\star(\theta_t)$, the policy gradient used in the update is correspondingly close to the exact gradient $\nabla F_\tau(\theta_t)$ of the smooth worst-group objective.

The policy update can therefore be written as $\theta_{t+1} = \theta_t + \eta(\nabla F_\tau(\theta_t) + e_t)$. Using the L_F-smoothness of F_τ, together with $\eta \leq 1/(2L_F)$ and a standard bound on the cross term between $\nabla F_\tau(\theta_t)$ and e_t, gives

$$F_\tau(\theta_{t+1}) - F_\tau(\theta_t) \geq \frac{\eta}{2}\|\nabla F_\tau(\theta_t)\|^2 - \frac{\eta}{2}\|e_t\|^2.$$

This inequality has a direct interpretation, i.e., the actual improvement in F_τ is at least the improvement obtained from following the exact gradient, minus the loss caused by the approximate adversary. Summing the above inequality over $t = 0, \ldots, T-1$, using $F_\tau(\theta_T) \leq F_\tau^{\max}$, and substituting $\|e_t\| \leq C\epsilon_t$ yields the stated result.

When the inner adversary objective is smooth and strongly convex, projected gradient descent on q contracts toward $q^\star(\theta_t)$ for fixed θ_t under a suitable adversary step size. Thus, sufficiently accurate adversary updates make ϵ_t small. The theorem therefore shows that the adversary need not be solved exactly at every iteration; as long as it tracks its current optimum sufficiently well, the policy gradient remains a good approximation to the gradient of the smooth worst-group objective, and the alternating updates approach its stationary points.

Scope of the analysis. The results above characterize the optimization principle underlying adversarial reweighting rather than providing a global convergence guarantee for full LLM fine-tuning. In practice, both the policy and adversary are nonconvex neural networks, group values are estimated from finite logged data, and the policy and adversary use empirical surrogate objectives. Consequently, the assumptions of Theorem 1 need not hold globally. Nevertheless, Proposition 1 establishes that an expressive idealized adversary implements the intended worst-group criterion exactly, while Theorem 1 shows that, when a regularized adversary sufficiently tracks its inner optimum, alternating policy updates approach stationarity of a smooth approximation to that robust objective.

H Broader Impact

LLMs are increasingly fine-tuned using heterogeneous feedback from diverse users, domains, and instruction types. Under imbalanced data, optimizing average reward can disproportionately benefit frequent or easier inputs while degrading performance for underrepresented or difficult groups. Our adversarial reweighting framework aims to mitigate this failure mode by improving worst-group performance and reducing group-specific regressions, including settings where predefined group labels are unavailable.

Our fairness notions are nevertheless limited to performance disparities measured through observed or proxy rewards. In our experiments, these include LLM-based evaluators and SummaC, which may be noisy, incomplete, or systematically biased. If the reward signal is misaligned, adversarially emphasizing poorly performing samples may amplify evaluator artifacts or encourage reward hacking rather than improve actual user utility.

The choice of groups is also inherently incomplete. Task-defined slices such as cuisines, CPC sections, or input-length categories do not exhaustively represent sensitive or socially meaningful attributes, and improvements on these groups need not transfer to unseen or intersectional cohorts. We therefore do not claim demographic fairness or fairness with respect to all possible subgroups; our scope is worst-group robustness under task-specific performance criteria.

Finally, improving worst-group performance can trade off against average utility when objectives conflict. The stability parameter b provides a practical way to moderate this trade-off by limiting how strongly the adversary can concentrate on poorly performing groups. Deployment should therefore evaluate both aggregate and subgroup outcomes and select b according to the desired balance between worst-group robustness and overall utility, ideally with human feedback and application-specific safety checks. Our experiments use recipe and patent data in accordance with the licenses of the underlying sources, and no personally identifiable information is introduced beyond the original dataset content.