

# Neutralising Manipulation in Ukrainian News with Iterative Score-Guided LLM Rewriting\*

Vladyslav Hromenko<sup>1,\*†</sup>, Viktor Shevchenko<sup>1,†</sup>, Leonid Artiushyn<sup>2,†</sup>, Vladislav Shostak<sup>2,†</sup>

<sup>1</sup> Institute of Software Systems National Academy of Sciences of Ukraine, 40 Academician Glushkov Avenue, building 5, Kyiv, 03187, Ukraine

<sup>2</sup> State Research Institute of Aviation, 6 Kazarmenna Str., Kyiv, Ukraine, 01135

## Abstract

Automated detection of manipulation techniques in Ukrainian news has received growing attention, yet one natural downstream application of detection systems – automatically reducing detected manipulation while preserving factual content – remains comparatively unaddressed. This paper presents an iterative score-guided pipeline for manipulation reduction in Ukrainian-language Telegram posts, formulated as a constrained rewriting optimisation problem: given an article and its detected manipulation technique labels, a writer LLM is iteratively prompted to produce a neutralised rewrite that minimises residual manipulation while satisfying hard lower bounds on entity preservation and semantic similarity. Self-assessment is replaced by three independent automatic scorers – LoRA-adapted MamayLM-Gemma-3-4B-IT multilabel manipulation classifier, Entity Preservation Scorer with fuzzy morphological matching, and Ukrainian sentence-transformer semantic similarity scorer – whose outputs are converted into structured natural-language critiques fed back into each subsequent generation step, with the loop terminated by a formally defined quality gate. Three entity preservation strategies of increasing structural commitment are proposed and comparatively evaluated on a 204-article stratified sample from the UNLP 2025 Shared Task dataset: naïve (unconstrained) rewriting, entity-grounded rewriting with a deterministic reconciliation attempt, and placeholder-based rewriting in which all named entities are substituted with typed tokens before generation and restored afterwards. The placeholder-based mode achieves the best entity preservation (EPS-F1 0.822) and the highest quality-gate pass rate (33.82%), while the naïve mode achieves the strongest manipulation reduction (MRS 0.684) at the cost of entity drift. A comparison of three writer LLMs reveals that the domain-adapted MamayLM-Gemma-3-12B-IT FP8, despite being 2.25× smaller than the 27B-parameter alternatives, outperforms both Gemma 3 27B and Qwen 3.5 27B on manipulation reduction while requiring only a single L4 GPU, suggesting that the combination of Ukrainian adaptation and the iterative pipeline is sufficient to outperform larger off-the-shelf models.

## Keywords

manipulation reduction, social media manipulation, natural language processing, large language models, constrained text rewriting, score-guided text generation

## 1. Introduction

The information environment surrounding the ongoing war in Ukraine has placed an extraordinary strain on the integrity of public discourse. Ukrainian-language Telegram channels have become a primary vector for manipulative content, employing techniques such as loaded language, appeals to fear, whataboutism, and cherry-picking to shape audience perceptions [1, 2]. Automated detection of such techniques has emerged as an active research priority, exemplified by the UNLP 2025 Shared Task on Detecting Social Media Manipulation [1]. Detection systems, however, address only one half of the problem: they identify that a text is manipulative but do not offer a concrete, actionable alternative with the manipulation removed and the content preserved.

---

2nd Workshop “Intelligent information technologies” UkrProg-IIT’2026 co-located with 16th International Scientific and Practical Programming Conference UkrPROG’2026, May 5-6, 2026, Kyiv, Ukraine

\*Corresponding author.

† These authors contributed equally.

✉ [gromvlad12@gmail.com](mailto:gromvlad12@gmail.com) (Vladyslav Hromenko); [gi2014@ukr.net](mailto:gi2014@ukr.net) (Viktor Shevchenko); [artleonid2017@gmail.com](mailto:artleonid2017@gmail.com) (Leonid Artiushyn); [alkononov@gmail.com](mailto:alkononov@gmail.com) (Vladislav Shostak)

🆔 0009-0001-5285-9912 (Vladyslav Hromenko); 0000-0002-9457-7454 (Viktor Shevchenko); 0000-0002-7488-7244 (Leonid Artiushyn); 0000-0002-2956-1069 (Vladislav Shostak)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

This paper addresses the complementary downstream task of manipulation reduction: given a Ukrainian news text and detected technique labels, automatically producing a rewritten version that neutralises the identified techniques while preserving entities, numerical data, and semantic content. Manipulation manifests as a heterogeneous set of ten distinct pragmatic strategies, each requiring a qualitatively different neutralisation action, which – combined with the imperative to avoid entity hallucination in a morphologically rich language – motivates an iterative approach in which external scorers provide structured feedback after each generation step. Self-assessment is replaced with three independent automatic scorers: manipulation classifier, a spaCy NER module for entity preservation, and a sentence-transformer semantic similarity scorer, terminating the loop via a formally defined quality gate.

This paper makes four contributions: (1) manipulation reduction is formulated as a constrained rewriting optimisation problem instantiated as an iterative score-guided process with multi-dimensional scoring; (2) three entity preservation strategies – naïve, entity grounded, and placeholder-and-restore – are proposed and their trade-offs empirically characterised; (3) a reference-free evaluation framework is introduced combining manipulation reduction score, NER-based entity preservation score and cosine semantic similarity; (4) an empirical comparison is conducted across five experimental configurations on a 204-article stratified sample from the UNLP 2025 dataset [1].

## **2. Related work**

### **2.1. Manipulation and Propaganda Detection**

The computational study of propaganda detection has evolved from English span-level identification [8] to comprehensive multilingual frameworks [6, 9, 10]. For Ukrainian specifically, prior to the establishment of a dedicated benchmark, automated propaganda detection was approached through binary and coarse multi-class classification – distinguishing propaganda, non-propaganda, and suspicious content – using ensembles of recurrent neural network architectures [7]. The UNLP 2025 Shared Task [1] then established the first dedicated Ukrainian benchmark, targeting a 10-technique taxonomy specific to wartime Telegram discourse; top-performing systems in that shared task effectively leverage fine-tuned LLMs and LoRA adaptations [2, 11]. Despite this progress, the literature has focused exclusively on identifying manipulation – the downstream task of actively reducing or neutralising detected techniques remains mostly unaddressed.

### **2.2. Text Style Transfer and Bias Neutralisation**

Manipulation reduction bears a methodological resemblance to text style transfer (TST) and subjective bias neutralisation [3, 4, 16], which balance style accuracy with content preservation. In both TST and manipulation reduction, unconstrained optimisation inherently risks degenerate outcomes where models achieve high style transfer (or near-zero manipulation scores) by trivially deleting content or generating unrelated neutral text [16]. However, manipulation reduction diverges from standard bias neutralisation in three critical respects. First, manipulation is not a monolithic stylistic attribute; it comprises distinct pragmatic strategies requiring varied interventions, such as lexical substitution for loaded language versus complete argument deletion for whataboutism. Second, neutralisation must maintain coherence across full-length document posts rather than isolated parallel sentences. Finally, Ukrainian's seven-case nominal declension system makes entity preservation substantially harder than in analytical languages like English.

### **2.3. Entity Preservation in Generative Rewriting**

Entity-level hallucination – generating incorrect, dropped, or distorted named entities – is a primary failure mode in constrained generative rewriting [12, 13]. While NER-based evaluation and entity masking are established mitigation strategies for English text, their application to Ukrainian

introduces novel challenges. Specifically, traditional exact-string replacement fails due to morphological variations (e.g., nominative "Бахмут" (Bakhmut) vs. genitive "Бахму́ту" (Bakhmutu)), necessitating the fuzzy matching and entity grounded strategies used in this work.

## 2.4. Iterative LLM Refinement

Inference-time iterative refinement – where an LLM generates, evaluates, and revises its output – significantly improves generation quality without additional training [5, 14, 15]. Our approach builds on frameworks like Self-Refine [5] but departs from them in two fundamental ways. Rather than relying on the LLM’s potentially unreliable self-critique, our pipeline utilizes three independent, quantitative external scorers (classifier, NER module, semantic embedder) mapped via a score-to-critique bridge. Additionally, iterations are terminated by a formally defined quality gate, utilizing a conjunction of hard metric thresholds rather than relying on the LLM’s ad-hoc judgment.

## 3. Dataset

The experiments reported in this paper were conducted on data drawn from the UNLP 2025 Shared Task on Detecting Social Media Manipulation [1]. The full benchmark dataset consists of 9,557 posts collected from Ukrainian Telegram news and political blog channels, manually annotated for the presence of 10 manipulation techniques by experienced journalists, editors, and media analysts from the Ukrainian data journalism organisation Texty.org.ua [1]. The annotation taxonomy – developed through focus groups of Ukrainian media professionals and refined via discussion of hard cases [1] – comprises the following techniques: appeal to fear, bandwagon (appeal to the people), cherry-picking (selective presentation of facts), cliché, euphoria, FUD (fear/uncertainty/doubt), glittering generalities, loaded language, straw man, and whataboutism. This taxonomy was specifically adapted to the Ukrainian media landscape and includes categories such as euphoria and glittering generalities that are not present in the SemEval-2020 taxonomy [6] but are characteristic of the rhetorical patterns observed in Ukrainian wartime Telegram discourse [1]. The dataset is released under a CC BY-NC-SA 4.0 licence.

The annotation is multilabel: a single post may simultaneously exhibit multiple manipulation techniques. This property is central to the task, as manipulation in real-world texts typically combines several strategies within the same passage – for instance, loaded language paired with appeal to fear, or cherry-picking combined with glittering generalities.

For the experiments in this paper, a stratified random sample of 204 articles is drawn from the full dataset, preserving the original label distribution. The technique frequency within this experimental sample, measured by the number of articles in which each technique is detected with classifier probability exceeding 0.5, is as follows: loaded language (152), cliché (87), glittering generalities (69), euphoria (50), cherry-picking (31), FUD (25), whataboutism (17), appeal to fear (16), bandwagon (15), and straw man (6). The distribution is heavily skewed, with loaded language appearing in approximately 75% of sampled articles, while the straw man appears in fewer than 3%. The mean manipulation score sum across the sample is 2.36, indicating that the average article triggers approximately two to three techniques above the detection threshold.

## 4. Task Formulation

The manipulation reduction task is formally defined as a constrained rewriting optimisation problem. Given an article  $x$  with a detected manipulation technique label set  $L \subseteq 1, \dots, 10$ , the goal is to produce a rewritten version  $x^*$  that minimises the residual manipulation score while satisfying hard constraints on entity preservation and semantic fidelity in Equation 1:

$$x^* = \arg \min_{x'} \text{ManipSum}(x') \quad \text{s.t.} \quad \text{EPS}_{\text{recall}}(x, x') \geq \theta_F, \quad \text{CosSim}(x, x') \geq \theta_S \quad (1)$$

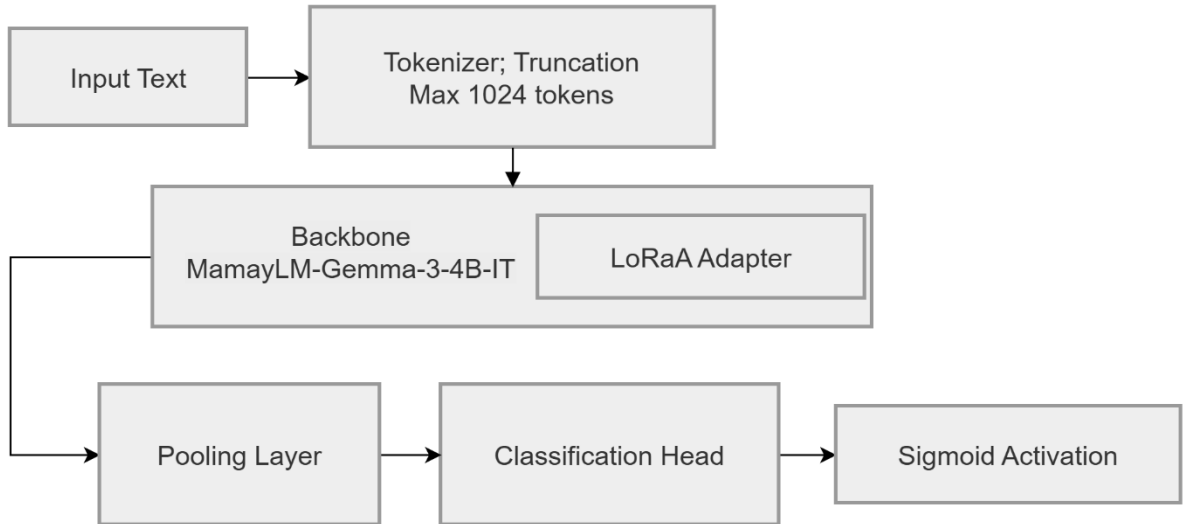
where  $\text{ManipSum}(x') = \sum_{t=1}^{10} p_t(x')$  is the sum of the manipulation classifier's sigmoid probabilities across all ten techniques,  $\text{EPS}_{\text{recall}}(x, x')$  is the named-entity recall between original and rewritten text, and  $\text{CosSim}(x, x')$  is the cosine similarity between the sentence embeddings of  $x$  and  $x'$ .

The constrained formulation is essential because the optimisation landscape is non-differentiable – the manipulation classifier, NER-based entity preservation scorer, and embedding model are all discrete black-box components – and because the rewriting function is implemented by an autoregressive LLM, direct gradient-based optimisation is not applicable. This challenge is well established in the text generation literature: as demonstrated in foundational work on sequence-level training [17], metrics such as BLEU and ROUGE are non-differentiable, necessitating either reinforcement learning or iterative approximate methods. In the present setting, the problem is further compounded by the multi-objective nature of the constraints (manipulation reduction, entity recall, and semantic similarity must be satisfied simultaneously). The problem is therefore solved approximately through the iterative score-guided pipeline, which treats the LLM as a stochastic rewriting operator and uses structured critique from the three scoring modules to guide successive approximations toward the feasible region.

## 5. Manipulation detector

The manipulation detector acts as a fixed, pre-trained scoring component within the pipeline and is not updated during the iterative rewriting process. While its full development and evaluation fall outside the scope of this work, a brief overview of its architecture is necessary to understand the scoring mechanism.

The detector is a multilabel classifier trained to identify the 10 manipulation techniques from the UNLP 2025 taxonomy, visualized in Figure 1.



**Figure 1:** Manipulation techniques detector architecture.

It utilizes a backbone-pooler-head architecture:

- **Backbone:** A MamayLM-Gemma-3-4B-IT [18] model fine-tuned via LoRA [19] and loaded in 4-bit NF4 quantization for efficient single-GPU deployment.
- **Pooler:** Computes a masked mean over the final layer's token-level hidden states to produce a fixed-dimensional document representation.
- **Classification Head:** A single-layer MLP that maps the pooled representation to 10 output logits.

During inference, inputs are truncated at 1024 tokens. In our sample no articles exceed this limit, so all reported scores are computed on full article content. The model outputs 10 raw logits, which are converted to calibrated probabilities  $p_t \in [0,1]$  via a sigmoid function. A technique is considered active if  $p_t > 0.5$ .

## 6. Pipeline architecture

### 6.1. Overview and Loop Structure

The pipeline operates as a detect–rewrite–score–critique loop that iteratively refines a manipulative article toward a neutral version satisfying three simultaneous quality constraints. The main steps of the pipeline are described in Algorithm 1.

At each iteration  $i \in \{1, \dots, N\}$  with  $N = 5$ , the following steps are executed sequentially. A writer prompt is assembled from the original article (or its placeholder-substituted form), the detected technique labels with technique-specific neutralisation actions, and – for iterations  $i > 1$  – the structured critique from the previous cycle and the previous rewrite. The Writer LLM produces a candidate rewrite, extracted via XML tag parsing (`<rewrite>...</rewrite>`). If the placeholder mode is active, typed tokens are restored to original entity strings immediately after extraction. If the entity-grounded mode is active, the entity reconciler performs deterministic post-generation correction. Three independent scorers then evaluate the current version: the Manipulation Scorer (per-technique sigmoid probabilities), the Entity Preservation Scorer (NER-based entity recall), and the Semantic Similarity scorer (cosine similarity of sentence embeddings). Their outputs are converted into a structured natural-language critique in Ukrainian. The iteration terminates early if the quality gate is satisfied like described in Equation 2.

---

#### Algorithm 1 Iterative Score-Guided Manipulation Reduction Pipeline

---

**Require:** Article  $x$ , detected label set  $L \subseteq \{1, \dots, 10\}$ , scorers  $\mathcal{M}$ ,  $\mathcal{F}$ ,  $\mathcal{S}$ , mode  $\in \{\text{naive, grounded, placeholder}\}$

**Ensure:** Best rewrite  $x^*$ , iteration history  $\mathcal{H}$

```

1:  $S^{\text{orig}} \leftarrow \mathcal{M}.\text{score}(x).\text{ManipSum}$  ▷ Baseline manipulation scores
2: if mode = placeholder then
3:    $\hat{x}, \mathcal{P} \leftarrow \text{PlaceholderReplace}(x)$  ▷ Build placeholder map once; reuse across iterations
4: else
5:    $\hat{x} \leftarrow x$ 
6: end if
7:  $x'_0 \leftarrow \emptyset$ ;  $c_0 \leftarrow \emptyset$ ;  $\mathcal{H} \leftarrow []$ 
8: for  $i = 1$  to  $N$  do
9:    $\pi \leftarrow \text{BuildPrompt}(\hat{x}, L, x'_{i-1}, c_{i-1}, \text{mode})$  ▷ Roster injected if mode = grounded
10:   $\tilde{x}'_i \leftarrow \text{WriterLLM}(\pi)$  ▷ Generate candidate rewrite
11:  if mode = placeholder then
12:     $x'_i \leftarrow \mathcal{P}.\text{restore}(\tilde{x}'_i)$  ▷ Replace placeholders with originals
13:  else
14:     $x'_i \leftarrow \tilde{x}'_i$ 
15:  end if
16:  if mode = grounded then
17:     $x'_i, r_i \leftarrow \text{EntityReconciler}(x, x'_i)$  ▷ Exact → fuzzy → escalate
18:  end if
19:   $m_i \leftarrow \mathcal{M}.\text{score}(x'_i)$  ▷ Per-technique sigmoid probabilities
20:   $f_i \leftarrow \mathcal{F}.\text{score}(x, x'_i)$  ▷ NER entity recall, precision, F1
21:   $s_i \leftarrow \mathcal{S}.\text{score}(x, x'_i)$  ▷ Cosine similarity of sentence embeddings
22:   $c_i \leftarrow \text{BuildCritique}(m_i, f_i, s_i, S^{\text{orig}}, \mathcal{P} \text{ if mode=placeholder else } \emptyset)$  ▷ Structured feedback
23:   $\mathcal{H}.\text{append}((x'_i, m_i, f_i, s_i, c_i))$ 
24:  if  $m_i.\text{ManipSum} \leq \tau_M \wedge f_i.\text{recall} \geq \theta_F \wedge s_i \geq \theta_S$  then
25:    break ▷ Quality gate satisfied
26:  end if
27: end for
28:  $i^* \leftarrow \arg \max_i \left[ \alpha \cdot \frac{S^{\text{orig}} - m_i.\text{ManipSum}}{S^{\text{orig}}} + \beta \cdot f_i.\text{recall} + \gamma \cdot s_i \right]$  ▷ Best-of- $N$ ;  $\alpha=0.4, \beta=0.3, \gamma=0.3$ 
29: return  $x'_{i^*}, \mathcal{H}$ 

```

---

$$\text{ManipSum}(x'_i) \leq \tau_M \wedge \text{EPS}_{\text{recall}}(x, x'_i) \geq \theta_F \wedge \text{CosSim}(x, x'_i) \geq \theta_S \quad (2)$$

*with  $\tau_M = 1.2, \theta_F = 0.85, \theta_S = 0.80$ .*

where  $\text{ManipSum}(x'_i)$  is the sum of sigmoid probabilities across all ten techniques for the rewrite at iteration  $i$ ,  $\text{EPS}_{\text{recall}}(x, x'_i)$  is the named-entity recall between the original and the rewrite, and  $\text{CosSim}(x, x'_i)$  is the cosine similarity.

The threshold  $\tau_M = 1.2$  corresponds approximately to one active technique remaining at full probability, representing a substantial but not necessarily complete reduction from the sample mean of 2.36;  $\theta_F = 0.85$  requires that at least 85% of source entities are preserved in the rewrite; and  $\theta_S = 0.80$  guards against semantic drift while leaving sufficient room for meaningful lexical reformulation. The primary objective of the quality gate is to minimize redundant iterations, thereby increasing the overall efficiency of the pipeline.

## 6.2. Prompt Architecture

The prompt design follows five principles, each addressing a specific failure mode observed during development. First, the system prompt is divided into three explicit separation of instruction blocks – `<role>`, `<constraints>`, and `<task>` – to reduce instruction intent misalignment [24]. The `<constraints>` block specifies four hard constraints, with an optional fifth injected only when the placeholder mode is active: C1 (no addition of facts, names, dates, or numbers absent from the original); C2 (no truncation: paragraph and sentence count must remain identical); C3 (no modifications of proper nouns, numerical data, or factual quotations); C4 (no commentary: output must consist solely of rewritten text within `<rewrite>` tags); and C5 (placeholder preservation: typed placeholder tokens must be copied verbatim). C5 is excluded from non-placeholder runs, as including inapplicable constraints has been observed to degrade compliance on the constraints that are relevant.

Second, rather than providing a generic neutralisation instruction, the prompt supplies technique-specific action specifications with concrete Ukrainian before/after examples for each detected technique. For instance, the action for loaded language instructs the replacement of evaluative epithets with neutral descriptors, illustrated by the example "злочинний режим" ("criminal regime") to "керівництво країни" ("leadership of the country").

Third, for iterations  $i > 1$ , the prompt is restructured to lead with a constraint reminder anchor followed by the structured score critique, the previous rewrite, and the original article. This correction-first ordering reduces the likelihood that the model reproduces errors from the previous version. Fourth, when the placeholder mode is active, a placeholder rules block lists all active typed tokens and provides an explicit copy-through instruction with a positive and two negative examples. Fifth, the format instruction appears exactly once as the final line of the user turn, ensuring consistent extraction via regex parsing, with stop sequences configured to prevent non-article content.

## 6.3. Mode A – Naïve Rewriting

The naïve mode serves as the ablation baseline. No entity protection mechanism is active: neither prompt-level roster injection, nor post-generation reconciliation, nor pre-generation placeholder substitution. Score critiques from the three scorers are assembled and injected into each subsequent iteration's prompt, but entity preservation relies entirely on the LLM's compliance with constraints. This mode isolates the contribution of the iterative loop itself from the contribution of the entity preservation strategies.

---

**Algorithm 2** Entity Reconciliation (Entity-Grounded Mode)

---

**Require:** Original article  $x$ , candidate rewrite  $\tilde{x}'_i$ , fuzzy threshold  $\delta = 88$ , failures from previous iteration  $F_{i-1}$

**Ensure:** Reconciled rewrite  $x'_i$ , failure list  $F_i$

```
1:  $\mathcal{E} \leftarrow \text{ExtractEntities}(x)$  ▷ Named entities + numeric tokens from original
2:  $x'_i \leftarrow \tilde{x}'_i$ ;  $F_i \leftarrow []$ 
3: for each  $(e_{\text{norm}}, e_{\text{canon}}) \in \mathcal{E} \cup F_{i-1}$  do
4:   if  $e_{\text{canon}} \in x'_i$  then ▷ Stage 1: Exact match – skip
5:     continue
6:   end if
7:    $\mathcal{E}' \leftarrow \text{ExtractEntities}(x'_i)$  ▷ Stage 2: Fuzzy substitution
8:    $e^* \leftarrow \arg \max_{e' \in \mathcal{E}'} \text{Lev}_{\text{sort}}(e_{\text{norm}}, e')$ 
9:   if  $\text{Lev}_{\text{sort}}(e_{\text{norm}}, e^*) \geq \delta$  then
10:     $x'_i \leftarrow x'_i.\text{replace}(e^*, e_{\text{canon}}, \text{count}=1)$ 
11:   else ▷ Stage 3: Escalate to next iteration
12:     $F_i.\text{append}(e_{\text{canon}})$ 
13:   end if
14: end for
15: return  $x'_i, F_i$ 
```

$\text{Lev}_{\text{sort}}(a, b)$ : normalised Levenshtein similarity ratio computed on token-sorted forms of  $a$  and  $b$ , scaled to  $[0, 100]$ .

---

#### 6.4. Mode B – Entity-Grounded Rewriting

The entity-grounded mode augments the iterative loop with NER-based entity grounding at two points. Before generation, a protected entities block listing all persons, organisations, locations, and numbers extracted from the original by the spaCy pipeline is injected into the writer prompt. After generation, the entity reconciler applies a three-stage deterministic correction cascade to the rewrite, as illustrated in Algorithm 2. Stage 1 checks for exact-match presence and skips the entity if it is already present verbatim in the rewrite. Stage 2 applies fuzzy substitution using a normalised Levenshtein similarity ratio computed on the token-sorted forms of both strings [25], with a matching threshold of 88 – deliberately set lower than the EPS scoring threshold of 92 – to capture morphological case inflections characteristic of Ukrainian nominal declension (e.g., nominative "Бахмут" (Bakhmut) vs. genitive "Бахмуту" (Bakhmutu)). Stage 3 escalates unresolvable entities to the next iteration's prompt via a reconciliation failures block. Previously escalated entities – those that could not be resolved in iteration  $i - 1$  – are carried forward as  $F_{i-1}$  and re-checked in the current iteration, giving the reconciler an additional opportunity to recover entities that the LLM was explicitly prompted to include but still omitted. The entity reconciler shares the spaCy model instance with the Entity Preservation Scorer, requiring no additional GPU memory.

#### 6.5. Mode C – Placeholder-Based Rewriting

The placeholder-based mode eliminates entity hallucination by removing all real entity strings from the LLM's input. All NER-tagged spans are replaced with typed placeholders of the form  $\langle \text{TYPE\_INDEX} \rangle$  (e.g.,  $\langle \text{PER\_1} \rangle$ ,  $\langle \text{ORG\_2} \rangle$ ,  $\langle \text{LOC\_1} \rangle$ ,  $\langle \text{NUM\_3} \rangle$ ). Overlapping spans are resolved by retaining the longer span; repeated occurrences of the same surface form receive the same placeholder; replacement proceeds longest-first to prevent partial substitutions; and a position-aware sweep catches surface-form repetitions that the NER model tagged only once. The placeholder map – a bidirectional dictionary between typed tokens and original strings – is computed once per article and reused across all iterations.

After rewrite extraction, placeholders are restored to their original entity strings via exact replacement, with a fuzzy regex fallback for slightly malformed tokens (e.g., extra whitespace within angle brackets). Any placeholders the LLM dropped entirely are detected and reported for downstream correction.

## 6.6. Best-of-N Selection

Because metric quality does not necessarily improve monotonically across iterations, the final output is selected from the full iteration history by maximising a composite score, described in Equation 3.

$$i^* = \arg \max_{i \in 1, \dots, k} \left[ \alpha \cdot \frac{\text{ManipSum}^{\text{orig}} - \text{ManipSum}'_i}{\text{ManipSum}^{\text{orig}}} + \beta \cdot \text{EPS}_{\text{recall}}(x, x'_i) + \gamma \cdot \text{CosSim}(x, x'_i) \right] \quad (3)$$

$$\alpha = 0.4, \beta = 0.3, \gamma = 0.3$$

where  $\text{ManipSum}^{\text{orig}}$ ,  $\text{ManipSum}'_i$  is the sum of sigmoid probabilities across all ten manipulation techniques for the original article and rewritten article on step  $i$ , respectively. The manipulation reduction term is normalised by  $\text{ManipSum}^{\text{orig}}$  to make it scale-invariant across articles with different baseline scores. If the quality gate from Equation 2 is satisfied at some iteration, that iteration is not guaranteed to be selected: a later iteration that further improves one dimension without regressing on the others will score higher on the composite. The manipulation reduction term is assigned the highest weight ( $\alpha = 0.4$ ) to reflect that neutralisation is the primary objective. Unlike the quality gate, which treats all three dimensions as hard binary constraints of equal standing, the Best-of-N composite encodes a continuous preference ordering among iterations.

## 7. Evaluation Framework

The rewriting pipeline is evaluated via residual manipulation, entity drift, and semantic drift. Because no gold-standard neutralised rewrites exist for the UNLP 2025 corpus, the task is treated as a constrained optimisation problem rather than a sequence-to-sequence mapping. This reference-free approach avoids the prohibitive cost of expert annotation and accounts for the fact that multiple distinct rewrites can be equally valid. Evaluating based on measurable objectives rather than an N-gram overlap target is structurally similar to recent reference-free faithfulness evaluations using NLI and atomic-fact enumeration [26, 27, 28].

### 7.1. Manipulation Reduction Score (MRS)

MRS from Equation 4 quantifies the removal of manipulative content. To avoid diluting the score by averaging over absent techniques, MRS is computed as the absolute probability reduction macro-averaged only over techniques detected in the original text, yielding a conceptually related per-dimension reduction metric to those used successfully in automatic subjective-bias neutralisation [3]. Building upon the probabilities introduced during the task formulation, the score calculates the difference between the original and rewritten posterior probabilities for active techniques, as formalized in Equation 4.

$$\text{MRS}(x, x') = \frac{1}{|T_d|} \sum_{t \in T_d} (p_t^{\text{orig}} - p_t^{\text{rew}}), \quad T_d = \{t \in T : p_t^{\text{orig}} > \theta_{\text{det}}\} \quad (4)$$

where  $x$  and  $x'$  denote the original and rewritten texts,  $T_d$  represents the subset of detected manipulation techniques from the full taxonomy  $T$ ,  $p_t^{\text{orig}}$  and  $p_t^{\text{rew}}$  are the probabilities assigned by the detector to technique  $t$  for the original and rewritten texts respectively, and  $\theta_{\text{det}} = 0.5$  is the detection threshold. Because  $T_d$  contains only techniques with  $p_t^{\text{orig}} > 0.5$ , MRS is bounded in  $(-0.5, 1]$ , where positive values indicate net probability reduction and negative values indicate that the rewrite increased the classifier's probability for at least one originally-detected technique — i.e., reinforcement of an existing technique rather than its removal. Inputs with empty detection sets are excluded. Because the score uses the same detector that defines the targets, it operates as a relative self-consistency metric across configurations rather than an absolute estimator [20].

## 7.2. Entity Preservation Score (EPS)

EPS measures entity-level drift between the original and rewritten text through named-entity overlap, focusing on entity recall because it correlates more strongly with human judgements of factual consistency than token-level overlap [21]. Entity substitutions or deletions account for a substantial share of factual errors in abstractive generation [22]. Entities are extracted using the spaCy NER backbone [23] alongside regex-captured numeric tokens, and their strings are normalised to account for Telegram-sourced text artifacts. To accommodate Ukrainian's rich morphology, entities are matched using fuzzy equivalence via token-sort ratio, a normalised variant of Levenshtein distance [25] where equivalence requires a score of 92 or higher.

Given the normalised entity sets for the original and rewritten text, three disjoint sets are defined representing preserved (P), missing (M), and hallucinated (H) entities, as outlined in Equation 5:

$$P = \{e \in E(x) : \exists e' \in E(x'), \text{Lev}_{\text{sort}}(e, e') \geq \theta_{EPS}\}; M = E(x) \setminus P; \\ H = \{e' \in E(x') : \forall e \in E(x), \text{Lev}_{\text{sort}}(e, e') < \theta_{EPS}\} \quad (5)$$

where  $\theta_{EPS} = 92$ ,  $E(x)$  and  $E(x')$  are the normalised sets of extracted entities from the original and rewritten texts,  $e$  and  $e'$  are individual entity strings within those sets, and  $\text{Lev}_{\text{sort}}(a, b)$  defined in Algorithm 2.

These sets produce the primary metrics for recall, precision, and F1 score, expanding upon the recall constraint defined in the optimisation problem formulation, as shown in Equation 6.

$$\text{EPS}_{\text{recall}} = \frac{|P|}{|E(x)|}, \text{EPS}_{\text{precision}} = \frac{|P|}{|P| + |H|}, \text{EPS}_{\text{F1}} = \frac{2 \text{EPS}_{\text{precision}} \text{EPS}_{\text{recall}}}{\text{EPS}_{\text{precision}} + \text{EPS}_{\text{recall}}} \quad (6)$$

where  $|P|$ ,  $|H|$ , and  $|E(x)|$  represent the absolute counts of the preserved entity set, the hallucinated entity set, and the original entity set, respectively. The recall metric serves as the primary entity preservation signal. For articles where the rewrite contains no extracted entities ( $|P| + |H| = 0$ ), precision also defaults to 1.0 [21]; when  $|E(x)| = 0$  (no entities in the original), EPS-recall is set to 1.0; articles where the original contains no entities but the rewrite introduces new ones are handled by the hallucination set H alone, which in such cases contributes only to precision degradation without affecting recall.

## 7.3. Semantic Similarity

Semantic drift is measured via cosine similarity of L2-normalized sentence embeddings. This process utilizes a 278-million-parameter variant of the multilingual MPNet architecture fine-tuned on Ukrainian paraphrase data [29, 30], as document-level cosine similarity on Siamese sentence encoders is established as a reliable proxy for semantic equivalence [31]. The calculation follows the cosine similarity function for document-level embeddings established alongside the task constraints, detailed in Equation 7.

$$\text{SimScore}(x, x') = \cos(\phi(x), \phi(x')) = \phi(x)^\top \phi(x') \quad (7)$$

where  $\phi(x)$  and  $\phi(x')$  represent the L2-normalized 768-dimensional document-level embedding vectors of the original text and the rewritten text, respectively. Unlike the previous two metrics, higher is not strictly better. The pipeline targets a mid-range band between 0.8 and 0.9. Scores above 0.95 suggest under-rewriting by failing to remove manipulation, while scores below 0.6 indicate severe semantic drift and potential content distortion.

# 8. Experiments and Results

## 8.1. Experimental Setup

All experiments are conducted on the 204-article stratified sample described in Section 3. Every configuration processes the identical article set with a fixed random seed and inference parameters.

Three writer LLMs are evaluated. MamayLM-Gemma-3-12B-IT FP8 [32] is a 12-billion-parameter Ukrainian-adapted instruction-tuned model, quantised to FP8 static precision, built on the INSAIT MamayLM-Gemma-3-12B-IT-v1.0 base [33]. It serves as the primary writer for the three pipeline-mode experiments (Modes A–C). Gemma 3 27B IT [34] is Google's 27-billion-parameter instruction-tuned dense model, served in BF16 native precision. Qwen 3.5 27B [35, 36] is a 27-billion-parameter dense model from Alibaba, also served in BF16; Qwen's thinking mode is explicitly disabled. All models are served through a local vLLM [37] OpenAI-compatible API server.

MamayLM-12B FP8 fits within 22 GB on a single NVIDIA L4 GPU (24 GB VRAM), making it accessible on cost-efficient cloud instances. Gemma 3 27B IT and Qwen 3.5 27B BF16 require an NVIDIA A100 80 GB GPU. Throughput measurements, reported in Table 1 as wall-clock seconds per iteration averaged over the 204-article run, reflect the full iteration cost including prompt construction, generation and restoration (where applicable), and all three scoring passes.

**Table 1**

Hardware footprint and mean wall-clock time per pipeline execution.

| Model                  | Precision  | GPU         | VRAM used | Speed (s/pipeline) |
|------------------------|------------|-------------|-----------|--------------------|
| MamayLM-Gemma-3-12B-IT | FP8 static | NVIDIA L4   | 22 GB     | ~75                |
| Gemma 3 27B IT         | BF16       | NVIDIA A100 | 80 GB     | ~115               |
| Qwen 3.5 27B           | BF16       | NVIDIA A100 | 80 GB     | ~115               |

## 8.2. Main Comparison

Aggregate results at the best-of-N iteration are reported in Table 2. The naïve mode achieves the highest MRS (0.684) but the lowest EPS across all three metrics (recall 0.554, F1 0.543), demonstrating that iterative rewriting without entity grounding effectively reduces manipulation at the cost of factual drift. Its gate pass rate of 8.82% confirms that simultaneously satisfying all three thresholds is nearly infeasible.

The entity-grounded mode improves EPS-recall substantially to 0.778 and EPS-F1 to 0.740, with a gate pass rate of 23.53%. The three-stage reconciliation cascade resolves 14.6% of missing entities via fuzzy substitution; the remaining 85.4% are escalated to subsequent iterations via the reconciliation failures block. MRS is marginally lower than the naïve mode (0.663 vs. 0.684), reflecting that roster injection slightly constrains rephrasing freedom without strongly limiting reduction capability.

The placeholder-based mode achieves the strongest overall entity preservation (EPS-recall 0.853, EPS-precision 0.839, EPS-F1 0.822) and the highest gate pass rate of all MamayLM configurations (33.82%). The residual ManipSum (0.526) is the highest among the three MamayLM modes, suggesting that placeholder substitution constrains the LLM's rephrasing freedom, moderately limiting manipulation reduction.

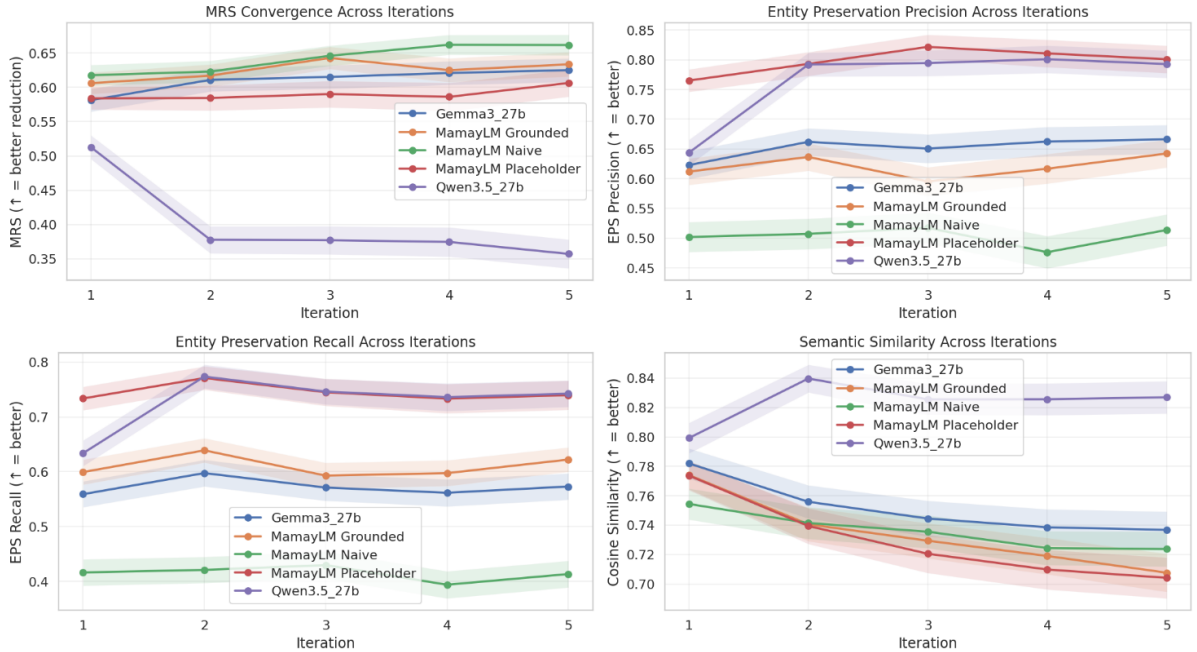
Paired Wilcoxon signed-rank tests ( $n = 204$ ) reveal that on MRS all pairwise comparisons are significant ( $p < 0.01$ ) with one exception: MamayLM Placeholder vs. Gemma 3 27B ( $p = 0.805$ ). On EPS-recall, all pairs are significant except MamayLM Placeholder vs. Qwen ( $p = 0.075$ ). On CosSim, two pairs are not significant: MamayLM Grounded vs. MamayLM Placeholder ( $p = 0.197$ ) and MamayLM Placeholder vs. Gemma 3 27B ( $p = 0.296$ ). The grounded and placeholder modes are statistically distinguishable on both MRS ( $p = 0.006$ ) and EPS-recall, while being equivalent on semantic similarity.

**Table 2**Best-of- $N$  results across all five configurations ( $n = 204$ , Original Manipulation Sum = 2.363).

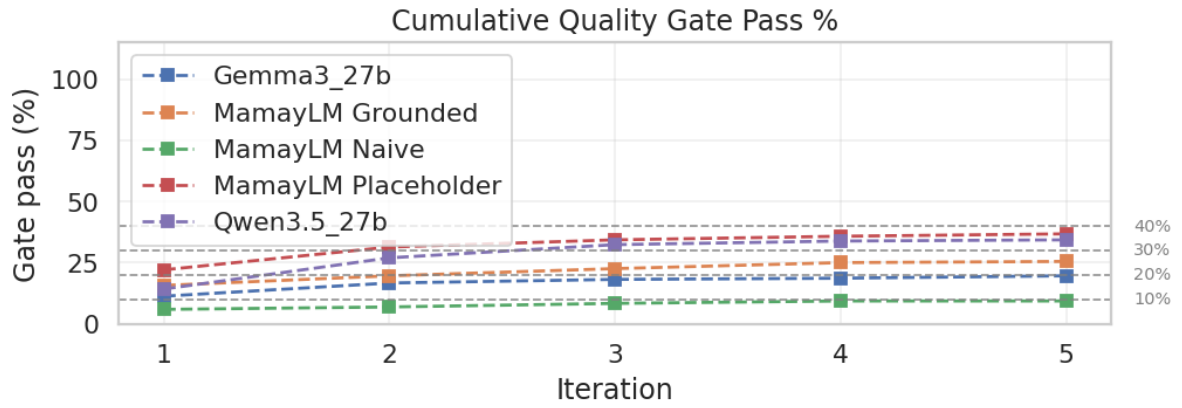
| Configuration       | MRS $\uparrow$ | EPS-R $\uparrow$ | EPS-P $\uparrow$ | EPS-F1 $\uparrow$ | CosSim $\uparrow$ | ManipSum $\downarrow$ | Gate%        | Avg.Iter |
|---------------------|----------------|------------------|------------------|-------------------|-------------------|-----------------------|--------------|----------|
| MamayLM Naïve       | <b>0.684</b>   | 0.554            | 0.587            | 0.543             | 0.753             | <b>0.397</b>          | 8.82         | 4.70     |
| MamayLM Grounded    | 0.663          | 0.778            | 0.740            | 0.740             | 0.764             | 0.444                 | 23.53        | 4.17     |
| MamayLM Placeholder | 0.641          | <b>0.853</b>     | <b>0.839</b>     | <b>0.822</b>      | 0.772             | 0.526                 | <b>33.82</b> | 3.76     |
| Qwen 3.5 27B        | 0.501          | 0.819            | 0.799            | 0.782             | <b>0.836</b>      | 0.967                 | 28.43        | 3.93     |
| Gemma 3 27B         | 0.633          | 0.696            | 0.729            | 0.684             | 0.773             | 0.534                 | 19.12        | 4.35     |

### 8.3. Convergence and Gate Pass Analysis

Metrics across all iterations are shown in Figure 2. For naïve rewriting, MRS increases from iteration 1 through iteration 3 and plateaus near 0.66, while per-iteration mean EPS recall remains around 0.42 with substantial inter-article variance. Because the composite score (Equation 3) weights entity preservation alongside manipulation reduction, Best-of- $N$  selection recovers a higher aggregate mean EPS recall of 0.554 by selecting iterations where entity preservation was not sacrificed for manipulation reduction. The placeholder mode achieves the highest EPS recall among MamayLM variants, stabilising near 0.74–0.77 from iteration 2 onward, bounding factual drift from early passes. The grounded mode shows a slightly lower EPS recall trajectory ( $\approx 0.60$ – $0.64$ ), consistent with less explicit entity-slot enforcement compared to placeholder substitution. As shown in Figure 3, gate pass rates accumulate most rapidly under the placeholder mode, reaching approximately 28% by iteration 2 and 34% by iteration 5. The grounded mode reaches 23% by iteration 5, while the naïve mode remains below 9% at all steps. Both Qwen and Gemma 3 27B reach intermediate pass rates (28% and 19% respectively) but require more iterations on average.

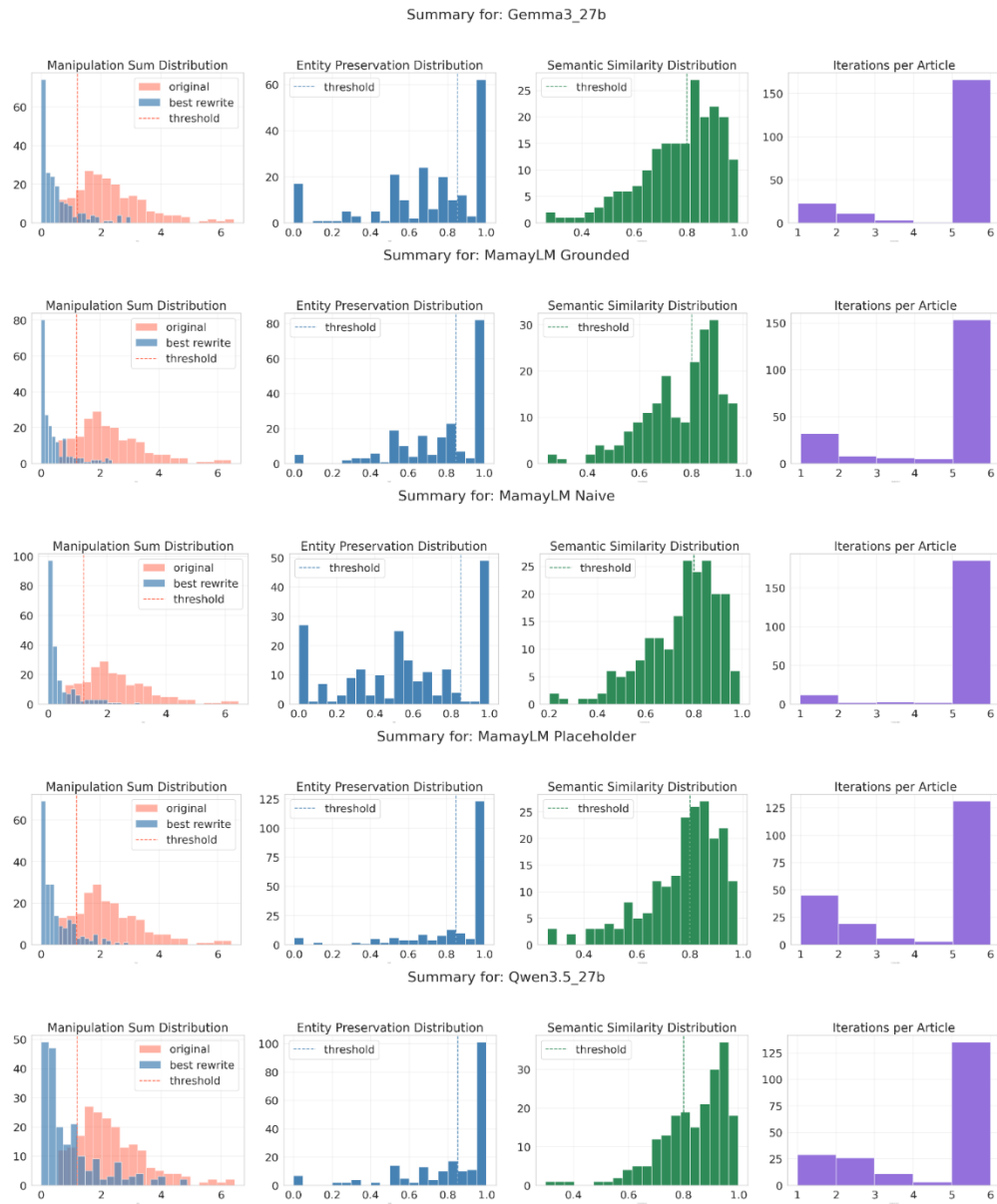


**Figure 2:** MRS, EPS-recall, EPS-precision and CosSim across iterations 1–5 for all five configurations. Shaded bands show  $\pm 1$  standard error of the mean computed across articles at each iteration.



**Figure 3:** Cumulative quality-gate pass rate (%) by iteration for all five configurations, with per-step percentage annotations.

## 8.4. Metric Distributions

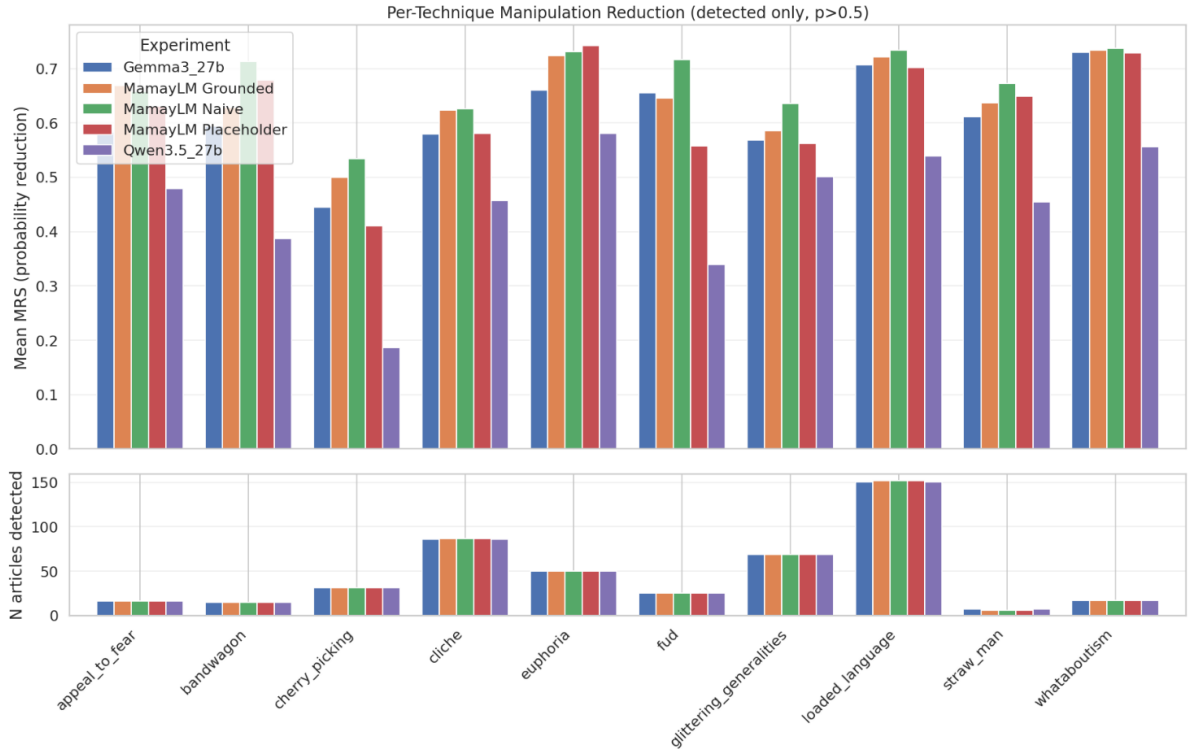


**Figure 4:** Per-experiment distributions of ManipSum (original vs. best rewrite, with threshold line at  $\tau_M = 1.2$ ), EPS-recall (with threshold at  $\theta_F = 0.85$ ), CosSim (with threshold at  $\theta_S = 0.80$ ), and iteration count, for all five configurations.

The distribution plots in Figure 4 reveal structural differences between configurations that aggregate means in Table 2 do not capture. For ManipSum, all MamayLM configurations shift the distribution sharply leftward relative to the original (mean 2.363), with the naïve mode producing the lowest final ManipSum concentrations. Qwen’s ManipSum distribution remains substantially overlapping with the original, confirming its conservative rewriting behaviour. For EPS-recall, the naïve mode produces a wide bimodal distribution – many articles achieve near-perfect entity preservation while others suffer severe entity loss – whereas both the grounded and placeholder modes concentrate mass above 0.7. CosSim distributions are broadly similar across MamayLM configurations, while Qwen’s distribution is shifted right (mean 0.836), reflecting its tendency toward minimal lexical change. The iterations-per-article histograms show that the placeholder mode more frequently terminates early (iterations 2–3), whereas the naïve mode almost always exhausts all five iterations.

### 8.5. Per-Technique MRS Breakdown

Loaded language (152 articles) and euphoria (50) show the strongest and most consistent reduction across all configurations. Glittering generalities and cliché show moderate reduction. Cherry-picking is the least responsive: the LLM tends to paraphrase rather than structurally remove the rhetorical device. Straw man and whataboutism results (6 and 17 articles) should be interpreted with caution.



**Figure 5:** Per-technique mean MRS at best-of-N iteration for all five configurations. Only articles in which the technique is detected are included. Article counts per technique shown on secondary axis.

### 8.6. LLM Model Comparison

In the naïve mode, MamayLM-12B FP8 achieves the strongest manipulation reduction (MRS 0.684), outperforming Gemma 3 27B (MRS 0.633) and Qwen 3.5 27B (MRS 0.501) despite being approximately 2.25× smaller in parameter count. All pairwise MRS comparisons involving Qwen

are significant ( $p < 0.001$ ); the MamayLM-12B vs. Gemma 3 27B difference in the naïve mode only is also significant ( $p < 0.001$ ). Since parameter count and Ukrainian adaptation differ jointly across these three models, the observed gap may reflect either factor or their interaction, and we view the result as suggestive rather than as isolating the contribution of adaptation; it is broadly in line with reports that smaller domain-tuned models can be competitive with larger general-purpose ones on targeted tasks [38]. This finding has practical implications for deployment – the MamayLM-12B FP8 configuration reduces per-article processing cost by approximately 35% in wall-clock time while producing better neutralisation quality, making it the recommended writer for production settings where hardware budget is constrained.

## 9. Discussion and Limitations

**Failure mode analysis.** Table 3 summarises the dominant gate failure criterion per configuration. Under naïve rewriting, low EPS-recall is the primary obstacle (82% of failures). The grounded mode reduces but does not eliminate this problem (72%), as fuzzy reconciliation alone recovers only a minority of dropped entities. The placeholder mode inverts the bottleneck to semantic drift (72%), since placeholder-constrained generation produces factually consistent but stylistically more distant rewrites. Qwen is the only configuration in which residual manipulation constitutes a substantial share of failures (37%). Also, the absolute values of both EPS and CosSim are dependent on the choice of NER backbone and embedding model respectively.

**Table 3**

Failure-mode breakdown for articles that never satisfied the early-exit criterion. Columns are non-disjoint: an article can fail on multiple criteria simultaneously, so row totals exceed 100%.

| Configuration       | Never passed | ManipSum high | EPS-recall low | CosSim low |
|---------------------|--------------|---------------|----------------|------------|
| MamayLM Naïve       | 186 (91%)    | 10%           | 82%            | 61%        |
| MamayLM Grounded    | 156 (76%)    | 13%           | 72%            | 62%        |
| MamayLM Placeholder | 135 (66%)    | 18%           | 49%            | 72%        |
| Qwen 3.5 27B        | 146 (72%)    | 37%           | 56%            | 49%        |
| Gemma 3 27B         | 165 (81%)    | 17%           | 77%            | 56%        |

**Limitations.** MRS carries a degree of circularity – the same classifier defines  $T_d$  and produces the probability deltas – and should be interpreted as a relative metric across configurations rather than an absolute neutralisation estimate. The article-level detection backbone provides no span-level localisation, preventing more targeted rewriting of specific manipulative passages. No human evaluation is conducted, so the correspondence between automatic metrics and the judgements of Ukrainian media analysts remains unverified. Processing a single article through five iterations requires approximately 75 seconds on an L4 GPU or 115 seconds on an A100, which may constrain high-throughput deployment.

Finally, all writer prompts and score critiques are composed in Ukrainian; whether English-language prompts would produce different neutralisation quality for Ukrainian-language articles was not ablated and remains an open question.

**Ethical considerations.** The pipeline is dual-use: the same architecture could introduce manipulation into neutral text if differently prompted. The system is intended as a decision-support tool for editors and media analysts, and its outputs should not be presented to readers without human review.

## 10. Conclusion

An iterative score-guided pipeline for manipulation reduction in Ukrainian news texts has been proposed and evaluated across five experimental configurations on a 204-article stratified sample from the UNLP 2025 dataset. Three factual preservation strategies are comparatively evaluated with MamayLM-Gemma-3-12B-IT FP8 as the primary writer, and two additional 27B-parameter models are benchmarked in the naïve mode. The placeholder-based mode achieves the best overall entity preservation (EPS-F1 0.822) and the highest gate pass rate among MamayLM configurations (33.82%), making it the recommended configuration when factual integrity is the primary concern. The entity-grounded mode provides an intermediate trade-off (EPS-recall 0.778, gate pass rate 23.53%) with competitive manipulation reduction (MRS 0.663). A key finding concerns the relationship between model scale and task performance: MamayLM-12B FP8 achieves the strongest manipulation reduction (MRS 0.684) despite being approximately  $2.25\times$  smaller than the comparison models, outperforming both Gemma 3 27B IT (MRS 0.633) and Qwen 3.5 27B (MRS 0.501). Because adaptation and scale differ jointly across these models, we frame this as evidence that the combination of Ukrainian adaptation and the iterative pipeline is sufficient to outperform larger off-the-shelf models, leaving the relative contribution of each factor as an open question.

Three directions are identified for future work: fine-tuning the writer LLM on verified (manipulative, neutral) article pairs to reduce dependence on iterative correction; extending the pipeline to span-level targeted rewriting once a span-identification model for Ukrainian is available; and conducting a human evaluation study with professional Ukrainian media analysts to validate and calibrate the automatic metrics.

## Declaration on Generative AI

During the preparation of this work, the authors used Claude Sonnet 4.6 in order to: Paraphrase and reword; Citation management. After using these tools, the authors reviewed and edited the content as needed and took full responsibility for the publication’s content.

## References

- [1] R. Kyslyi, N. Romanyshyn, V. Sydorskyi, The UNLP 2025 Shared Task on Detecting Social Media Manipulation, in: Proceedings of the Fourth Ukrainian Natural Language Processing Workshop (UNLP 2025), Association for Computational Linguistics, Vienna, Austria, 2025. doi:10.18653/v1/2025.unlp-1.12.
- [2] K. Akhynko, O. Kosovan, M. Trokhymovych, Hidden Persuasion: Detecting Manipulative Narratives on Social Media During the 2022 Russian Invasion of Ukraine, in: Proceedings of the Fourth Ukrainian Natural Language Processing Workshop (UNLP 2025), Association for Computational Linguistics, Vienna, Austria, 2025, pp. 169–179. doi:10.18653/v1/2025.unlp-1.19.
- [3] R. Pryzant, R. Diehl Martinez, N. Dass, S. Kurohashi, D. Jurafsky, D. Yang, Automatically Neutralizing Subjective Bias in Text, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 01, 2020, pp. 480–489. doi:10.1609/aaai.v34i01.5385.
- [4] K. Madanagopal, J. Caverlee, Bias Neutralization in Non-Parallel Texts, in: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP 2023), Association for Computational Linguistics, 2023, pp. 14232–14248. doi:10.18653/v1/2023.emnlp-main.882.
- [5] A. Madaan, N. Tandon, P. Gupta, S. Hallinan, L. Gao, S. Wiegrefe, U. Alon, N. Dziri, S. Prabhumoye, Y. Yang, S. Welleck, B. P. Majumder, S. Gupta, A. Yazdanbakhsh, P. Clark, Self-Refine: Iterative Refinement with Self-Feedback, in: Advances in Neural Information Processing Systems 36 (NeurIPS 2023), 2023. doi:10.5555/3666122.3668141.
- [6] G. Da San Martino, A. Barron-Cedeno, H. Wachsmuth, R. Petrov, P. Nakov, SemEval-2020 Task 11: Detection of Propaganda Techniques in News Articles, in: Proceedings of the

- Fourteenth Workshop on Semantic Evaluation (SemEval 2020), International Committee for Computational Linguistics, Barcelona (online), 2020, pp. 1377–1414. doi:10.18653/v1/2020.semeval-1.186.
- [7] Iu. V. Krak, V. O. Didur, M. O. Molchanova, O. V. Mazurets, O. V. Sobko, O. O. Zalutska, O. V. Barmak, Method for political propaganda detection in internet content using neural network natural language processing tools, *Problems in Programming* 2024; 2-3: 288–295. doi:10.15407/pp2024.02-03.288.
  - [8] G. Da San Martino, A. Barron-Cedeno, P. Nakov, Findings of the NLP4IF-2019 Shared Task on Fine-Grained Propaganda Detection, in: *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom (NLP4IF 2019)*, Association for Computational Linguistics, Hong Kong, China, 2019, pp. 162–170. doi:10.18653/v1/D19-5024.
  - [9] J. Piskorski, N. Stefanovitch, G. Da San Martino, P. Nakov, SemEval-2023 Task 3: Detecting the Category, the Framing, and the Persuasion Techniques in Online News in a Multi-lingual Setup, in: *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, Association for Computational Linguistics, Toronto, Canada, 2023, pp. 2343–2361. doi:10.18653/v1/2023.semeval-1.317.
  - [10] T. Hromadka, T. Smolen, T. Remis, B. Pecher, I. Srba, KInITVeraAI at SemEval-2023 Task 3: Simple yet Powerful Multilingual Fine-Tuning for Persuasion Techniques Detection, in: *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, Association for Computational Linguistics, Toronto, Canada, 2023, pp. 629–637. doi:10.18653/v1/2023.semeval-1.86.
  - [11] M. Khandoga, Y. Kostiuk, A. Polishko, K. Kozlov, Y. Filipchuk, A. Kiulian, Framing the Language: Fine-Tuning Gemma 3 for Manipulation Detection, in: *Proceedings of the Fourth Ukrainian Natural Language Processing Workshop (UNLP 2025)*, Association for Computational Linguistics, Vienna, Austria, 2025, pp. 49–54. doi:10.18653/v1/2025.unlp-1.6.
  - [12] L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, W. Peng, X. Feng, B. Qin, T. Liu, A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions, *ACM Transactions on Information Systems* 43 (2025) 1–55. doi:10.1145/3703155.
  - [13] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, P. Fung, Survey of Hallucination in Natural Language Generation, *ACM Computing Surveys* 55 (2023) 1–38. doi:10.1145/3571730.
  - [14] Y. Bai, S. Kadavath, S. Kundu, A. Askell, J. Kernion, A. Jones, A. Chen, A. Goldie, A. Mirhoseini, C. McKinnon, C. Chen, C. Olsson, C. Olah, D. Hernandez, D. Drain, D. Ganguli, D. Li, E. Tran-Johnson, E. Perez, et al., Constitutional AI: Harmlessness from AI Feedback, *arXiv preprint arXiv:2212.08073*, 2022. doi:10.48550/arXiv.2212.08073.
  - [15] N. Shinn, F. Cassano, A. Gopinath, K. Narasimhan, S. Yao, Reflexion: Language Agents with Verbal Reinforcement Learning, in: *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*, 2023. doi:10.48550/arXiv.2303.11366.
  - [16] D. Jin, Z. Jin, Z. Hu, O. Vechtomova, R. Mihalcea, Deep Learning for Text Style Transfer: A Survey, *Computational Linguistics* 48 (2022) 155–205. doi:10.1162/coli\_a\_00426.
  - [17] M. Ranzato, S. Chopra, M. Auli, W. Zaremba, Sequence Level Training with Recurrent Neural Networks, in: *Proceedings of the 4th International Conference on Learning Representations (ICLR 2016)*, San Juan, Puerto Rico, 2016.
  - [18] INSAIT, MamayLM-Gemma-3-4B-IT-v1.0, Hugging Face model repository, 2025. URL: <https://huggingface.co/INSAIT-Institute/MamayLM-Gemma-3-4B-IT-v1.0>.
  - [19] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, LoRA: Low-Rank Adaptation of Large Language Models, in: *Proceedings of the 10th International Conference on Learning Representations (ICLR 2022)*, 2022. doi:10.48550/arXiv.2106.09685.
  - [20] P. Manakul, A. Liusie, M. J. F. Gales, SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models, in: *Proceedings of EMNLP*, 2023. doi:10.18653/v1/2023.emnlp-main.557.

- [21] F. Nan, et al., Entity-level factual consistency of abstractive text summarization, in: Proceedings of EACL, 2021, pp. 2727–2733. doi:10.18653/v1/2021.eacl-main.235.
- [22] A. Pagnoni, V. Balachandran, Y. Tsvetkov, Understanding factuality in abstractive summarization with FRANK: A benchmark for factuality metrics, in: Proceedings of NAACL-HLT, 2021, pp. 4812–4829. doi:10.18653/v1/2021.naacl-main.383.
- [23] Explosion AI, spaCy Ukrainian model: uk\_core\_news\_sm, 2023. URL: [https://spacy.io/models/uk#uk\\_core\\_news\\_sm](https://spacy.io/models/uk#uk_core_news_sm).
- [24] V. Oliynyk, A. Biziuk, Z. Deineko, and V. Chelombitko, "Formalization of text prompts to artificial intelligence systems," Eastern-European Journal of Enterprise Technologies, vol. 5, no. 2 (137), pp. 84–97, 2025. doi:10.15587/1729-4061.2025.335473.
- [25] H. R. Bosker, Using fuzzy string matching for automated assessment of listener transcripts in speech intelligibility studies, Behavior Research Methods 53 (2021) 1945–1953. doi:10.3758/s13428-021-01542-4.
- [26] P. Laban, T. Schnabel, P. N. Bennett, M. A. Hearst, SummaC: Re-visiting NLI-based models for inconsistency detection in summarization, Transactions of the Association for Computational Linguistics 10 (2022) 163–177. doi:10.1162/tacl\_a\_00453.
- [27] S. Min, et al., FActScore: Fine-grained atomic evaluation of factual precision in long-form text generation, in: Proceedings of EMNLP, 2023. doi:10.18653/v1/2023.emnlp-main.741.
- [28] A. R. Fabbri, W. Kryscinski, B. McCann, C. Xiong, R. Socher, D. Radev, SummEval: Re-evaluating summarization evaluation, Transactions of the Association for Computational Linguistics 9 (2021) 391–409. doi:10.1162/tacl\_a\_00373.
- [29] Y. Laba, V. Mudryi, D. Chaplynskyi, M. Romanyshyn, O. Dobosevych, Contextual embeddings for Ukrainian: A large language model approach to word sense disambiguation, in: M. Romanyshyn (Ed.), Proceedings of the Second Ukrainian Natural Language Processing Workshop (UNLP), Association for Computational Linguistics, Dubrovnik, Croatia, 2023, pp. 11–19. doi:10.18653/v1/2023.unlp-1.2.
- [30] D. Chaplynskyi, Introducing UberText 2.0: A corpus of modern Ukrainian at scale, in: Proceedings of the Second Ukrainian Natural Language Processing Workshop (UNLP), 2023, pp. 1–10. doi:10.18653/v1/2023.unlp-1.1.
- [31] N. Reimers, I. Gurevych, Sentence-BERT: Sentence embeddings using Siamese BERT-networks, in: Proceedings of EMNLP-IJCNLP, 2019, pp. 3982–3992. doi:10.18653/v1/D19-1410.
- [32] MamayLM-Gemma-3-12B-IT-v1.0-FP8-Static-Nadiia-v2, Hugging Face model repository, 2025. URL: <https://huggingface.co/oshkorinova/MamayLM-Gemma-3-12B-IT-v1.0-FP8-Static-Nadiia-v2>.
- [33] INSAIT, MamayLM-Gemma-3-12B-IT-v1.0, Hugging Face model repository, 2025. URL: <https://huggingface.co/INSAIT-Institute/MamayLM-Gemma-3-12B-IT-v1.0>.
- [34] Google, Gemma 3 27B IT, Hugging Face model repository, 2025. URL: <https://huggingface.co/google/gemma-3-27b-it>.
- [35] Qwen Team, Qwen3.5: Towards Native Multimodal Agents, 2026. URL: <https://qwen.ai/blog?id=qwen3.5>.
- [36] Alibaba Cloud, Qwen3.5-27B, Hugging Face model repository, 2026. URL: <https://huggingface.co/Qwen/Qwen3.5-27B>.
- [37] W. Kwon, Z. Li, S. Zhuang, Y. Sheng, L. Zheng, C. H. Yu, J. Gonzalez, H. Zhang, I. Stoica, Efficient Memory Management for Large Language Model Serving with PagedAttention, in: Proceedings of the 29th Symposium on Operating Systems Principles (SOSP), 2023. doi:10.1145/3600006.3613165.
- [38] B. O. Kuzikov, O. A. Shovkoplias, P. O. Tytov, S. R. Shovkoplias, Application of small language models for semantic analysis of Web interface accessibility, Problems in Programming 2025; 2: 77–86. doi:10.15407/pp2025.02.077.