Personalised Classifier-Guided Decoding: Steering LLM Toxicity Along User-Specified Directions

Miruna Elena Coroi

Supervisors: Jie Yang, Anne Arzberger, Enrico Liscio
EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 17, 2026

Name of the student: Miruna Elena Coroi
Final project course: CSE3000 Research Project
Thesis committee: Jie Yang, Anne Arzberger, Enrico Liscio

Abstract

Toxic content is not universally defined: what one user finds offensive, another may find acceptable depending on cultural background, context, and purpose. Current LLM safety systems apply a single global toxicity threshold to every user, and adapting this behaviour after deployment is expensive. This paper asks whether a frozen LLM can instead be steered at inference time to follow individual users' toxicity preferences across six toxicity dimensions, without retraining. A classifier-guided decoding framework driven by a per-user sensitivity vector is instantiated as three deployable strategies and evaluated on the PRISM preference dataset. All three strategies reduce per-user toxicity error by 15–21%, while preserving general-knowledge accuracy to within 0.7 pp of the unguided baseline. The central finding is *directional steerability*: the decoder responds to the *shape* of a user's preference vector, producing category-specific reductions that align with per-user weights (median cosine similarity 0.845, $p = 0.0097$ above a permutation baseline). These results show that meaningful personalised toxicity control is achievable at deployment time, without retraining the model.

1 Introduction

Large language models (LLMs) generate fluent text, but controlling harmful or toxic content in their outputs remains an open problem. Standard alignment techniques, such as supervised fine-tuning and reinforcement learning from human feedback, enforce a single global notion of acceptable behaviour that is encoded in the model parameters and expensive to update after deployment. However, users differ in their cultural backgrounds, contexts, and personal sensitivities, and toxicity itself is not universally defined [1, 2]. For example, a doctor is comfortable with graphic medical language, while a patient discussing the same condition may be sensitive to it. A single global threshold cannot accommodate both. Users differ not just in how much toxicity they tolerate overall, but in which categories matter most to them. For instance, one user may be unbothered by profanity but highly sensitive to identity attacks, while another has the opposite preferences.

Training-free alignment methods offer an alternative: instead of modifying model parameters, they adjust behaviour only at inference time [3]. In classifier-guided decoding, token selection is made toxicity-aware without touching the model weights. At each generation step, the base model produces a shortlist of likely next words. A lightweight classifier then re-ranks that shortlist by assigning lower probability to any continuation it predicts would push the text toward toxicity. The model picks from this re-ranked shortlist. Because the intervention happens word-by-word, a toxic phrase can be blocked before it is written rather than flagging a completed response after the fact. Existing classifier-guided methods such as FUDGE [4] and GeDi [5] apply a single scalar control signal, which cannot express that one user may be sensitive to identity attacks while another cares most about profanity.

This paper studies *multi-category classifier-guided decoding*: instead of a single scalar toxicity score, a toxicity classifier rates each candidate continuation across six dimensions (toxicity, severe toxicity, identity attack, insult, profanity, and threat). These six categories come from the two toxicity classifiers used in this paper: Detoxify, which guides decoding, and Google's Perspective API, which evaluates the result. A per-user sensitivity vector specifies how strongly a user penalises each dimension, so the goal is not only to reduce toxicity overall, but to steer reductions in the direction implied by the user's profile. Because decoding-time steering directly changes token choices, it must also preserve fluency: if the guidance is too strong, the response may become less readable, less consistent with the prompt, or less useful.

Research question. *How can classifier-guided decoding be calibrated to reflect individual users' toxicity thresholds while maintaining fluency?*

This question is decomposed into three sub-questions: (SQ1) how much does multi-category classifier-guided decoding reduce per-user toxicity error relative to an unguided baseline? (SQ2) how does the guidance strength affect fluency, utility, and decoding throughput? (SQ3) do per-user weight vectors steer the decoder in a *directional* manner, so that per-category reductions follow the user's preferences?

Contributions. The research question is answered through one framework and three contributions. The basis is a training-free classifier-guided decoding framework, using a per-user weight vector expressing which toxicity categories matter most. For *SQ1*, three methods are implemented: **M1** reduces toxicity for all users; **M2** does not apply guidance to users who tolerate more toxic content; and **M3** applies guidance only when a response's toxicity exceeds the user's threshold. A fourth extension (**M4**), which learns how much guidance each prompt needs, did not generalise under the available PRISM-derived supervision signal. For *SQ2*, I evaluate fluency, MMLU accuracy, and throughput. For *SQ3*, I test whether per-category reductions follow each user's weights against a shuffled-weight baseline.

Key findings. All three strategies reduce per-user toxicity error (15–21% overall, up to 34% where guidance is active) while having minimal impact on fluency (perplexity increases by at most ≈ 0.2 points) and keeping utility close to the baseline. Most importantly, the reductions are *directional*: the decoder lowers the cat-

egories each user cares about most (median alignment 0.845, $p = 0.0097$). Personalised toxicity control is thus achievable from the existing ratings, without retraining the model.

2 Background and Related Work

This section covers the building blocks the method rests on: inference-time steering mechanisms, toxicity classifiers, the top-k truncation that makes per-step scoring feasible, and per-user preference data.

Inference-time controlled generation. Recent research studies how frozen language models can be steered at inference time without updating their parameters. Pan et al. [3] distinguish between *pre-decoding* methods such as prompting, *in-decoding* methods which modify token selection during generation, and *post-decoding* methods which filter or rerank completed outputs. This paper uses in-decoding classifier guidance rather than post-decoding filters. Consequently, a toxic token can be blocked before it is written, rather than discarding a fully generated output. Pre-decoding prompting offers no hard guarantee on the output distribution, so it is also set aside.

Several in-decoding methods have been proposed to steer generation without modifying model parameters. PPLM [6] uses gradients from an attribute classifier to perturb generation. GeDi [5] and FUDGE [4] reweight candidate tokens with a discriminative signal. PPLM requires a backward pass through the model at every generation step, which is slow when many candidates need scoring. GeDi and FUDGE reweight the output distribution directly instead, which is faster and is the approach adopted here.

These methods show that decoding-time control is possible without retraining, but they apply a single scalar control signal. They cannot express that two users may care about different *kinds* of toxicity. The gap addressed here is *personalised multi-category control*. Instead of one toxicity score, an external classifier returns scores across the six dimensions (toxicity, severe toxicity, identity attack, insult, profanity, and threat), weighted by a per-user vector that reflects which dimensions each user penalises most.

Toxicity classifiers: roles and limitations. This work uses two toxicity classifiers for two separate roles: one *guides* generation, the other *evaluates* the result. Detoxify [7] guides token selection during decoding, because it can score candidate continuations efficiently on-device across all six categories in a single forward pass. Perspective API [8] evaluates completed responses after generation. Keeping these roles separate matters: if the same classifier both guided generation and evaluated the output, measured improvements could reflect overfitting to that classifier's own behaviour rather than a genuine reduction in toxicity. This separation reduces circularity, but does not address a limitation both classifiers share.

Automated toxicity classifiers are not neutral. Prior work shows they can assign high toxicity scores to text that is not actually abusive, such as identity-related language [9]. In practice, this means Detoxify may penalise a non-abusive token during decoding, causing the model to avoid language that would have been perfectly acceptable. Results throughout the paper should therefore be read as classifier-based toxicity measurements rather than direct human judgements of harm.

Top-k decoding. Restricting guidance to the top-k most probable tokens makes the approach computationally feasible: the classifier scores k candidates per step rather than the full vocabulary. Top-k truncation keeps only the k most probable tokens and discards the unreliable low-probability tail [10]. It is a widely used decoding choice; for example, $k = 10$ in [11] and $k = 40$ in [10]. The main experiments use $k = 20$, as a trade-off between steering flexibility and decoding cost.

User-level toxicity preferences. The decoding mechanism and classifiers above handle *how* to steer, so the remaining question is *whose* preferences to steer towards. Pluralistic alignment argues that aggregating diverse human values into one global objective erases meaningful differences between users [12, 2]. For toxicity control, this means a single safety threshold is insufficient: a useful system should know both how strict a user is overall and which categories matter most to them. PRISM [13] is one of the few public datasets that supports this kind of per-user analysis, providing persistent user identities, ratings, and accepted responses. It was designed to capture diverse human preferences around helpfulness and response quality across a wide range of open-domain conversations.

3 Methodology

This section defines the training-free decoding framework. The base language model is never updated; personalisation is introduced entirely at inference time. Section 3.1 defines how per-user profiles are derived from PRISM ratings. Section 3.2 defines the shared guided-decoding loop. Section 3.3 introduces four penalty functions (M1–M4) that instantiate progressively more complex steering policies.

3.1 User modelling

Per-user sensitivity vector $\mathbf{w}_u$. For each user u with annotation history P_u from the PRISM dataset (Section 4.1), the per-response *dislike weight* is $d_{u,p} = 1 - R_{u,p}/100 \in [0, 1]$, where $R_{u,p}$ is the rating u assigned to response p. The per-category sensitivity is the dislike-

weighted average toxicity of u's rated responses:

$$w_{u,c} = \frac{\sum_{p \in P_u} d_{u,p}\, T_c(p)}{\sum_{p \in P_u} d_{u,p}}. \quad (1)$$

Categories that u frequently rated down in toxic responses receive a high weight; categories tolerated or rarely encountered in disliked content receive a low weight. The vector $\mathbf{w}_u$ is computed *once per user*. In PRISM, users with a high mean weight $\bar{w}_u$ tend to have encountered more toxic content overall, so $\bar{w}_u$ reflects toxicity *exposure* as much as sensitivity. This distinction matters for M2 (Section 3.3.2).

Evaluation target. For each evaluated prompt, the target $\mathbf{s}_{\text{target}}$ is the Perspective API toxicity vector of the response accepted by the same PRISM user in the original conversation. This target is a proxy for the user's tolerated toxicity profile rather than a direct toxicity label; the implications are discussed in Section 6. The evaluation metrics derived from this target are defined in Section 4.

3.2 Shared decoding framework

The framework works the same way at every generation step. The base language model proposes the next tokens, and guidance re-scores them so toxic ones become less likely while harmless ones stay largely unchanged. All four methods follow this loop and differ only in how the penalty is computed.

Let $y_{<t}$ be the text generated so far. The base model gives a distribution over the next token, from which the decoder keeps the k most probable candidates ($k = 20$) to keep per-step scoring cheap. Each such candidate v receives a penalty $\pi_u(v, y_{<t}) \geq 0$ measuring how toxic that token is for user u. Its base log-probability is reduced by this penalty:

$$\log P_{\text{guided}}(v \mid y_{<t}) = \log P_{\text{base}}(v \mid y_{<t}) - \pi_u(v, y_{<t}). \quad (2)$$

A larger penalty makes a toxic token less likely to be chosen; a zero penalty leaves its base ranking unchanged. The penalty is built from per-category toxicity scores. $\mathcal{C}$ denotes the six toxicity categories: toxicity, severe toxicity, identity attack, insult, profanity, and threat. For a candidate v, $T_c(v \mid y_{<t}) \in [0, 1]$ is Detoxify's score for category c, with higher values indicating greater toxicity. Detoxify provides these scores during decoding, whereas Perspective API is used only afterwards to evaluate completed responses. The two share the same six axes, except that Detoxify's `obscene` score corresponds to Perspective's `profanity` attribute.

3.3 Penalty function methods

All four methods fit Eq. (2) and differ only in how the penalty π_u is defined. They form a progression in *how much guidance each user or prompt receives*. M1 applies the same guidance strength to everyone. M2 keeps that strength but turns guidance off for users who tolerate more toxic content. M3 makes the penalty conditional, acting only when a candidate is more toxic than what the user accepts. M4 goes further and tries to learn the right strength for each prompt. The per-user *direction* of steering, given by $\mathbf{w}_u$, is shared by all four; the methods change only *when* and *how strongly* guidance is applied.

3.3.1 M1: Always-on linear guidance

M1 is the personalised baseline of the paper: it applies guidance to every user at every step, letting the user's sensitivity vector $\mathbf{w}_u$ decide which toxicity categories to suppress most. A single strength setting controls how hard the guidance pushes, the same for everyone.

The penalty for a candidate token is the weighted sum of its six Detoxify scores, where each category is weighted by how much the user dislikes it:

$$\pi_u(v, y_{<t}) \;=\; \alpha \sum_{c \in \mathcal{C}} w_{u,c}\, T_c(v \mid y_{<t}). \quad (3)$$

The global coefficient $\alpha \geq 0$ sets the overall guidance strength, while the per-user direction comes entirely from $\mathbf{w}_u$. Setting $\alpha = 0$ removes the penalty and recovers the unguided baseline exactly, which makes M1 a clean ablation point.

Hyperparameter. $\alpha = 30{,}000$ is selected on a 700-prompt validation set disjoint from all three reported test seeds. For each candidate $\alpha \in \{1.5, 3, 8, 15, 30, 35, 40, 45, 50, 80\} \times 10^3$, responses are generated and scored with Perspective API; the value minimising mean MAE against the per-user target is retained and held fixed for all reported results.

3.3.2 M2: User-level gated guidance

M2 addresses a limitation of M1: lowering toxicity does not help users who are already comfortable with it. These tolerant users have historically accepted fairly toxic responses, so M1's penalty would push their outputs below the level they accept, applying guidance where it is not needed. M2 therefore leaves the most tolerant users unguided and applies M1 to everyone else.

Each user's tolerance is summarised by the aggregate weight $\bar{w}_u = \frac{1}{|\mathcal{C}|} \sum_c w_{u,c}$, the average of their per-category weights. M2 compares $\bar{w}_u$ to a threshold τ, and does not apply the algorithm for users that accepted higher toxicity scores in the past:

$$\alpha_u = \begin{cases} \alpha_{\text{on}} & \text{if } \bar{w}_u \leq \tau, \\ 0 & \text{otherwise,} \end{cases} \qquad \pi_u = \alpha_u \sum_{c \in \mathcal{C}} w_{u,c}\, T_c(v \mid y_{<t}). \quad (4)$$

The gate is evaluated once per user from $\bar{w}_u$ alone, so M2 has the same runtime cost as M1 after the gate assignment.

The per-category direction $\mathbf{w}_u$ is unchanged; the gate decides only *whether* the penalty is applied.

Hyperparameters. $\alpha_{\text{on}} = 35{,}000$ is selected by the same validation-set sweep as M1. The gate threshold $\tau = 0.030$ is chosen from a sweep over $\{0.015, 0.020, 0.022, 0.024,\ 0.025, 0.026, 0.028, 0.030\}$ on the same validation set, selecting the value that minimised mean MAE jointly across the full user population.

3.3.3 M3: Thresholded log-space guidance

M1 and M2 apply the same guidance to every prompt from a user. M3 adds prompt-level awareness: it acts only on tokens whose toxicity rises above the user's threshold. When the output stays within what the user normally accepts, M3 applies no penalty; it activates only on the parts that overshoot. Consequently, benign prompts are left alone and steering is used only where it is needed.

Per-user threshold. Each user gets a threshold per category, set to the median Detoxify score of the responses they have accepted, P_u^{acc}:

$$\tau_{u,c} = \underset{p \in P_u^{\text{acc}}}{\text{median}}\ T_c(p). \tag{5}$$

The median is used instead of the mean because these toxicities are skewed: a few high values would pull the mean up and inflate the threshold. Across the 1,227 PRISM users, the median predicts a held-out accepted response 14–16% more accurately than the mean, rating-weighted mean, or quartile-midpoint estimators.

Threshold penalty. For each category, the penalty is the amount by which a candidate's toxicity exceeds the user's threshold, and zero below it (with $\varepsilon = 10^{-5}$):

$$\pi_c(v) = \big[\max\big(0,\ \log(T_c(v)+\varepsilon) - \log(\tau_{u,c}+\varepsilon)\big)\big]^2, \tag{6}$$

$$\pi_u(v, y_{<t}) = \alpha \sum_{c \in \mathcal{C}} w_{u,c}\, \pi_c(v). \tag{7}$$

Three properties follow: (i) the penalty is zero whenever the candidate is at or below the user's threshold; (ii) the log scale separates small but real differences at low toxicity (e.g. 0.01 vs 0.02 is a factor-of-two gap); (iii) squaring punishes large overshoots harder. The per-category penalties are weighted by $\mathbf{w}_u$, preserving the same per-user direction as M1/M2. Note that α is not comparable across methods, since this squared log-ratio is dimensionally different from the linear product in M1/M2.

Hyperparameter. $\alpha = 75.0$, selected on the same 700-prompt validation set by a sweep over $\{0.5, 1, 2, 5, 30, 75, 150, 300\}$.

3.3.4 M4: Learning α per prompt

The previous three methods determine guidance strength before seeing the prompt. M4 attempts to learn α as a function of the (user, prompt) pair using a gradient-boosted classifier, so that benign prompts receive little steering and toxic-prone ones receive more. The per-user direction $\mathbf{w}_u$ is unchanged, only α varies. Labels are obtained by decoding each pair at every candidate α from M1's sweep and selecting the best; the classifier is trained on 991 labeled pairs with user-grouped 5-fold cross-validation. The outcome is reported in Section 5.2.

4 Experimental Setup

Model. LLaMA-3.1-8B-Instruct, frozen, bf16.

Classifiers. Detoxify `unbiased` [7] for guidance (batched fp16) and Perspective API as a decoupled evaluator.

Decoding. Every response is generated greedily: at each step the decoder selects the most likely token, up to a limit of 128 tokens. This makes generation deterministic for a given prompt. Steering does not change this rule, it only re-scores the top $k = 20$ candidate tokens before that choice is made.

Hardware. Single A100 40 GB on TU Delft DAIC.

4.1 PRISM dataset

PRISM [13] is a preference dataset in which over 1,500 participants across 75 countries rated AI responses on a 0–100 scale and marked accepted responses. Its persistent user identifiers make per-user analysis possible. Most preference datasets do not track who gave which rating, so they cannot support this kind of analysis.

Of the full dataset, 1,227 users with at least two rated responses contribute a sensitivity vector $\mathbf{w}_u$. For evaluation, 200 users are sampled per random seed $\{0, 13, 100\}$. A 700-prompt validation set is used for hyperparameter selection.

One limitation to keep in mind: PRISM was designed around helpfulness, not toxicity. The per-user toxicity targets are therefore noisy proxies, a point revisited in Section 6.

4.2 Evaluation metrics

All generated responses are scored by Perspective API, yielding a six-dimensional vector $\mathbf{s} \in [0, 1]^6$. Three metrics are used:

MAE measures the mean absolute distance from the user's target across all six categories:

$$\text{MAE} = \frac{1}{6} \sum_c |s_{\text{gen},c} - s_{\text{target},c}|. \tag{8}$$

Relative MAE reduction reports improvement over the unguided baseline:

$$\Delta_{\text{MAE}} = 100 \cdot \frac{M_{\text{gen}} - M_{\text{base}}}{M_{\text{base}}}, \tag{9}$$

where negative values indicate improvement.

Win/Tie rate is the fraction of prompts where the steered MAE $\leq$ baseline MAE.

Affected subset. On some prompts, every top-k candidate has a near-zero Detoxify score, so the penalty is negligible and steering has no effect. Results are reported both on the full 200-prompt set and on the *affected subset* (prompts where at least one candidate is penalised and the method can actually change the output). This subset covers $\approx$ 78% of prompts for M1, $\approx$ 70% for M2 and $\approx$ 13% for M3.

Fluency and utility. Perplexity (PPL) of the generated text under the base LM measures fluency. General knowledge is checked on MMLU [14]. Each question is shown with its four answer options (A - D), and the model is asked to reply with a single letter. The questions come from the `cais/mmlu all` set, shuffled with seed 42, and every method is tested on the same 1,000 questions.

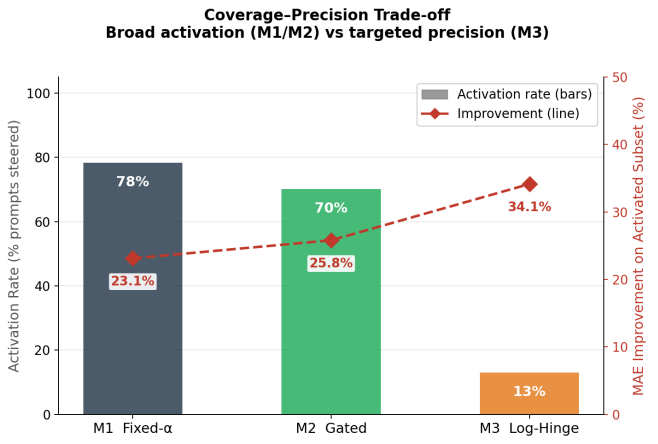

Figure 1: Coverage–precision trade-off across M1–M3: activation rate (bars, left axis) vs. MAE improvement on the affected subset (line, right axis). M3 steers fewer prompts but corrects more strongly when it acts.

5 Results

5.1 SQ1: Toxicity personalisation

Table 1 reports per-seed and seed-averaged results for the three strategies.

Full dataset. Across all 200 prompts per seed, M1 achieves the largest average reduction (-21% MAE), followed by M2 (-19.2%) and M3 (-14.8%). Win/tie rates are 68.8%, 70.7%, and 88.8% respectively. M3's high win/tie rate is not a quality signal in itself: because it leaves most responses unchanged (counted as ties), a method that did nothing would score close to 100%. The relevant reading is that M3 rarely degrades the output when it does act.

Affected subset. On prompts where steering actually changes the output, the ranking reverses: M3 gives the strongest reduction (-34.1%, but on only $N = 78$ pooled prompts), then M2 (-25.8%) and M1 (-23.1%). This reflects a coverage-precision trade-off: M1 activates on $\approx$ 78% of prompts and steers broadly, while M3 activates on only $\approx$ 13% but corrects more when it does. M2 outperforms M1 because it removes the users for whom guidance would push generation below their preferred level. The prompts that remain are those where correction actually helps, so the average gain rises, at the cost of slightly lower coverage ($\approx$ 70%). Figure 1 summarises this trade-off.

Statistical significance. To confirm that MAE reductions are not due to sampling variance, a paired Wilcoxon signed-rank test compares steered vs. baseline MAE for each prompt, pooled across all prompts from the three seeds. All three methods reduce MAE significantly over the unguided baseline: M1 ($W = 41{,}457$, $p < 0.001$), M2 ($W = 35{,}896$, $p < 0.001$), M3 ($W = 6{,}995$, $p < 0.001$), meaning the steered responses are closer to the per-user toxicity target more often than chance would predict.

Seed variance. Seed 13 draws a more toxic-baseline subset (0.01274, vs. $\approx$ 0.009 for seeds 0 and 100), driving the larger reductions across all three methods. M3's spread is additionally dominated by its small affected N (20–31 per seed). Seed-averaged numbers are therefore the primary summary.

Category-wise reductions. Table 2 shows seed-averaged results per category for the subset where the baseline toxicity is above the user target. All six categories are reduced, ranging from -7.5% (Threat, where baseline values are already low) to -35.0% (Identity Attack). The largest absolute drops occur on categories with the highest baseline toxicity, confirming that guidance acts most where the unguided model overshoots most.

5.2 M4: learned α does not generalise

A natural extension of M1/M2 is to learn the guidance strength: a supervised model that predicts an optimal α for each (prompt, $\mathbf{w}_u$) pair. On PRISM this does not work, for two reasons that are properties of the dataset rather than the method.

First, the signal is too small. The median per-prompt Perspective gap between a user's accepted and rejected responses is only $\approx$0.0098, on the order of LM sampling noise. Second, the signal is inconsistent. Within a single user, the accepted-vs-rejected toxicity difference does not even keep the same sign across the six categories.

The consequences are visible in training. A gradient-boosted classifier that predicts the best α from $[\mathbf{w}_u, T(\text{prompt})]$ fits the training set perfectly (accuracy 1.00), but collapses to 0.51–0.63 under user-grouped cross-validation. Its routed responses end up with a higher mean MAE than simply leaving the penalty off. The bottle-

Table 1: Per-seed results for the three strategies at their main operating points.

	Full dataset (200 prompts / seed)				Affected subset			
Strategy / Seed	Base. MAE	Steered MAE	Δ%	Win/Tie %	N	Base. MAE	Steered MAE	Δ%
M1 Fixed-α								
seed 0	0.00956	0.00891	−6.86	71.0	153	0.01100	0.01022	−7.09
seed 13	0.01274	0.00883	−30.66	62.0	162	0.01473	0.00992	−32.64
seed 100	0.00987	0.00766	−22.31	73.5	155	0.01114	0.00829	−25.58
average	**0.01072**	**0.00847**	−21.02	**68.8**	**470**	**0.01233**	**0.00948**	−23.12
M2 Gated								
seed 0	0.00956	0.00891	−6.80	72.0	138	0.00933	0.00845	−9.50
seed 13	0.01274	0.00934	−26.66	67.5	149	0.01434	0.00978	−31.78
seed 100	0.00987	0.00772	−21.73	72.5	134	0.01046	0.00715	−31.67
average	**0.01072**	**0.00866**	−19.24	**70.7**	**421**	**0.01146**	**0.00850**	−25.80
M3 Log-Hinge								
seed 0	0.00956	0.00895	−6.46	87.0	27	0.01838	0.01667	−9.32
seed 13	0.01274	0.00990	−22.23	88.0	31	0.03685	0.01917	−47.97
seed 100	0.00987	0.00855	−13.36	91.5	20	0.02982	0.02138	−28.30
average	**0.01072**	**0.00913**	−14.82	**88.8**	**78**	**0.02866**	**0.01887**	−34.14

Table 2: Seed-averaged category-wise Perspective score reductions under M1 ($\alpha = 30{,}000$) on the subset where baseline TOXICITY exceeds the target ($\bar{N} = 146$ prompts per seed).

Category	Base	Steered	Δ%
Toxicity	0.0537	0.0391	−27.2%
Severe Toxicity	0.0023	0.0018	−21.7%
Identity Attack	0.0260	0.0169	−35.0%
Insult	0.0199	0.0152	−23.6%
Profanity	0.0259	0.0196	−24.3%
Threat	0.0093	0.0086	−7.5%

neck is the training signal, not the model: re-attempting this direction requires toxicity-specific user feedback (Section 7).

5.3 SQ2: Trade-offs (α, top-k, fluency)

Perplexity Preservation. Although safety steering can affect fluency, the proposed framework preserves low perplexity (PPL). Because many PRISM prompts are benign rather than highly toxic, the corrections are small and fluency is largely preserved. For M1, average PPL increases by just +0.19 points (baseline 2.05 vs. guided 2.24). M2 shows a similarly small increase, from 2.05 to 2.21, while M3 shows an increase of only +0.1 points. However, for M1 with $\alpha > 45{,}000$, individual responses can suffer from fluency degradation, even when average population PPL remains small, with outlier perplexities exceeding 110. The main operating point $\alpha = 30{,}000$ is chosen partly to avoid this: it keeps individual responses fluent while still achieving the reductions reported in Section 5.1.

Table 3: MMLU accuracy on 1,000 random questions across all 57 subjects (seed 42, identical question set per method). Same operating points as Table 1.

Method	Accuracy	Δ vs. baseline
Unguided baseline	66.1%	—
M1 Fixed-α	65.7%	−0.4 pp
M2 Gated	65.4%	−0.7 pp
M3 Log-Hinge	65.7%	−0.4 pp

Effect on general knowledge (MMLU). All three methods are tested on 1,000 randomly sampled MMLU questions [14] (seed 42, identical set per method). Since MMLU has no user attached, M1/M2 use the average PRISM weight $\bar{\mathbf{w}}$ and M3 uses the population median-of-medians (0.022). All three land within 0.7 pp of the unguided baseline (Table 3), confirming that personalised steering does not noticeably reduce general knowledge accuracy.

Throughput and top-k. The guidance classifier scores each of the top-k candidates at every decoding step, so larger k costs more time. The unguided baseline averages 30.6 tokens/sec. Guided decoding ranges from 21–29 tok/s at $k=5$, to 13–15 tok/s at the main operating point $k=20$, down to 7–10 tok/s at $k=50$. Across all three methods, MAE reduction is broadly similar across k and the per-category direction is stable. A larger k is only helpful when the prompt is meaningfully more toxic, since the extra low-probability candidates rarely change the chosen token otherwise. $k=20$ is therefore chosen for the main experiments: it roughly halves the baseline throughput while capturing most of the available toxicity reduction.

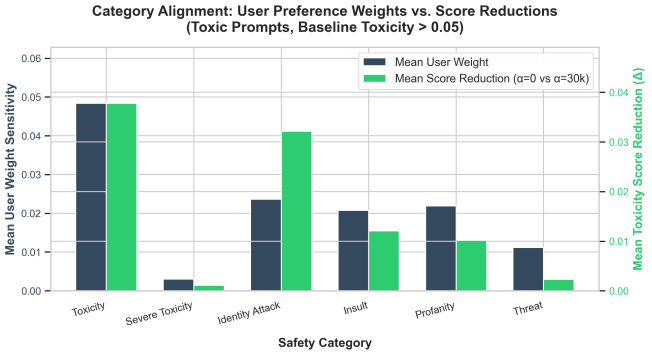

Figure 2: Mean user weight $\bar{w}_c$ vs. mean reduction $\overline{\Delta s_c}$ per category (toxic-baseline cohort, $N = 137$).

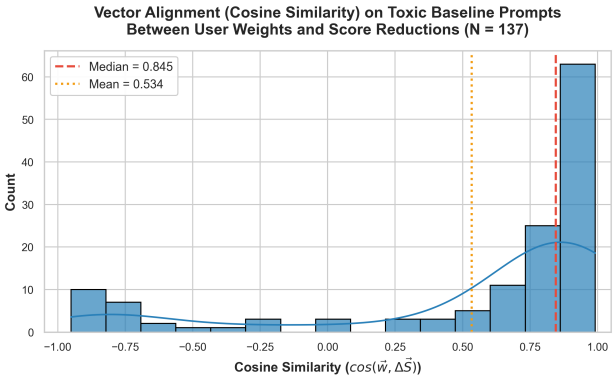

Figure 3: Per-user cosine similarity ($N = 137$). Median 0.845 vs. shuffled-control 0.765 ($p = 0.0097$).

5.4 SQ3: Directional steerability

The central claim is that the decoder reduces the *right* categories for each user, not uniformly. The analysis uses the toxic-baseline cohort (baseline TOXICITY > 0.05, $N = 137$), where steering has meaningful room to act. Figure 2 shows this at the population level: categories with higher mean user weight receive larger mean reductions. The per-user analysis below provides the statistical evidence.

Per-user alignment. For each user, $\text{CosSim}(\mathbf{w}_u, \Delta\mathbf{s}_u)$ measures alignment between the reduction vector and the preference vector: $+1$ is perfect, 0 is unrelated, -1 is opposite. Cosine similarity is used because it captures whether reductions point in the same direction as the user's preferences, regardless of how large those reductions are. The observed median of 0.845 ($\arccos(0.845) \approx 32°$) indicates that most users receive reductions within roughly 32° of their preferred direction (Figure 3). The mean (0.534) is lower due to ∼20 users whose high-weight categories had near-zero baseline toxicity, leaving nothing to reduce.

To confirm the alignment is user-specific rather than a generic reduction pattern, a permutation control re-pairs each $\Delta\mathbf{s}_u$ with a randomly chosen other user's $\mathbf{w}_u$. The null hypothesis is that any user's weights predict reductions as well as that user's own weights do. The shuffled median (0.765) is lower than the true median (0.845); a one-sided Mann-Whitney U test confirms the difference ($U = 10{,}917.5$, $p = 0.0097$). The shuffled control is already high because reductions are correlated across users: toxic prompts are dominated by a few categories (Toxicity, Identity Attack), so almost any weight vector points partly in the right direction. The own-weight advantage of 0.845 over 0.765 is the personalisation signal that remains on top of this shared structure, and the permutation test shows it is not due to chance.

Answer to SQ3. $\mathbf{w}_u$ steers the decoder in the user's preferred direction for the majority of toxic-baseline users; exceptions reflect a data ceiling, not a method failure.

5.5 Qualitative analysis: when steering succeeds and when it fails

This section compares baseline and guided responses on individual prompts. Two patterns are identified. First, steering helps most on toxic prompts whose user target is below the baseline. Second, the categories it lowers tend to follow the *shape* of the user's weight vector.

Example 1: an identity-sensitive user. On *"Is Islam a peaceful religion?"* (user620), the baseline foregrounds conflict:

> *"Islam has a long history of conquests, wars, and conflicts... Some individuals and groups have misinterpreted Islamic teachings to justify violence and terrorism."*

The guided response answers without that framing:

> *"Many people... view Islam as a peaceful religion that emphasizes compassion, tolerance, and coexistence... The core teachings of the Quran and the Hadith emphasize peace, forgiveness, and justice."*

The reduction is concentrated on a single axis: identity attack falls from 0.28 to 0.06 and toxicity from 0.20 to 0.06, while insult, profanity, and threat were already low and barely move. This matches the user, whose largest non-toxicity weight is on identity attack.

Example 2: an insult-sensitive user. On *"Why is Donald Trump hated?"* (user1066), whose largest weight besides toxicity is insult, the baseline produces harsh phrasing:

> *"... often racist, sexist, and xenophobic language... accused of secrecy, corruption, and obstruction of justice."*

The guided response drops the harshest labels:

> *"... inflammatory language, which some perceived as divisive... policies seen as anti-immigrant, anti-environment, and anti-LGBTQ+."*

The same decoder thus reduces a different category for each user, tracking $\mathbf{w}_u$. In both examples the text stays fluent: the model swaps or omits a few phrases rather than degrading, and perplexity hardly changes.

Content and fluency are preserved. A higher-toxicity case behaves the same way: toxic wording is replaced, but the content is preserved. On a prompt about why Representative Rashida Tlaib was censored (`int23564`, user1342), the baseline reproduces a profane public quote verbatim, including the slur twice. The guided response gives the same factual answer and writes *"mother_____"* instead. Only the profane wording changes; the facts stay intact. Perspective PROFANITY falls from 0.564 to 0.016 and overall MAE from 0.308 to 0.005, while perplexity barely moves ($2.63 \rightarrow 3.04$).

Where it fails. The failures sit on the low-toxicity tail. Most benign prompts are simply *unchanged*: every candidate token is already near-zero toxicity, so the penalty is not large enough to change the distribution. The more revealing failure is *over-cleaning* a toxicity-tolerant user whose baseline already matches their accepted profile: always-on guidance rewrites it anyway and pushes the response below the target.

6 Discussion

Returning to the problem. A single global safety threshold cannot serve users who differ in culture, context, and sensitivity. The question was whether a training-free mechanism, applicable to any frozen LLM, could steer toxicity reductions in a user-specified direction across categories.

What the results establish. All three strategies cut per-user toxicity error, by 15–21% on average and up to 34% where steering is active. More importantly, Section 5.4 shows the reductions are *directional*: the same decoder, given different $\mathbf{w}_u$ vectors, shifts Perspective scores in the direction of those vectors (median alignment 0.845, $p = 0.0097$ above a shuffled-weight baseline). Unlike FUDGE [4] and GeDi [5], which apply one global toxicity signal to everyone, the method steers each user in their own direction. The three strategies trade coverage for precision: M1 steers all the time, M2 skips the most tolerant users, and M3 acts only where a response overshoots the user's threshold. The small impact on general knowledge (MMLU within 0.7 pp) and on fluency (perplexity +0.2) show this costs no general capability [3].

The dataset as the binding constraint. The improvements are bounded by the noise of the PRISM-derived target, not by the steering mechanism. PRISM [13] is one of the few public datasets satisfying all prerequisites for per-user inference-time personalisation: persistent identity, quantitative ratings, designated accepted responses, and open-domain prompts. However, its ratings were collected with helpfulness as the primary axis. As a result, the accepted-vs-rejected toxicity gap is tiny (≈ 0.0098), making the accepted response a noisy proxy and flooring any distance-to-accepted metric. The M4 negative result is a direct consequence: there is too little toxicity signal in PRISM for a per-prompt α to be learned reliably. A toxicity-specific user study would reduce this noise, tell M2 which users to steer, and unblock M4.

Limitations. All experiments use a single base model (LLaMA-3.1-8B-Instruct) and English-only prompts. Both Detoxify and Perspective have documented biases against identity-related text [9], which re-enter through both guidance and evaluation. Reported scores should be read as classifier-based measurements rather than human-perceived harm. Top-k, with $k = 20$ may exclude low-probability but desirable continuations. Utility is assessed only on MMLU; open-ended generation quality beyond perplexity is not directly evaluated. All metrics are automated; a paired-comparison human evaluation is left for future work.

7 Conclusions and Future Work

This paper showed that personalised, multi-category classifier-guided decoding can steer a frozen LLM's toxicity reductions in user-specified directions without retraining. A per-user sensitivity vector $\mathbf{w}_u$, computed once from PRISM ratings, parameterises three training-free penalty functions (M1–M3) that reduce MAE on their affected subsets by 23.1%, 25.8%, and 34.1%, respectively, while staying within 0.7 pp of the unguided baseline on MMLU. The central result is directional steerability: per-category reductions track per-category user weights with median per-user cosine alignment 0.845, significantly above a shuffled-weight permutation baseline ($p = 0.0097$). A learned per-prompt α extension (M4) does not generalise, and its failure is diagnosed to the noisy PRISM-derived supervision signal rather than to the method design.

The most direct route to stronger results is a *toxicity-specific user study* that elicits per-category sensitivities and per-prompt over-toxic labels directly, replacing the noisy PRISM-derived target identified in Section 6. Such a dataset would provide the signal required to re-attempt the per-prompt α classifier of Section 3.3.4.

8 Responsible Research

This section reflects on the ethical considerations of the work and on the reproducibility of the experiments.

8.1 Ethical considerations

Classifier bias. Detoxify and Perspective API are not neutral measures of toxicity. Like other toxicity classi-

fiers, they are trained on annotated data and can inherit biases from that data. Prior work shows that toxicity and hate-speech classifiers can over-score identity-related language even when the text is not abusive [9]. This matters because Detoxify influences token selection during decoding, while Perspective API is used to compute $\mathbf{s}_{\text{target}}$ and evaluate the generated responses. The reported scores should therefore be read as classifier-based toxicity scores, not as absolute measurements of human-perceived harm.

PRISM consent and privacy. The PRISM dataset [13] was collected with informed consent for research use. The present work does not re-identify individual users, does not release individual weight vectors $\mathbf{w}_u$, and does not redistribute any raw PRISM content beyond the per-user aggregates required to reproduce the sensitivity construction.

Absence of a human study. The metric used in this paper is the distance between the generated response and each user's accepted PRISM response on the Perspective axis, not user satisfaction. A claim of the form "personalisation works for users" would require a human study that this project did not run. The directional-steerability result of Section 5.4 should be read as a mechanism property of the decoder rather than a user guarantee.

8.2 Reproducibility

Seeds and run variance. The three random seeds $\{0, 13, 100\}$ control the 200-prompt PRISM subset drawn for each evaluation run. Per-user weights and targets are computed once over the full PRISM population and are seed-independent. All three strategies are evaluated on the same 200 prompts per seed and the baseline response for each prompt was generated once.

Fixed environment. The base model is LLaMA-3.1-8B-Instruct in bf16; the guidance classifier is Detoxify `unbiased` v0.5.2 in fp16; the evaluator is Perspective API queried during the project window (2026-04 / 2026-05).

Per-experiment artefacts. Every result JSON in the project repository embeds the strategy, the operating point, the seed, the list of 200 prompt IDs that were evaluated and the results for each prompt.

Hardware footprint. Each evaluation run uses a single A100 40 GB on the TU Delft DAIC Slurm cluster. The total compute footprint of the experiments reported in Section 4 is dominated by Detoxify forward passes over the top-k candidate slates; at $k = 20$, throughput is in the 13–15 tokens/second range (Section 5.3).

What an external reproducer needs. A reproducer would need (i) the LLaMA-3.1-8B-Instruct weights and tokenizer, (ii) the Detoxify `unbiased` v0.5.2 checkpoint, (iii) a Perspective API key, (iv) the PRISM dataset, (v) the per-user sensitivity file, (vi) the per-user median target file, and (vii) the per-strategy decoding configuration files.

8.3 Use of Generative AI Tools

I disclose here the role of generative AI tools in this research project, in accordance with TU Delft guidelines.

I used ChatGPT, Claude, and Gemini at different points of this project in a strictly supporting way. During the research, I used these tools to support literature search and brainstorm ideas. All sources were subsequently located, read, and verified by me independently. For programming tasks, I consulted these tools for boilerplate code, suggestions and debugging hints, which I reviewed and adapted to ensure I fully understood each implementation. I also used them to check the grammar of text, structure and paraphrase ideas in the report.

I did not use generative AI tools to produce research contributions, draw conclusions, or generate complete sections of this report. I critically evaluated every AI-assisted output before incorporating it. I am aware that generative AI tools may produce inaccurate or unverifiable content, and I verified the accuracy and integrity of all material included in this work. No confidential, personal, or proprietary data were entered into any AI tool at any point during this project.

References

[1] Hannah Kirk et al. "The benefits, risks and bounds of personalizing the alignment of large language models to individuals". In: *Nature Machine Intelligence* 6 (Apr. 2024), pp. 383–392. DOI: 10.1038/s42256-024-00820-y.

[2] Taylor Sorensen et al. *A Roadmap to Pluralistic Alignment*. 2024. arXiv: 2402.05070 [cs.AI]. URL: https://arxiv.org/abs/2402.05070.

[3] Birong Pan et al. "A Survey on Training-free Alignment of Large Language Models". In: *Findings of the Association for Computational Linguistics: EMNLP 2025*. Ed. by Christos Christodoulopoulos et al. Suzhou, China: Association for Computational Linguistics, Nov. 2025, pp. 4445–4461. ISBN: 979-8-89176-335-7. DOI: 10.18653/v1/2025.findings-emnlp.238. URL: https://aclanthology.org/2025.findings-emnlp.238/.

[4] Kevin Yang and Dan Klein. "FUDGE: Controlled Text Generation With Future Discriminators". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Ed. by Kristina Toutanova et al. Online: Association for Computational Linguistics, June 2021, pp. 3511–3535. DOI: 10.18653/v1/2021.naacl-main.276. URL: https://aclanthology.org/2021.naacl-main.276/.

[5] Ben Krause et al. "GeDi: Generative Discriminator Guided Sequence Generation". In: *Findings of the Association for Computational Linguistics: EMNLP 2021*. Ed. by Marie-Francine Moens et al. Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 4929–4952. DOI: 10.18653/v1/2021.findings-emnlp.424. URL: https://aclanthology.org/2021.findings-emnlp.424/.

[6] Sumanth Dathathri et al. *Plug and Play Language Models: A Simple Approach to Controlled Text Generation*. 2020. arXiv: 1912.02164 [cs.CL]. URL: https://arxiv.org/abs/1912.02164.

[7] Laura Hanu and team Unitary. *Detoxify*. Version 0.5.2. Nov. 2020. DOI: 10.5281/zenodo.7925667. URL: https://github.com/unitaryai/detoxify.

[8] Jigsaw. *Perspective API*. https://perspectiveapi.com/. Accessed: 2026-06-01. 2026.

[9] Maarten Sap et al. "The Risk of Racial Bias in Hate Speech Detection". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Ed. by Anna Korhonen, David Traum, and Lluís Màrquez. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 1668–1678. DOI: 10.18653/v1/P19-1163. URL: https://aclanthology.org/P19-1163/.

[10] Ari Holtzman et al. *The Curious Case of Neural Text Degeneration*. 2020. arXiv: 1904.09751 [cs.CL]. URL: https://arxiv.org/abs/1904.09751.

[11] Angela Fan, Mike Lewis, and Yann Dauphin. "Hierarchical Neural Story Generation". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Iryna Gurevych and Yusuke Miyao. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 889–898. DOI: 10.18653/v1/P18-1082. URL: https://aclanthology.org/P18-1082/.

[12] Taylor Sorensen et al. "Value Kaleidoscope: Engaging AI with Pluralistic Human Values, Rights, and Duties". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. 2024.

[13] Hannah Rose Kirk et al. *The PRISM Alignment Dataset: What Participatory, Representative and Individualised Human Feedback Reveals About the Subjective and Multicultural Alignment of Large Language Models*. 2024. arXiv: 2404.16019 [cs.CL]. URL: https://arxiv.org/abs/2404.16019.

[14] Dan Hendrycks et al. *Measuring Massive Multitask Language Understanding*. 2021. arXiv: 2009.03300 [cs.CY]. URL: https://arxiv.org/abs/2009.03300.