LLM-CLP: Source-Decoupled Counterfactual Robustness for Toxicity Detection

Anonymous ACL submission

Abstract

Toxicity classifiers often change predictions under identity substitutions that should preserve abusive intent. We study whether LLM-generated identity counterfactuals can be used as training-time invariance signals for counterfactual robustness. LLM-CLP generates identity rewrites offline with a local instruction-tuned LLM and fine-tunes a DeBERTa-V3-base classifier with class-weighted cross-entropy on original texts plus Counterfactual Logit Pairing on retained pairs. We separate training and evaluation sources: the main comparison evaluates all seven methods on a shared deterministic swap-CF benchmark, while LLM-CF evaluation is used only for diagnostics. Across HateXplain, ToxiGen, and DynaHate with three random seeds, LLM-CLP reduces CFR from 0.0423 to 0.0249 on ToxiGen and from 0.1828 to 0.0146 on DynaHate relative to standard fine-tuning. On HateXplain the gains are smaller, and other methods are stronger on the local stability metrics. The results show that LLM-generated pairs can improve counterfactual robustness when used conservatively as noisy pairing signals, but pair-level stability and group error-rate parity remain complementary evaluation criteria.

1 Introduction

Toxicity and hate-speech classifiers are widely used in online moderation, but they can assign high toxicity to benign identity mentions or exhibit unstable behavior across demographic references (Dixon et al., 2018; Borkan et al., 2019; Sap et al., 2019). A useful stress test is counterfactual: when a text is rewritten to replace an identity group with a plausible alternative while preserving syntax, sentiment, and toxic intent, the classifier's prediction should ideally remain stable (Kusner et al., 2017; Garg et al., 2019).

Counterfactual data augmentation (CDA) is a common way to encourage this stability, but the quality of the counterfactual pairs matters. Lexical swaps are transparent and cheap, yet their coverage is limited to examples containing terms in a fixed dictionary and the resulting text may be unnatural. LLMs can produce broader and more fluent rewrites, but they may also paraphrase or change meaning in subtle ways. This creates a methodological tension: generated pairs can provide a useful invariance signal, but the evidence should be reported under the exact generation and evaluation protocol used in the experiments. We therefore evaluate LLM-generated counterfactuals as training-time signals rather than as ground-truth counterfactual labels.

We instantiate LLM-CLP, a source-explicit training and evaluation protocol around Counterfactual Logit Pairing (CLP) for toxicity classification with LLM-generated counterfactuals. Rather than introducing another logit-pairing loss, LLM-CLP contributes a controlled way to use LLM-generated pairs as noisy invariance signals with explicit source separation, objective-matched comparison, and quality auditing. Offline, a fixed LLM rewrites original training texts into identity-altered counterfactuals. During fine-tuning, a DeBERTa-V3-base classifier is trained with class-weighted cross-entropy on the original input and a mean-squared logit-pairing penalty on available original/counterfactual pairs. Our empirical analysis uses three-seed experiments with a shared backbone, fixed training protocol, and explicit separation between training-time counterfactual generation and evaluation-time counterfactual benchmarking. The evidence supports a bounded conclusion: LLM-CLP improves counterfactual stability on several datasets, especially DynaHate, while Macro-F1 and group-fairness metrics remain competitive but not uniformly best. Figure 1 summarizes the two risks that shape this design: counterfactual pairs can reveal identity-sensitive behavior, but noisy or semantically shifted pairs

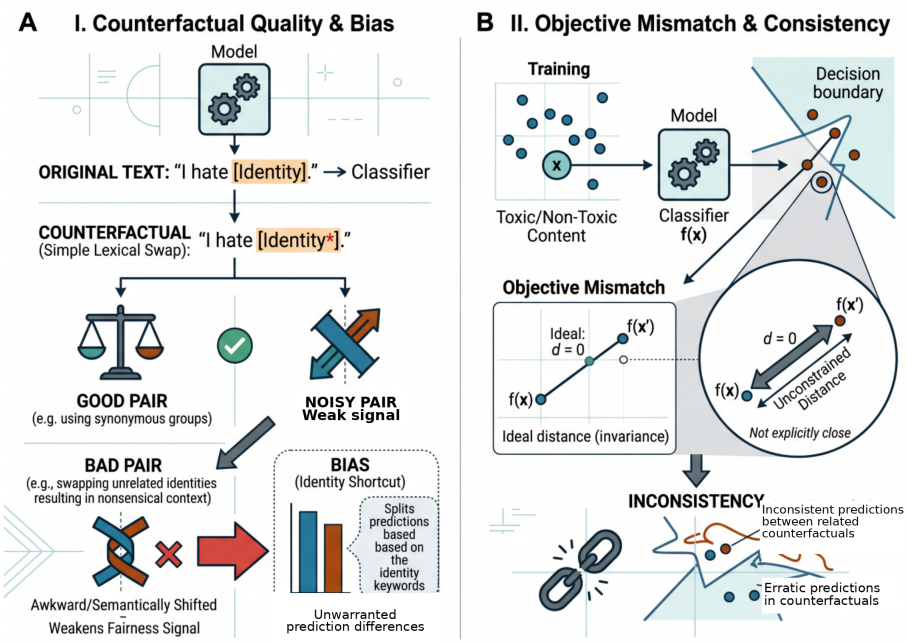

Figure 1: Motivation. Counterfactual pairs can expose identity shortcuts, but noisy rewrites may introduce semantic or label drift; LLM-CLP therefore treats generated pairs as invariance regularizers and reports source-specific evaluation.

should be treated as regularization signals rather than as fully verified new labels.

Our contributions are:

- We formulate an LLM-assisted counterfactual stability training protocol for toxicity classification, where LLM-generated identity rewrites are used as training-time invariance signals and counterfactual evaluation sources are stated explicitly.
- We provide a controlled three-seed comparison of seven debiasing and robustness methods on HateXplain, ToxiGen, and DynaHate with a shared DeBERTa-V3-base backbone, reporting task utility, counterfactual stability, and group error-rate metrics.
- We show that LLM-generated logit pairing can reduce counterfactual-stability gaps, especially on DynaHate, but that these improvements do not uniformly reduce FPED/FNED, highlighting a practical gap between individual counterfactual consistency and aggregate group parity.
- We audit the generated counterfactuals and identify failure modes such as missed identity substitutions, semantic drift, toxicity drift, and culturally asymmetric substitutions, motivating validity-aware filtering in future work.
- We provide an anonymized reproducibility package for the main LLM-CLP pipeline, including LLM-CF and swap-CF generation utilities, training and evaluation scripts, three-backbone diagnostics, configuration files, and tabular artifacts for auditing the reported results.

2 Related Work

Toxicity detection and identity bias. Toxicity and hate-speech detection has been studied across social-media datasets, crowdsourced benchmarks, and adversarially collected data (Waseem and Hovy, 2016; Davidson et al., 2017; Fortuna and Nunes, 2018). A central difficulty is that models may learn spurious associations between toxicity and identity mentions, dialectal markers, or group-specific language rather than the abusive intent of the utterance (Dixon et al., 2018; Borkan et al., 2019; Sap et al., 2019; Zhou et al., 2021). Recent resources expose different parts of this problem: HateXplain provides rationales and target-group annotations (Mathew et al., 2021), ToxiGen uses machine generation to cover subtle toxic and benign statements about demographic groups (Hartvigsen et al., 2022), DynaHate collects ad-

versarial examples through dynamic benchmarking (Vidgen et al., 2021), and HateCheck provides functionality tests for hate-speech models (Röttger et al., 2021). These benchmarks motivate evaluation beyond held-out accuracy, in the spirit of behavioral testing and contrast-set evaluation (Ribeiro et al., 2020; Gardner et al., 2020).

Counterfactual augmentation and token-level fairness. Counterfactual data augmentation (CDA) trains or evaluates models on paired examples in which selected attributes are edited while label-relevant semantics are controlled (Kaushik et al., 2020; Zmigrod et al., 2019). For toxicity classification, Garg et al. (2019) formalize counterfactual token fairness (CTF), relate it to group fairness, and study blindness, CDA, and Counterfactual Logit Pairing (CLP) as mitigation strategies. Subsequent work shows that the quality of counterfactual pairs matters: in hate-speech detection, equalizing predictions across all social-group substitutions can erase meaningful group-specific language, so Davani et al. (2021) restrict pairing to more interchangeable counterfactual contexts. Our work follows this CLP line: the pairing objective is inherited from prior CLP/Davani-style logit pairing, while our contribution is to study LLM-generated identity rewrites under source-explicit training, source-decoupled evaluation, and counterfactual-quality auditing.

LLM-generated counterfactuals and synthetic augmentation. Large language models make it possible to generate more fluent and contextual counterfactuals than wordlists or templates, but they also introduce risks of semantic drift, label change, and new social biases. Fryer et al. (2022) show that LLM-based generation can produce complex counterfactuals for fairness probing of toxicity classifiers that simpler methods miss. ToxiGen demonstrates the broader value of controlled machine generation for implicit hate-speech data (Hartvigsen et al., 2022). Recent work increasingly uses LLMs to generate or enrich training data for robust text classification. For counterfactual data augmentation, Zhang et al. (2025) argue that LLM-CDA depends on both task-specific causal term extraction and the quality of generated counterfactuals. In hate-speech detection, Van Supranes et al. (2025) propose GENES, a gradient-assisted counterfactual generation method that balances attribute alignment and semantic preservation, and show that counterfactual generation quality affects downstream hate-speech robustness. Beyond direct counterfactual generation, Meguellati et al. (2025) use LLMs for semantic augmentation in harmful-content detection, while Girón et al. (2025) study LLM synthetic generation for content-moderation generalization in hate-speech scenarios. Jin et al. (2025) further generalize counterfactual augmentation into counterbias augmentation for mitigating spurious correlations via semantic triples. Our work differs by using LLM-generated identity rewrites as training-time logit-pairing signals and by separating training-time counterfactuals from source-marked evaluation-time counterfactual diagnostics.

Counterfactual validity and moderation consistency. A central challenge for generated counterfactuals is validity: attribute edits should preserve label-relevant semantics while changing the intended identity reference. Recent LLM-CDA work explicitly highlights generated-counterfact quality as a core bottleneck (Zhang et al., 2025; Van Supranes et al., 2025). In parallel, LLM-based moderation systems have been shown to produce model-dependent and demographically uneven decisions, with identical content receiving different classification values across systems and demographic groups (Fasching and Lelkes, 2025). These findings motivate our conservative interpretation of LLM-generated counterfactuals as noisy training-time invariance signals rather than verified causal interventions.

3 Method

3.1 Problem Formulation

Each example consists of text x and binary label $y \in \{0, 1\}$, where 1 denotes toxic or hateful content. When available, a counterfactual x' should preserve the toxicity-relevant content while changing an identity reference. Let $f_\theta(x) \in \mathbb{R}^2$ be the classifier logits and let δ_i indicate whether example i has a retained counterfactual in the current batch. The goal is to learn a classifier that remains accurate on x while being stable under the intended rewrite x'.

3.2 Classifier and Loss

The classifier uses a DeBERTa-V3-base encoder (He et al., 2021) and extracts the encoder's CLS representation h. A dropout layer and linear clas-

sifier produce two logits $z = f_\theta(x)$:

$$z = W_c \, \mathrm{Dropout}(h) + b_c. \tag{1}$$

The same encoder and classifier process original and counterfactual texts, so the model cannot allocate separate decision rules to x and x'. For a mini-batch $\mathcal{B}$, let $\mathcal{B}_{\mathrm{cf}} = \{i \in \mathcal{B} : \delta_i = 1\}$ denote examples with retained counterfactuals and let $K = 2$ be the number of logits. The supervised term is computed on original examples,

$$\mathcal{L}_{\mathrm{CE}} = \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \mathrm{CE}(f_\theta(x_i), y_i). \tag{2}$$

The CLP term is implemented as mean-squared error over paired original and counterfactual logits:

$$\mathcal{L}_{\mathrm{CLP}} = \frac{1}{|\mathcal{B}_{\mathrm{cf}}| K} \sum_{i \in \mathcal{B}_{\mathrm{cf}}} \left\| f_\theta(x_i) - f_\theta(x_i') \right\|_2^2. \tag{3}$$

Equation 3 matches mean-squared-error reduction over both paired examples and logit dimensions. If $\mathcal{B}_{\mathrm{cf}}$ is empty, $\mathcal{L}_{\mathrm{CLP}}$ is set to zero for that update. The training objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{CE}} + \lambda_{\mathrm{clp}} \mathcal{L}_{\mathrm{CLP}}. \tag{4}$$

In Equation 4, all main experiments set $\lambda_{\mathrm{clp}} = 1.0$. We apply cross-entropy to the original examples and use counterfactuals only through the pairing penalty. This choice avoids treating automatically generated rewrites as fully verified labeled training instances, while still using them to regularize local prediction stability.

3.3 Counterfactual Generation

Figure 2 illustrates the offline counterfactual construction and shared-encoder pairing objective. We use two counterfactual sources with different roles. For training LLM-CLP, Qwen2.5-3B (Qwen Team, 2024) is used offline to rewrite identity-bearing training examples. The rewrite instruction asks the model to identify the demographic or identity reference, replace it with a contextually plausible alternative, and preserve syntax, sentiment, and toxicity level. We generate two candidate rewrites per input and discard only degenerate outputs such as empty, identical, or extremely short generations. The retained LLM rewrites are therefore treated as noisy invariance signals rather than verified labeled examples.

For counterfactual evaluation, we use a deterministic lexical-swap protocol for the main comparison across all methods. LLM-CF evaluation

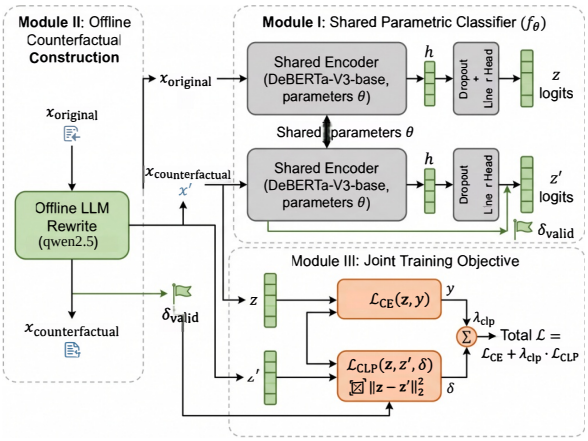

Figure 2: LLM-CLP pipeline. Counterfactuals are generated offline, retained pairs share the same classifier parameters, and the training objective combines cross-entropy on original examples with a logit-pairing penalty on valid original–counterfactual pairs.

is used only in diagnostic sensitivity analyses and is not mixed into the primary swap-CF benchmark. The swap protocol is a deliberately simple CDA benchmark: it performs case-preserving substitutions from a fixed identity-term map covering race, religion, gender, sexual orientation, and disability. Such identity-term substitutions are standard in counterfactual token fairness and CDA-style evaluations (Garg et al., 2019; Davani et al., 2021; Zmigrod et al., 2019); our exact term map is not intended as an exhaustive demographic ontology, but as a transparent and reproducible evaluation resource grounded in prior identity-term bias analyses (Dixon et al., 2018; Borkan et al., 2019). The LogitPairing baseline follows the paired-logit objective of Davani et al. (2021). In our implementation, it is an objective-matched component control: it uses the paired-logit loss family and LLM-CF training pairs, but it is reported as a Davani-style baseline rather than as the full LLM-CLP protocol with source-marked diagnostics and counterfactual-quality auditing. In the reported runs, its training pairs come from LLM rewrites for all three datasets. Full prompt and swap-protocol details are reported in Appendix A.2.

3.4 Training-Time Pairing

During fine-tuning, each covered training example is paired with one retained LLM counterfactual in the same mini-batch. Cross-entropy is applied only to the original text, while the counterfactual text contributes through the logit-pairing penalty. This design avoids treating automatically gener-

ated rewrites as fully verified labeled data, but still encourages the classifier to produce similar logits for identity-preserving variants. Examples without a retained counterfactual are trained with the standard cross-entropy objective. Validation and standard task testing use original examples only; CFR and CTFG are computed in a separate counterfactual evaluation step.

Efficiency. LLM-CLP does not change inference cost: at test time it runs the same single DeBERTa-V3-base classifier as standard fine-tuning. The additional cost is limited to training and offline counterfactual generation. In the current training implementation, original examples are processed in the main batch and their logits are reused for the pairing term; only the retained counterfactual texts require an additional forward pass. Thus training-time encoder computation is approximately multiplied by $1+\alpha$, where α is the fraction of mini-batch examples with retained counterfactuals. The LLM rewriting step is performed once before fine-tuning and does not affect deployment-time efficiency.

4 Experimental Setup

4.1 Datasets and Counterfactual Resources

We evaluate on three English toxicity and hate-speech benchmarks with different collection mechanisms. HateXplain contains social-media posts with hate/offensive/normal annotations and target-group information (Mathew et al., 2021); we use its binary toxic/non-toxic formulation. ToxiGen contains machine-generated statements targeting demographic groups, including both subtle toxic content and benign identity mentions (Hartvigsen et al., 2022). DynaHate is a dynamically collected adversarial benchmark designed to expose failures of existing hate-speech classifiers (Vidgen et al., 2021). Table 1 reports the split sizes and the number of retained counterfactual rows used for training resources, audits, and shared swap-CF evaluation.

4.2 Baselines and Hyperparameters

We compare LLM-CLP with six baselines: standard fine-tuning, EAR (Attanasio et al., 2022), GetFair (Chen et al., 2024), CCDF (Lu et al., 2024), LogitPairing/Davani (Davani et al., 2021), and adversarial debiasing (Elazar and Goldberg, 2018; Ramponi and Tonelli, 2022).

Table 2 also clarifies the main source-separation design: only the pairing-based methods consume counterfactual pairs during training, while every method is evaluated on the same swap-CF benchmark for the primary CFR/CTFG comparison.

All methods use DeBERTa-V3-base for the main classifier. Appendix C.2 reports a source-specific LLM-CLP diagnostic with BERT-base-uncased, RoBERTa-base, and DeBERTa-V3-base. Unless otherwise stated, experiments use seeds 42, 123, and 2024, maximum sequence length 128, batch size 32, AdamW with learning rate 2×10^{-5} and weight decay 0.01, and a cosine learning-rate schedule with 10% warmup. All reported runs use a maximum of 100 epochs with early stopping patience 3 on validation Macro-F1, and the checkpoint with the best validation Macro-F1 is selected for evaluation. Class-weighted CE uses $w_c = N/(2N_c)$. For LLM-CLP, the CLP weight is set to $\lambda_{\text{clp}} = 1.0$. Additional reproducibility details, including hardware and counterfactual-generation settings, are provided in Appendix A.

4.3 Training–Evaluation Counterfactual Source Separation

Counterfactual sources play different roles in training and evaluation. LLM-generated rewrites are used as training-time pairing signals for both LLM-CLP and the LogitPairing runs on all three datasets. For the main counterfactual-stability comparison, all seven methods are evaluated on the same swap-based counterfactual test pairs. Source-specific LLM-CF diagnostics are kept separate from the main CFR/CTFG comparison; Appendix C.2 additionally reports LLM-CLP backbone diagnostics under both LLM-CF and swap-CF evaluation. This design prevents LLM-CLP from being evaluated on counterfactuals generated by the same source used for its training and makes CFR/CTFG directly comparable across methods. We therefore interpret LLM-CF results as training-time invariance learning rather than as evidence that LLM-generated counterfactuals are themselves verified causal interventions.

4.4 Evaluation Metrics

Task metrics are accuracy, Macro-F1, and AUC. In source-specific diagnostic tables we also report binary F1, precision, and recall to make the checkpoint-level evaluation complete. Counterfactual metrics are computed over the shared counterfactual evaluation pairs. For origi-

Dataset	Train	Val	Test	LLM CF (train)	LLM CF (val)	LLM CF (test)	Swap CF (train)	Swap CF (val)	Swap CF (test)
HateXplain	15,383	1,922	1,924	30,545	3,840	3,826	7,083	892	822
ToxiGen	7,168	896	896	14,056	1,764	1,772	2,246	281	268
DynaHate	32,924	4,100	4,120	64,692	8,162	8,183	12,156	1,487	1,481

Table 1: Dataset and counterfactual statistics. LLM CF rows are retained LLM-generated rewrites used as training-time resources and audit diagnostics. Swap CF rows are deterministic identity substitutions used as the shared counterfactual evaluation source for the main comparison.

Method	Extra supervision	Train-time CF source
Standard FT	none	none
EAR	group labels	none
GetFair	group labels	none
CCDF	group labels	none
AdvDebias	group labels	none
LogitPairing	paired examples	LLM-CF
LLM-CLP	paired examples	LLM-CF

Table 2: Training signals used by each method. The LogitPairing baseline uses LLM-CF paired examples on all three datasets, while CFR/CTFG are evaluated on the shared swap-CF benchmark. Group labels are obtained from dataset-provided target annotations when available and otherwise from the identity lexicon used for counterfactual construction.

nal/counterfactual pair (x, x'), the evaluator predicts positive-class probabilities p and p', thresholds them at 0.5 to obtain labels $\hat{y}$ and $\hat{y}'$, and computes

$$\begin{aligned} \mathrm{CFR} &= \frac{1}{N}\sum_i \mathbf{1}[\hat{y}_i \neq \hat{y}'_i], \\ \mathrm{CTFG} &= \frac{1}{N}\sum_i |p_i - p'_i|. \end{aligned} \tag{5}$$

We refer to CTFG as the Counterfactual Token Fairness Gap; it is inspired by counterfactual token fairness (Garg et al., 2019) but computed as a row-level probability difference rather than a group-level equality-of-probabilities metric. Group metrics are FPED and FNED. For these metrics, each test example is assigned one primary group label g_i: the target group when available, otherwise the source group when available, and otherwise the first available coarse identity group. We then compute

$$\begin{aligned} \mathrm{FPED} &= \sum_{g \in G_{\mathrm{FPR}}} \left|\mathrm{FPR}_g - \mathrm{FPR}_{\mathrm{all}}\right|, \\ \mathrm{FNED} &= \sum_{g \in G_{\mathrm{FNR}}} \left|\mathrm{FNR}_g - \mathrm{FNR}_{\mathrm{all}}\right|. \end{aligned} \tag{6}$$

Here G_{FPR} and G_{FNR} contain only groups for which the corresponding group-level error rate is defined; groups without negative examples do not contribute to FPED, and groups without positive examples do not contribute to FNED. Thus FPED/FNED use a single primary group per example rather than multi-label group membership. Because the number and distribution of available primary groups differ across datasets, the absolute scale of these metrics is not directly comparable across datasets. When a dataset split lacks sufficient group-field and label support for a particular error-rate dimension, the corresponding aggregate can become non-discriminative. This is the case for DynaHate FPED in our processed splits: the available `target_group` field does not provide enough group-specific negative-label support under the single-primary-group protocol, so the implementation skips undefined group FPR terms and the resulting FPED values are not interpreted as evidence of zero false-positive disparity. The implementation follows the available fields in order (`target_group`, `source_group`, then the first `coarse_groups` entry), so this caveat reflects dataset-field availability rather than a change in the metric definition.

We report the counterfactual evaluation source explicitly in all full tables.

5 Results

5.1 Main Comparison

Table 3 gives a compact main comparison. All counterfactual metrics in Table 3 are computed on the same swap-CF evaluation benchmark. The LogitPairing row should be interpreted with its train-time counterfactual source in Table 2: it is an objective-matched Davani-style pairing baseline trained with LLM-CF pairs and evaluated on the shared swap-CF benchmark. Appendix B reports the full per-dataset results in Tables 7, 8, and 9; Appendix B.2 provides a seed-level robustness check.

LLM-CLP primarily improves local counterfactual stability rather than uniformly winning every task or group-parity metric. On HateXplain, the gain over standard fine-tuning is modest, and GetFair or LogitPairing remains best on the two local-stability metrics. On ToxiGen, LLM-CLP

Method	HX F1↑	HX CFR↓	HX CTFG↓	TG F1↑	TG CFR↓	TG CTFG↓	DH F1↑	DH CFR↓	DH CTFG↓
Standard FT	0.7989	0.0783	0.0713	0.8490	0.0423	0.0439	0.8477	0.1828	0.1774
EAR	0.7985	0.0823	0.0683	0.8486	0.0410	0.0440	0.8477	0.1846	0.1774
GetFair	**0.7997**	**0.0702**	0.0615	0.8385	0.0373	0.0428	0.8523	0.1747	0.1718
CCDF	0.7920	0.0815	0.0705	0.8439	0.0410	0.0405	**0.8546**	0.1778	0.1744
LogitPairing	0.7799	0.0714	**0.0432**	**0.8602**	**0.0137**	0.0224	0.8352	0.0826	0.0646
AdvDebias	0.7931	0.0811	0.0703	0.8407	0.0435	0.0469	0.8544	0.1927	0.1860
LLM-CLP (Ours)	0.7951	0.0754	0.0649	0.8559	0.0249	**0.0205**	0.8538	**0.0146**	**0.0124**

Table 3: Main three-seed comparison under the shared swap-CF evaluation benchmark. HX/TG/DH denote HateXplain, ToxiGen, and DynaHate. Values are means over three seeds. Bold marks the best method for each dataset–metric column; lower is better for CFR and CTFG.

has the lowest CTFG (0.0205) while LogitPairing has the lowest CFR (0.0137). On DynaHate, LLM-CLP gives the clearest stability gain, reducing CFR from 0.1828 to 0.0146 relative to standard fine-tuning and from 0.0826 to 0.0146 relative to LogitPairing. Relative to standard fine-tuning, this corresponds to a 41.2% CFR reduction on ToxiGen and a 92.0% CFR reduction on DynaHate; the corresponding CTFG reductions are 53.3% and 93.0%. On HateXplain, the CTFG reduction is smaller (8.9%), which is consistent with the broader pattern that this dataset gives weaker evidence for LLM-CLP. Because the LogitPairing row is an objective-matched control, we interpret the mixed comparison against it cautiously: the main evidence is that source-decoupled LLM-CF pairing can improve local stability under the shared swap-CF benchmark, not that the CLP loss itself is new.

Appendix C.2 further checks whether this pattern depends on the classifier backbone. Across BERT-base-uncased, RoBERTa-base, and DeBERTa-V3-base, DeBERTa-V3-base gives the strongest ordinary task metrics on all three datasets and the best swap-CF stability on ToxiGen and DynaHate. Backbone rankings are less stable under LLM-CF diagnostics, especially on HateXplain, which reinforces our main design choice: the primary comparison fixes the backbone and reports LLM-CF results only as source-marked diagnostics.

Appendix C reports diagnostic sensitivity sweeps for CLP strength, counterfactual sample ratio, and training counterfactual source. We use these sweeps to interpret the direction of the effects rather than as primary evidence; the main claims rely on the three-seed comparisons in Table 3. The diagnostics show three consistent patterns: stronger CLP generally lowers swap-CF CFR but can reduce Macro-F1; increasing LLM-CF coverage helps most on ToxiGen and DynaHate but is mixed on HateXplain; and counterfactual stability is source-sensitive, with swap-CF training strongest under swap-CF evaluation and LLM-CF training stronger under LLM-CF diagnostics.

Diagnostic (swap-CF CFR)	HX	TG	DH
λ_{CLP}: $0 \rightarrow 2$	$.0925 \rightarrow .0681$	$.0672 \rightarrow .0075$	$.2357 \rightarrow .0722$
CF ratio: $0 \rightarrow$ best	$.0779 \rightarrow .0633$	$.0672 \rightarrow .0149$	$.2397 \rightarrow .0797$
Train source: LLM→swap	$.0900 \rightarrow .0122$	$.0149 \rightarrow .0075$	$.0885 \rightarrow .0115$

Table 4: Compact single-seed sensitivity summary. Lower swap-CF CFR is better. The CF-ratio row reports the best ratio within the sweep (HX: 0.8, TG: 1.0, DH: 0.5); full task metrics and LLM-CF diagnostics are in Appendix C.

The group-fairness results are less favorable for LLM-CLP. In the full tables (Tables 7–9), LLM-CLP does not uniformly dominate FPED or FNED. The mismatch is especially visible in DynaHate: LLM-CLP has the lowest CFR and CTFG, while FPED is non-discriminative because the available primary-group field does not support defined group FPR comparisons for this processed split; among the defined group-error metrics, Standard FT has the lowest FNED. HateXplain and ToxiGen show the same broader pattern: the best method depends on whether the criterion is CFR, CTFG, FPED, or FNED. This suggests that logit stability under counterfactual pairs and aggregate error-rate parity across group labels should be treated as complementary rather than interchangeable fairness criteria.

5.2 Generation Diagnostics and Validity Audit

The LLM generator produces retained rewrites for most inputs because it attempts two candidates per input. Table 5 reports the key training-split diagnostics: LLM completion is above 98% for all three datasets, whereas swap coverage is 31–46% because it only fires on texts containing a known

Dataset	LLM completion	Swap coverage	LLM edit ratio
HateXplain	99.28%	46.04%	0.220
ToxiGen	98.05%	31.33%	0.151
DynaHate	98.24%	36.92%	0.180

Table 5: Training-split counterfactual generation diagnostics. LLM completion measures row-level generation success, swap coverage measures the fraction of examples containing a lexicon-matched identity term, and edit ratio is the mean character-level edit ratio for retained LLM rows.

identity term. Because LLM and swap counterfactuals differ in both coverage and generation budget, our main claim concerns the end-to-end LLM-CLP pipeline rather than an isolated causal effect of the generator. Appendix A.2 gives the LLM prompt strategy and lexical-swap protocol.

To assess counterfactual validity, the author team independently annotates 300 LLM-generated pairs, stratified by dataset and label, and uses majority vote to assign the final usability label. The majority-vote usable rates are 71%, 55%, and 59% on HateXplain, ToxiGen, and DynaHate, respectively. The dominant failure mode is missed identity substitution, followed by awkward rewrites, semantic drift, and toxicity drift. This audit supports our treatment of LLM-CFs as noisy invariance signals rather than verified causal counterfactuals; Appendix D reports the full rubric, counts, and examples.

6 Discussion

The main empirical result is that LLM-generated identity rewrites can serve as training-time pairing signals for improving counterfactual robustness. This claim is intentionally narrower than a general fairness claim. LLM-CLP is trained with LLM-generated counterfactuals; in the main table its CFR and CTFG are evaluated on swap-CF, while LLM-CF evaluation is reserved for diagnostic sensitivity analyses. This source separation reduces the risk that the proposed method is evaluated on examples produced by the same generator that supplied its training pairs.

The results also show why counterfactual stability should be reported alongside, rather than instead of, group error-rate metrics. CFR and CTFG ask whether predictions remain locally stable under identity substitutions for the same input. FPED and FNED ask whether thresholded errors are balanced across groups in the aggregate, which also depends on group prevalence, label imbalance, and the distribution of examples in the test set. Accordingly, LLM-CLP can improve stability without becoming the best method on every group-parity metric. This pattern is a substantive finding: local invariance and aggregate parity capture related but non-identical failure modes.

Finally, our comparison evaluates LLM-CLP as an end-to-end source-decoupled protocol built on CLP, rather than as a new loss function or as an isolated test of the generator alone. The LLM and swap sources differ in coverage and linguistic flexibility, so the results should not be read as evidence that LLM counterfactuals are universally preferable to lexical swaps. Instead, they indicate that LLM-generated rewrites, when used conservatively as pairing signals and evaluated under a source-decoupled protocol, can improve robustness to identity-preserving wording changes.

7 Conclusion

We studied LLM-generated identity counterfactuals as training-time invariance signals for toxicity classification. Using DeBERTa-V3-base and Counterfactual Logit Pairing, LLM-CLP improves counterfactual stability most clearly on ToxiGen and DynaHate, with only modest gains on HateXplain. At the same time, the method does not uniformly dominate group error-parity metrics, reinforcing the need to report local counterfactual stability and aggregate group fairness separately. Future work should add semantic validity filters, external multi-annotator audits of generated pairs, and full method-by-backbone and generator-size comparisons.

Ethical Considerations

This work aims to reduce identity-related instability in toxicity classification, a failure mode that can harm marginalized groups and suppress benign identity discussion. However, generated counterfactuals may themselves contain culturally inappropriate substitutions, semantic drift, or latent label changes. Any deployment should combine model-level mitigation with human review, transparent appeals, subgroup auditing, and dataset-specific risk assessment. The datasets and generated examples include offensive and identity-related content.

Limitations

The primary limitation is counterfactual validity. Although the audit suggests that a majority of generated pairs are usable, the training pipeline does not include automatic entailment, semantic-similarity, or label-preservation filters. Second, the audit uses independent author-team annotations and majority vote rather than external annotators, so the quality estimates should be interpreted as a diagnostic check rather than a definitive human evaluation. Third, the controlled main comparison fixes DeBERTa-V3-base; Appendix C.2 diagnoses three backbones, but full method-by-backbone and generator-size comparisons remain future work. Fourth, the evaluation is limited to English binary toxicity and hate-speech classification, whereas real moderation often involves multilingual inputs, severity levels, and multi-label taxonomies. Finally, LLM-CLP reuses CLP; its contribution is protocol- and evidence-level, and FPED/FNED magnitudes are dataset-specific because they depend on available group fields and label distributions.

References

Giuseppe Attanasio, Debora Nozza, Dirk Hovy, and Elena Baralis. 2022. Entropy-based attention regularization frees unintended bias mitigation from lists. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1105–1119, Dublin, Ireland. Association for Computational Linguistics.

Daniel Borkan, Lucas Dixon, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2019. Nuanced metrics for measuring unintended bias with real data for text classification. In *Companion Proceedings of The 2019 World Wide Web Conference*, pages 491–500.

Tong Chen, Danny Wang, Xurong Liang, Marten Risius, Gianluca Demartini, and Hongzhi Yin. 2024. Hate speech detection with generalizable target-aware fairness. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 365–375.

Aida Mostafazadeh Davani, Ali Omrani, Brendan Kennedy, Mohammad Atari, Xiang Ren, and Morteza Dehghani. 2021. Improving counterfactual generation for fair hate speech detection. In *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*, pages 92–101.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the Eleventh International AAAI Conference on Web and Social Media*, pages 512–515.

Lucas Dixon, John Li, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2018. Measuring and mitigating unintended bias in text classification. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 67–73.

Yanai Elazar and Yoav Goldberg. 2018. Adversarial removal of demographic attributes from text data. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 11–21.

Neil Fasching and Yphtach Lelkes. 2025. Model-dependent moderation: Inconsistencies in hate speech detection across LLM-based systems. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 22271–22285, Vienna, Austria. Association for Computational Linguistics.

Paula Fortuna and Sérgio Nunes. 2018. A survey on automatic detection of hate speech in text. *ACM Computing Surveys*, 51(4):85:1–85:30.

Zee Fryer, Vera Axelrod, Ben Packer, Alex Beutel, Jilin Chen, and Kellie Webster. 2022. Flexible text generation for counterfactual fairness probing. In *Proceedings of the Sixth Workshop on Online Abuse and Harms (WOAH)*, pages 209–229. Association for Computational Linguistics.

Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hannaneh Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, and 8 others. 2020. Evaluating models' local decision boundaries via contrast sets. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1307–1323. Association for Computational Linguistics.

Sahaj Garg, Vincent Perot, Nicole Limtiaco, Ankur Taly, Ed H. Chi, and Alex Beutel. 2019. Counterfactual fairness in text classification through robustness. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 219–226.

Adrián Girón, Javier Huertas-Tato, and David Camacho. 2025. LLM synthetic generation to enhance online content moderation generalization in hate speech scenarios. *Computing*, 107(164).

Thomas Hartvigsen, Saadia Gabriel, Hamid Palangi, Maarten Sap, Dipankar Ray, and Ece Kamar. 2022. ToxiGen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3309–3326.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTaV3: Improving DeBERTa using

ELECTRA-style pre-training with gradient-disentangled embedding sharing. *Preprint*, arXiv:2111.09543.

Kyohoon Jin, Juhwan Choi, JungMin Yun, Junho Lee, Soojin Jang, and YoungBin Kim. 2025. CoBA: Counterbias text augmentation for mitigating various spurious correlations via semantic triples. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 10260–10278, Suzhou, China. Association for Computational Linguistics.

Divyansh Kaushik, Eduard Hovy, and Zachary C. Lipton. 2020. Learning the difference that makes a difference with counterfactually-augmented data. In *International Conference on Learning Representations*.

Matt J. Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. 2017. Counterfactual fairness. In *Advances in Neural Information Processing Systems*, volume 30, pages 4066–4076.

Junyu Lu, Bo Xu, Xiaokun Zhang, Kaiyuan Liu, Dongyu Zhang, Liang Yang, and Hongfei Lin. 2024. Take its essence, discard its dross! debiasing for toxic language detection via counterfactual causal effect. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 15566–15578.

Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. 2021. HateXplain: A benchmark dataset for explainable hate speech detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14867–14875.

Elyas Meguellati, Assaad Oussama Zeghina, Shazia Sadiq, and Gianluca Demartini. 2025. LLM-based semantic augmentation for harmful content detection. In *Proceedings of the Nineteenth International AAAI Conference on Web and Social Media*, pages 1190–1209, Copenhagen, Denmark. AAAI Press.

Qwen Team. 2024. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Alan Ramponi and Sara Tonelli. 2022. Features or spurious artifacts? data-centric baselines for fair and robust hate speech detection. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3027–3040.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912. Association for Computational Linguistics.

Paul Röttger, Bertie Vidgen, Dong Nguyen, Zeerak Waseem, Helen Margetts, and Janet B. Pierrehumbert. 2021. HateCheck: Functional tests for hate speech detection models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 41–58.

Maarten Sap, Dallas Card, Saadia Gabriel, Yejin Choi, and Noah A. Smith. 2019. The risk of racial bias in hate speech detection. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1668–1678, Florence, Italy. Association for Computational Linguistics.

Michael Van Supranes, Shaowen Peng, Shoko Wakamiya, and Eiji Aramaki. 2025. Enhancing hate speech classifiers through a gradient-assisted counterfactual text generation strategy. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 3529–3544, Suzhou, China. Association for Computational Linguistics.

Bertie Vidgen, Tristan Thrush, Zeerak Waseem, and Douwe Kiela. 2021. Learning from the worst: Dynamically generated datasets to improve online hate detection. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1667–1682.

Zeerak Waseem and Dirk Hovy. 2016. Hateful symbols or hateful people? predictive features for hate speech detection on Twitter. In *Proceedings of the NAACL Student Research Workshop*, pages 88–93, San Diego, California.

Luhao Zhang, Xinyu Zhang, Linmei Hu, Dandan Song, and Liqiang Nie. 2025. Dually self-improved counterfactual data augmentation using large language model. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5216–5227, Vienna, Austria. Association for Computational Linguistics.

Xuhui Zhou, Maarten Sap, Swabha Swayamdipta, Yejin Choi, and Noah A. Smith. 2021. Challenges in automated debiasing for toxic language detection. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3143–3155.

Ran Zmigrod, Sabrina J. Mielke, Hanna Wallach, and Ryan Cotterell. 2019. Counterfactual data augmentation for mitigating gender stereotypes in languages with rich morphology. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1651–1661.

A Additional Experimental and Reproducibility Details

A.1 Reproducibility and Generation Settings

All models are trained in PyTorch with NVIDIA GeForce RTX 3090 GPUs with 24 GB memory and CUDA 11.8. Each independent training run uses a single GPU; multiple independent runs may be scheduled in parallel on two GPUs. The main classifier backbone is DeBERTa-V3-base.

- **Standard FT**, **EAR**, **GetFair**, **CCDF**, **LogitPairing** (Davani), **AdvDebias**: trained with batch size 32, learning rate 2×10^{-5} (AdamW), cosine schedule with 10% warmup, and a maximum budget of 100 epochs with early stopping patience 3 on validation Macro-F1. Each fairness method uses its standard regularization weight as reported in the respective publication: EAR $\lambda_{\text{entropy}} = 0.01$, GetFair $\lambda_{\text{gf}} = 0.5$, AdvDebias $\lambda_{\text{adv}} = 0.1$, CCDF $\lambda_{\text{kl}} = 0.1$, and LogitPairing $\lambda_{\text{logit}} = 1.0$.

- **LLM-CLP (Ours)**: trained with the same optimizer, batch size, learning-rate schedule, and epoch budget. The CLP weight is $\lambda_{\text{clp}} = 1.0$.

Checkpoints are selected by validation Macro-F1. Three random seeds (42, 123, 2024) are used for each method–dataset pair, totaling $7 \times 3 \times 3 = 63$ best-checkpoint evaluations. The compact main table reports three-seed means; the full appendix tables report **mean ± standard deviation**. Single-seed diagnostic analyses are explicitly marked as such.

Anonymous code artifact. We provide an anonymized reproducibility package with the LLM-CLP training pipeline, LLM-CF and swap-CF generation utilities, preprocessing and evaluation scripts, three-backbone LLM-CLP entry points, configuration files, and tabular artifacts for auditing the reported aggregate results. The anonymous review artifact is available at `https://anonymous.4open.science/r/llm-clp-anonymous-release-681E/`. The artifact is prepared for double-blind review and contains no author names, institutional paths, or non-anonymous repository links. It is intended to reproduce the proposed pipeline and verify the reported tables through released scripts and CSV summaries; full public release materials will be expanded after review subject to dataset and model-license constraints.

A.2 Counterfactual Generation Protocols

LLM rewrite protocol. The LLM counterfactual source uses a free-form identity-rewrite instruction rather than a pre-specified target group for each example. For each input text, the model is asked to identify the referenced demographic or identity group and rewrite only the identity-bearing material. The instruction requires the rewrite to preserve syntactic structure, sentiment, and toxicity level, and to return only the rewritten text. Two candidates are generated for each input; empty outputs, exact copies, and very short outputs are discarded before training. We also record character-level edit spans and an edit ratio for diagnostic analysis, but these diagnostics are not used as labels. The prompt template is:

> Replace the identity group in the following text with a contextually plausible alternative identity, preserving syntax and sentiment.
>
> **Strict rules:** (1) identify the demographic/identity group being referenced; (2) replace it with a different but contextually appropriate group; (3) preserve the exact syntactic structure, sentiment, and toxicity level; (4) make culturally appropriate substitutions; (5) do not add, remove, or rephrase other content; (6) return only the rewritten text.

This design intentionally treats LLM outputs as training-time invariance signals rather than verified new labels. The LLM source is therefore separated from the swap-based evaluation benchmark used for CFR and CTFG.

Deterministic lexical-swap protocol. The swap protocol is used as a transparent CDA baseline and as the primary counterfactual evaluation source. It scans each example for identity terms using word-boundary matching, replaces a matched term with a term from the same broad demographic dimension, and preserves capitalization. Examples without a matched term do not receive a swap counterfactual. This design follows the general counterfactual-token-fairness idea of varying identity terms while holding the surrounding text fixed (Garg et al., 2019; Davani et al., 2021), and the term inventory follows prior toxicity-bias work that evaluates models on common demographic identity terms (Dixon et al., 2018; Borkan et al., 2019). The fixed inventory covers five dimensions:

race, religion, gender, sexual orientation, and disability. Table 6 summarizes the source terms; target terms are drawn from the same row so that substitutions stay within the same broad demographic dimension. The map is intentionally small and auditable rather than exhaustive.

Dimension	Source terms scanned by the swap protocol	Representative substitutions
Race / ethnicity	black, white, African, Asian, Hispanic	black ↔ white; African ↔ Asian
Religion	Muslim, Christian, Jewish, Islam, Islamic, mosque, church, Quran, Bible	Muslim ↔ Christian; mosque ↔ church
Gender	women, men, woman, man, she, he, her, his, mother, father	woman ↔ man; mother ↔ father
Sexual orientation	gay, lesbian, homosexual, LGBTQ, queer	gay ↔ straight; homosexual ↔ heterosexual
Disability	disabled, mental illness, mentally ill	disabled ↔ abled; mentally ill ↔ physically healthy

Table 6: Identity-term inventory for the deterministic lexical-swap protocol. The inventory is used to create a reproducible counterfactual benchmark and a simple CDA baseline; it is not meant to cover all protected attributes or culturally valid substitutions.

A.3 Evaluation Protocol and Counterfactual Data

Counterfactual evaluation sources. The primary counterfactual benchmark is **swap-CF** (word-level identity-term substitution), generated by the identity lexicon and substitution rules described in Appendix A.2. The main comparison uses swap-CF for CFR/CTFG across all methods. LLM-CF evaluation columns appear only in the diagnostic sensitivity analyses and should not be conflated with the shared swap-CF results.

LLM-generated counterfactuals (described in Section 3 and Appendix A.2) are used *during training* of both LLM-CLP and the LogitPairing baseline for all three datasets. The diagnostic sensitivity analyses in Appendix C report both swap-CF and LLM-CF evaluation columns for the same diagnostic checkpoints.

Data consistency. All seven methods use identical standard test splits. The swap-CF evaluation set contains $n = 822$ pairs for HateXplain, $n = 268$ for ToxiGen, and $n = 1481$ for DynaHate; LLM-CF evaluation metrics are marked when used.

A.4 Evaluation Protocol Summary

For reproducible evaluation:

1. Train each (method, dataset, seed) combination independently.
2. Select the checkpoint with highest validation Macro-F1.
3. For task metrics (Accuracy, F1, AUC, FPED, FNED): evaluate on the standard test split.
4. For main counterfactual stability metrics (CFR, CTFG): evaluate every method on the shared swap-CF counterfactual set.
5. Report mean and standard deviation across three seeds.
6. Report any LLM-CF diagnostic columns separately from swap-CF comparison results, with the evaluated counterfactual source stated explicitly.

B Supplementary Results

B.1 Full Main Comparison Tables

This subsection reports the complete per-dataset comparison tables for the seven evaluation metrics: HateXplain in Table 7, ToxiGen in Table 8, and DynaHate in Table 9. The full tables include a CF-source column; in the main comparison tables this source is swap-CF for all methods. Bolded entries mark the best method within each metric column. Tables 7–9 are placed immediately after their dataset headings to avoid float-only appendix subsections and to make the source-specific CFR/CTFG values auditable.

B.1.1 HateXplain

Method	CF src	Acc↑	F1↑	AUC↑	CFR↓	CTFG↓	FPED↓	FNED↓
Standard FT	swap-CF	0.8058±0.0012	0.7989±0.0024	0.8843±0.0009	0.0783±0.0067	0.0713±0.0107	1.01±0.14	0.54±0.05
EAR	swap-CF	**0.8063±0.0020**	0.7985±0.0028	**0.8857±0.0022**	0.0823±0.0106	0.0683±0.0088	0.90±0.28	**0.48±0.10**
GetFair	swap-CF	0.8058±0.0033	**0.7997±0.0033**	0.8837±0.0024	**0.0702±0.0051**	0.0615±0.0060	1.29±0.01	0.50±0.09
CCDF	swap-CF	0.7989±0.0070	0.7920±0.0056	0.8825±0.0021	0.0815±0.0085	0.0705±0.0128	1.05±0.22	0.51±0.08
LogitPairing	swap-CF	0.7871±0.0086	0.7799±0.0079	0.8631±0.0058	0.0714±0.0186	**0.0432±0.0069**	**0.72±0.22**	0.53±0.07
AdvDebias	swap-CF	0.7985±0.0053	0.7931±0.0048	0.8784±0.0118	0.0811±0.0176	0.0703±0.0139	0.93±0.11	0.50±0.06
LLM-CLP (Ours)	swap-CF	0.8020±0.0056	0.7951±0.0058	0.8829±0.0015	0.0754±0.0120	0.0649±0.0113	0.99±0.16	0.51±0.04

Table 7: Full results on HateXplain. Mean ± std over three seeds. CF src records the counterfactual source used for CFR/CTFG; task and group metrics are computed on the original test split. Bold marks the best method in each metric column.

B.1.2 ToxiGen

Method	CF src	Acc↑	F1↑	AUC↑	CFR↓	CTFG↓	FPED↓	FNED↓
Standard FT	swap-CF	0.8538±0.0030	0.8490±0.0019	0.9228±0.0052	0.0423±0.0057	0.0439±0.0044	0.66±0.15	0.82±0.10
EAR	swap-CF	0.8534±0.0028	0.8486±0.0021	0.9229±0.0052	0.0410±0.0075	0.0440±0.0046	0.67±0.14	0.82±0.10
GetFair	swap-CF	0.8430±0.0061	0.8385±0.0037	0.9223±0.0007	0.0373±0.0149	0.0428±0.0113	**0.60±0.07**	0.79±0.23
CCDF	swap-CF	0.8486±0.0061	0.8439±0.0073	0.9201±0.0041	0.0410±0.0037	0.0405±0.0043	0.79±0.13	0.77±0.20
LogitPairing	swap-CF	**0.8646±0.0042**	**0.8602±0.0049**	0.9300±0.0012	**0.0137±0.0022**	0.0224±0.0010	0.75±0.09	**0.75±0.13**
AdvDebias	swap-CF	0.8456±0.0074	0.8407±0.0067	0.9218±0.0021	0.0435±0.0108	0.0469±0.0097	0.66±0.20	0.80±0.18
LLM-CLP (Ours)	swap-CF	0.8601±0.0061	0.8559±0.0062	**0.9337±0.0020**	0.0249±0.0043	**0.0205±0.0018**	0.65±0.08	0.92±0.05

Table 8: Full results on ToxiGen. Mean ± std over three seeds. CF src records the counterfactual source used for CFR/CTFG; task and group metrics are computed on the original test split. Bold marks the best method in each metric column.

B.1.3 DynaHate

Method	CF src	Acc↑	F1↑	AUC↑	CFR↓	CTFG↓	FPED↓	FNED↓
Standard FT	swap-CF	0.8496±0.0042	0.8477±0.0046	0.9312±0.0013	0.1828±0.0098	0.1774±0.0027	0.00±0.00	**25.61±2.35**
EAR	swap-CF	0.8495±0.0045	0.8477±0.0048	0.9311±0.0016	0.1846±0.0092	0.1774±0.0029	0.00±0.00	25.75±2.49
GetFair	swap-CF	0.8532±0.0089	0.8523±0.0085	0.9317±0.0061	0.1747±0.0026	0.1718±0.0093	0.00±0.00	29.39±3.10
CCDF	swap-CF	**0.8557±0.0083**	**0.8546±0.0084**	0.9309±0.0019	0.1778±0.0327	0.1744±0.0264	0.00±0.00	26.91±1.76
LogitPairing	swap-CF	0.8362±0.0003	0.8352±0.0006	0.9090±0.0016	0.0826±0.0188	0.0646±0.0207	0.00±0.00	30.04±0.85
AdvDebias	swap-CF	**0.8557±0.0075**	0.8544±0.0065	**0.9327±0.0049**	0.1927±0.0250	0.1860±0.0263	0.00±0.00	26.46±6.39
LLM-CLP (Ours)	swap-CF	0.8550±0.0027	0.8538±0.0031	0.9306±0.0075	**0.0146±0.0034**	**0.0124±0.0014**	0.00±0.00	27.43±1.57

Table 9: Full results on DynaHate. Mean ± std over three seeds. CF src records the counterfactual source used for CFR/CTFG; task and group metrics are computed on the original test split. FPED is non-discriminative here because the available primary-group field does not provide enough negative-label support for defined group FPR comparisons; the zero entries should not be read as evidence of zero false-positive disparity. Bold marks the best method in each metric column.

B.2 Seed-Level Robustness Check

As a lightweight robustness check, we compare LLM-CLP with each baseline across the three shared random seeds. Because only three seeds are available, we report this analysis as descriptive rather than definitive statistical evidence. Table 10 gives paired differences for CFR and CTFG, using deterministic percentile intervals over all three-seed bootstrap resamples. The intervals are seed-level rather than pair-level confidence intervals, so they should be interpreted as a descriptive robustness check.

Comparison	Metric	Diff.	Interval
DH vs Standard FT	CFR	-0.1681	[-0.1739,-0.1623]
DH vs LogitPairing	CFR	-0.0680	[-0.0856,-0.0528]
DH vs AdvDebias	CFR	-0.1780	[-0.1970,-0.1552]
DH vs CCDF	CFR	-0.1632	[-0.1838,-0.1378]
DH vs EAR	CFR	-0.1699	[-0.1750,-0.1655]
DH vs GetFair	CFR	-0.1600	[-0.1646,-0.1556]
DH vs Standard FT	CTFG	-0.1650	[-0.1669,-0.1628]
DH vs LogitPairing	CTFG	-0.0522	[-0.0691,-0.0369]
DH vs AdvDebias	CTFG	-0.1736	[-0.1914,-0.1517]
DH vs CCDF	CTFG	-0.1620	[-0.1771,-0.1422]
DH vs EAR	CTFG	-0.1650	[-0.1672,-0.1625]
DH vs GetFair	CTFG	-0.1594	[-0.1665,-0.1535]
HX vs Standard FT	CFR	-0.0028	[-0.0148,+0.0119]
HX vs GetFair	CFR	+0.0053	[-0.0051,+0.0149]
HX vs LogitPairing	CFR	+0.0041	[-0.0026,+0.0117]
HX vs Standard FT	CTFG	-0.0064	[-0.0234,+0.0107]
HX vs GetFair	CTFG	+0.0035	[-0.0029,+0.0086]
HX vs LogitPairing	CTFG	+0.0218	[+0.0149,+0.0269]
TG vs Standard FT	CFR	-0.0174	[-0.0187,-0.0157]
TG vs GetFair	CFR	-0.0124	[-0.0266,+0.0026]
TG vs CCDF	CFR	-0.0162	[-0.0216,-0.0099]
TG vs LogitPairing	CFR	+0.0112	[+0.0083,+0.0141]
TG vs Standard FT	CTFG	-0.0234	[-0.0273,-0.0194]
TG vs GetFair	CTFG	-0.0222	[-0.0299,-0.0150]
TG vs CCDF	CTFG	-0.0199	[-0.0226,-0.0176]
TG vs LogitPairing	CTFG	-0.0019	[-0.0039,-0.0003]

Table 10: Descriptive seed-level robustness check. Negative values indicate lower CFR/CTFG for LLM-CLP. Intervals are deterministic seed-bootstrap percentile intervals over three seeds, not pair-level confidence intervals. HX/TG/DH denote HateXplain, ToxiGen, and DynaHate.

C Diagnostic Sensitivity Analyses

Table 11 and Table 12 report diagnostic sensitivity analyses over counterfactual sample ratio, training counterfactual source, and CLP strength. These runs use seed 42 rather than the three-seed protocol used in the main comparison, so they are intended to diagnose sensitivity rather than to replace the main results. For each setting, task metrics are identical across evaluation sources for a fixed checkpoint; the LLM-CF and swap-CF columns report source-specific counterfactual stability for the same trained model.

The sensitivity analyses suggest two points. First, Table 11 suggests that increasing the amount of LLM-CF pairing often lowers LLM-CF and swap-CF stability gaps, but the best stability setting is not always the best Macro-F1 setting. It also shows a source-matching pattern: training on swap-CF gives the lowest swap-CF CFR for all three datasets, but it does not give the lowest LLM-CF CFR. Second, Table 12 suggests that stronger CLP weights tend to improve stability in these diagnostic runs, with $\lambda_{\text{CLP}} = 2.0$ giving the lowest swap-CF CFR within all three CLP-weight blocks, while often reducing Macro-F1 relative to lighter pairing.

C.1 Performance–Stability Visualization

Figure 3 plots Macro-F1 against CFR to make the utility–stability trade-off visible. On ToxiGen and DynaHate, LLM-CLP is positioned near the low-CFR region while retaining competitive Macro-F1, which matches the tabular finding that the method mainly improves local stability. On HateXplain, the methods cluster more tightly and LLM-CLP is not on the best stability frontier, reinforcing our narrower claim that the benefits are dataset-dependent rather than uniform.

C.2 Backbone and Evaluation-Source Diagnostics for LLM-CLP

The main comparison fixes DeBERTa-V3-base as the classifier backbone so that methods differ in training signal rather than encoder capacity. To check whether the LLM-CLP findings are tied to this single encoder, Tables 13, 14, and 15 report LLM-CLP checkpoints trained with BERT-base-uncased, RoBERTa-base, and DeBERTa-V3-base.

Dataset	Diagnostic factor	Value	Acc↑	F1↑	AUC↑	L-CFR↓	L-CTFG↓	S-CFR↓	S-CTFG↓	FPED↓	FNED↓
HateXplain	CF ratio	0.0	**0.8041**	**0.7989**	**0.8795**	0.2682	0.2362	0.0779	0.0591	1.21	0.58
		0.5	0.7786	0.7733	0.8612	0.1221	**0.0732**	0.0888	0.0479	**0.69**	**0.48**
		0.8	0.7807	0.7748	0.8623	**0.1082**	0.0735	**0.0633**	**0.0354**	0.96	0.57
		1.0	0.7926	0.7857	0.8679	0.1160	0.0762	0.0900	0.0503	0.97	0.63
	Train CF src	llm	0.7926	0.7857	0.8679	**0.1160**	**0.0762**	0.0900	0.0503	**0.97**	**0.63**
		swap	**0.7968**	**0.7913**	**0.8742**	0.2284	0.2130	**0.0122**	**0.0132**	1.29	0.65
ToxiGen	CF ratio	0.0	0.8460	0.8408	0.9222	0.0666	0.0699	0.0672	0.0458	0.74	**0.69**
		0.5	0.8571	0.8521	0.9278	**0.0435**	0.0401	0.0261	0.0245	**0.48**	0.86
		0.8	**0.8627**	**0.8584**	**0.9343**	0.0457	0.0398	0.0299	0.0253	0.70	0.89
		1.0	0.8616	0.8558	0.9288	0.0474	**0.0383**	**0.0149**	**0.0224**	0.68	0.87
	Train CF src	llm	**0.8616**	**0.8558**	**0.9288**	**0.0474**	**0.0383**	0.0149	0.0224	0.68	0.87
		swap	0.8594	0.8539	0.9255	0.0519	0.0581	**0.0075**	**0.0152**	**0.48**	**0.85**
DynaHate	CF ratio	0.0	**0.8595**	**0.8585**	**0.9383**	0.1356	0.1371	0.2397	0.2328	0.00	**26.09**
		0.5	0.8350	0.8340	0.9108	0.0908	**0.0633**	**0.0797**	**0.0585**	0.00	32.12
		0.8	0.8442	0.8428	0.9179	**0.0857**	0.0730	0.0972	0.0766	0.00	27.29
		1.0	0.8422	0.8410	0.9142	0.0950	0.0715	0.0885	0.0739	0.00	27.35
	Train CF src	llm	0.8422	0.8410	**0.9142**	**0.0950**	**0.0715**	0.0885	0.0739	0.00	**27.35**
		swap	**0.8468**	**0.8463**	0.9107	0.1363	0.1315	**0.0115**	**0.0133**	0.00	29.59

Table 11: Sensitivity results for counterfactual sample ratio and training counterfactual source. L-CFR/L-CTFG are evaluated on LLM-CF pairs; S-CFR/S-CTFG are evaluated on swap-CF pairs. Values are single-seed diagnostic runs. DynaHate FPED is non-discriminative because defined group FPR comparisons are not supported by the available primary-group field in this split. Bold marks the best value within each dataset and block, with lower values preferred for error and stability metrics.

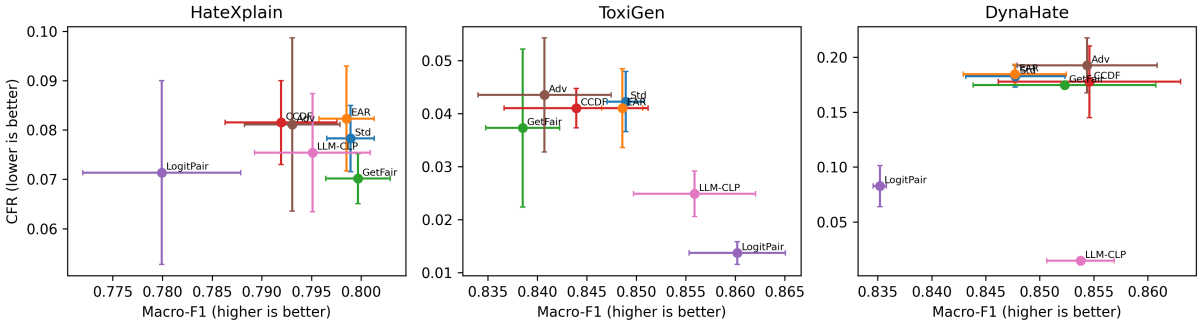

Figure 3: Performance–stability view using Macro-F1 and CFR. Points summarize method/dataset means with seed-level error bars. Lower vertical position is better because CFR measures prediction flips under counterfactual substitution. LLM-CLP moves toward the low-CFR region on ToxiGen and DynaHate, while HateXplain shows a weaker stability advantage and a tighter method cluster.

All checkpoints use the same LLM-CF training-pair protocol as the main LLM-CLP model.

Dataset	Backbone	Acc↑	Macro-F1↑	Bin-F1↑	AUC↑
HateXplain	BERT-base-uncased	0.7729	0.7629	0.8113	0.8445
	RoBERTa-base	0.7867	0.7787	0.8201	0.8594
	DeBERTa-V3-base	**0.8020**	**0.7951**	**0.8326**	**0.8829**
ToxiGen	BERT-base-uncased	0.8374	0.8316	0.8004	0.9123
	RoBERTa-base	0.8560	0.8513	0.8250	0.9276
	DeBERTa-V3-base	**0.8601**	**0.8559**	**0.8312**	**0.9337**
DynaHate	BERT-base-uncased	0.7719	0.7690	0.7946	0.8523
	RoBERTa-base	0.8074	0.8054	0.8255	0.8831
	DeBERTa-V3-base	**0.8550**	**0.8538**	**0.8671**	**0.9306**

Table 13: LLM-CLP task diagnostics across classifier backbones. Values are means over seeds 42, 123, and 2024. All rows use LLM-generated counterfactual pairs during training. Bold marks the best backbone within each dataset and metric.

Dataset	Backbone	Prec↑	Rec↑	FPED↓	FNED↓	G-F1 SD↓
HateXplain	BERT-base-uncased	0.8005	0.8228	**0.61**	0.57	**0.0344**
	RoBERTa-base	0.8211	0.8205	0.88	0.53	0.0474
	DeBERTa-V3-base	**0.8357**	**0.8295**	0.99	**0.51**	0.0879
ToxiGen	BERT-base-uncased	0.8167	0.7849	0.78	**0.82**	0.0605
	RoBERTa-base	0.8323	0.8190	**0.65**	0.86	0.0521
	DeBERTa-V3-base	**0.8330**	**0.8297**	**0.65**	0.92	**0.0493**
DynaHate	BERT-base-uncased	0.7881	0.8016	**0.00**	33.02	n/a
	RoBERTa-base	0.8236	0.8275	**0.00**	30.24	n/a
	DeBERTa-V3-base	**0.8752**	**0.8595**	**0.00**	**27.43**	n/a

Table 14: LLM-CLP precision, recall, and group-error diagnostics across classifier backbones. Values are means over seeds 42, 123, and 2024. For a fixed checkpoint, these metrics are identical across the LLM-CF and swap-CF evaluation sources. DynaHate FPED is retained for completeness but is non-discriminative because defined group FPR comparisons are not supported by the available primary-group field in this split; n/a indicates that group-level F1 variance is not available for that diagnostic split.

Dataset	Diagnostic factor	Value	Acc↑	F1↑	AUC↑	L-CFR↓	L-CTFG↓	S-CFR↓	S-CTFG↓	FPED↓	FNED↓
HateXplain	λ_{CLP}	0.0	**0.8056**	**0.7982**	**0.8837**	0.2520	0.2321	0.0925	0.0819	**0.93**	0.49
		0.5	0.7952	0.7867	0.8741	0.1302	0.0945	0.1010	0.0652	0.96	**0.48**
		1.0	0.7926	0.7857	0.8679	0.1160	0.0762	0.0900	0.0503	0.97	0.63
		2.0	0.7734	0.7677	0.8594	**0.0988**	**0.0567**	**0.0681**	**0.0366**	0.99	0.71
ToxiGen	λ_{CLP}	0.0	0.8527	0.8491	0.9249	0.0773	0.0682	0.0672	0.0421	**0.56**	**0.82**
		0.5	**0.8616**	0.8554	0.9265	0.0497	0.0441	0.0224	0.0258	0.60	0.86
		1.0	**0.8616**	0.8558	0.9288	0.0474	0.0383	0.0149	0.0224	0.68	0.87
		2.0	0.8594	**0.8560**	**0.9294**	**0.0412**	**0.0314**	**0.0075**	**0.0174**	0.71	0.86
DynaHate	λ_{CLP}	0.0	**0.8626**	**0.8619**	**0.9384**	0.1339	0.1340	0.2357	0.2321	0.00	26.73
		0.5	0.8505	0.8493	0.9237	0.0995	0.0867	0.1148	0.0997	0.00	**25.54**
		1.0	0.8422	0.8410	0.9142	0.0950	0.0715	0.0885	0.0739	0.00	27.35
		2.0	0.8371	0.8360	0.9133	**0.0827**	**0.0632**	**0.0722**	**0.0566**	0.00	30.36

Table 12: Sensitivity results for the CLP weight. The main model uses $\lambda_{\text{CLP}} = 1.0$; the single-seed diagnostics show how stability and utility change as the weight varies. DynaHate FPED is non-discriminative because defined group FPR comparisons are not supported by the available primary-group field in this split. Bold marks the best value within each dataset block, with lower values preferred for error and stability metrics.

Dataset	Backbone	L-CFR↓	L-CTFG↓	S-CFR↓	S-CTFG↓
HateXplain	BERT-base-uncased	**0.1115**	**0.0735**	**0.0560**	**0.0362**
	RoBERTa-base	0.1221	0.0748	0.0572	0.0380
	DeBERTa-V3-base	0.2706	0.2449	0.0754	0.0649
ToxiGen	BERT-base-uncased	0.0483	0.0398	0.0311	0.0236
	RoBERTa-base	0.0519	0.0423	0.0435	0.0273
	DeBERTa-V3-base	**0.0412**	**0.0369**	**0.0249**	**0.0205**
DynaHate	BERT-base-uncased	0.1091	**0.0676**	0.0770	0.0421
	RoBERTa-base	**0.1005**	0.0684	0.0682	0.0473
	DeBERTa-V3-base	0.1298	0.1156	**0.0146**	**0.0124**

Table 15: Source-specific counterfactual stability for the same LLM-CLP backbone checkpoints. L-CFR/L-CTFG are evaluated on LLM-CF test pairs, while S-CFR/S-CTFG are evaluated on deterministic swap-CF test pairs. Values are three-seed means; bold marks the best backbone within each dataset and metric.

Tables 13–15 support two additional observations. First, DeBERTa-V3-base is the strongest LLM-CLP backbone on the ordinary task metrics across all three datasets, which justifies using it in the controlled main comparison. Second, the preferred backbone can depend on the counterfactual evaluation source: DeBERTa-V3-base gives the best swap-CF stability on ToxiGen and DynaHate, whereas the LLM-CF stability columns sometimes favor BERT-base-uncased or RoBERTa-base. The FPED/FNED columns are retained in the backbone diagnostics to show the same evaluation surface as the main results; for DynaHate, however, the all-zero FPED column is an artifact of unavailable defined group FPR comparisons rather than a substantive fairness result. This source-dependent ranking is another reason to keep LLM-CF diagnostics separate from the shared swap-CF benchmark used for the main method comparison.

D LLM Counterfactual Quality Audit

We conduct an author-team audit of 300 randomly sampled LLM-generated counterfactual pairs (100 per dataset), stratified by original label (50 toxic and 50 non-toxic per dataset). Author-team members independently apply the same rubric to each sampled pair, and the final usability label is assigned by majority vote. Because the annotators are not external to the project, we use this audit as a diagnostic validity check rather than as a definitive human evaluation of counterfactual quality. The audit records include pair identifiers, source texts, rewrites, edit statistics, Boolean quality labels, final usability labels, and failure-mode tags. Table 16 reports the aggregate quality labels, Table 17 breaks down failure modes, and Table 18 provides representative checked cases. The majority-vote annotations show a stricter result than the automatic generation filters alone would suggest: most label and fluency checks pass, but successful identity replacement is less reliable, especially on ToxiGen and DynaHate.

Table 16: Author-team LLM counterfactual quality audit. Each dataset has 100 audited pairs. Usable (majority) is the final majority-vote usability decision; usable (recomputed checks) is the deterministic conjunction of the Boolean checks.

Criterion	HateXplain	ToxiGen	DynaHate
Identity changed	79%	58%	63%
Label preserved	94%	100%	100%
No semantic drift	94%	94%	100%
No toxicity drift	96%	99%	98%
Fluent	98%	94%	92%
No cultural asymmetry	100%	98%	100%
Usable (majority)	**71%**	**55%**	**59%**
Usable (recomputed checks)	71%	55%	60%

Table 17 breaks down non-usable cases by the multi-label failure-mode tags in the author-team audit annotations. Because one row can have multiple tags, the failure-mode counts need not sum to the total number of non-usable rows. The largest issue is missed or insufficient identity substitution: it appears in 100 of the 115 majority-vote non-usable pairs. Semantic drift is present but is not the dominant failure mode in the author-team annotations.

Table 17: Failure-mode counts in the author-team audit. Counts are multi-label over non-usable pairs, so columns can sum above the total non-usable count.

Failure mode	**HateXplain**	**ToxiGen**	**DynaHate**
No identity change	21	42	37
Label not preserved	6	0	0
Semantic drift	6	6	0
Toxicity drift	4	1	2
Awkward rewrite	2	6	8
Cultural asymmetry	0	2	0
Total non-usable	29	45	41

Table 18: Representative audited LLM counterfactual cases. Usable rows illustrate label-preserving identity substitution; non-usable rows illustrate common failure modes.

Type	**Original**	**LLM counterfactual**
Usable / HX	a real piece of work also bending over for the arabs metruth ausfam	a real piece of work also bending over for the jews metruth ausfam
Usable / TG	Islam is not a religion, it is a political ideology, and it is the only ideology on Earth that will kill you if you leave it	Judaism is not a religion, it is a political ideology, and it is the only ideology on Earth that will kill you if you leave it
No identity change	i hate the BNP	I hate the BNP
Semantic drift	is it me or are bin men the most retarded people on earth	is it me or are black men the most ignorant group in society
Toxicity drift	another typical nigger	another typical black person
Awkward rewrite	All women respect is the fist	All females respect is the first

Overall, the author-team audit sharpens the paper's main interpretation. The LLM counterfactuals often preserve labels and fluency, but 62% of the audited pairs receive a final majority-vote usable label. This supports using them conservatively as noisy CLP pairing signals, where cross-entropy is still applied only to the original text, but it does not support treating the generated rewrites as gold causal counterfactuals or as an evaluation oracle. Future work should add automatic identity-change checks, semantic-similarity filters, and human spot checks before expanding the role of generated pairs.