STEER, THEN CLASSIFY: ADAPTIVE DOCUMENT-SPECIFIC REPRESENTATION STEERING FOR TEXT CLASSIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Text classification is a fundamental natural language processing (NLP) problem, spanning many tasks such as intent and topic classification, toxicity and stance detection, entailment, factuality assessment and many more. Despite substantial advances in large language models (LLMs), modern text classifiers still largely follow the same paradigm, that encode the input with a pretrained language model and train a classification head on top of the resulting representation. This leaves the internal representations of the model largely passive during inference. In this paper, we introduce a different and novel approach to text classification based on representation steering. Rather than directly classifying a fixed document representation, our method dynamically modifies the representation of each input document individually along learned steering directions to make task-relevant information more separable in representation space. Building on this idea, we propose AdaSteer, an adaptive steering framework that determines how and where to intervene in an LLM's representations for each input. Across the Taarof and Multilingual Toxicity Benchmark datasets, AdaSteer consistently improves classification performance over conventional classification heads and existing steering baselines. Notably, when both approaches are initialized from the same underlying LLM, AdaSteer achieves a 1.89% average improvement in F1 score over the standard classifier baseline across four datasets (i.e. on the Irony, Taarof, Multilingual Toxicity and Sentiment datasets), while requiring comparable trainable parameters. Our experiments further show that per-document adaptive steering produces more discriminative representations and that the gains persist across model scales and datasets, providing a complementary alternative to conventional discriminative fine-tuning, and establishing representation steering as a promising paradigm for text classification. Our entire codebase will be released for facilitating reproducibility.

1 INTRODUCTION

Text classification remains an important problem in modern NLP, with many essential applications such as intent classification, stance detection, and factuality detection. The standard approach to text classification has evolved substantially with advances in pretrained language models, such that early feature-based classifiers (Bahrainian & Dengel, 2013) have largely given way to models in which a pretrained Transformer produces contextualized representations that are subsequently mapped to task labels (Devlin et al., 2019). More recently, large language models (LLMs) have introduced an alternative paradigm in which classification can be formulated directly as language generation and performed through prompting (Wei et al., 2023; Liu et al., 2021) or in-context learning (Brown et al., 2020). Despite their generality, however, prompted LLMs do not universally supersede supervised classifiers as shown by a large-scale evaluation across 16 classification datasets that revealed that fine-tuned, substantially smaller language models can outperform zero- and few-shot prompting of larger generative models (Edwards & Camacho-Collados, 2024a). Consequently, improving how pretrained models are specialized for discriminative tasks remains an important problem even in the era of increasingly capable LLMs. Recent approaches to text classification can broadly be viewed as operating at one of two levels: The *first* adapts the model parameters, for example through full

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

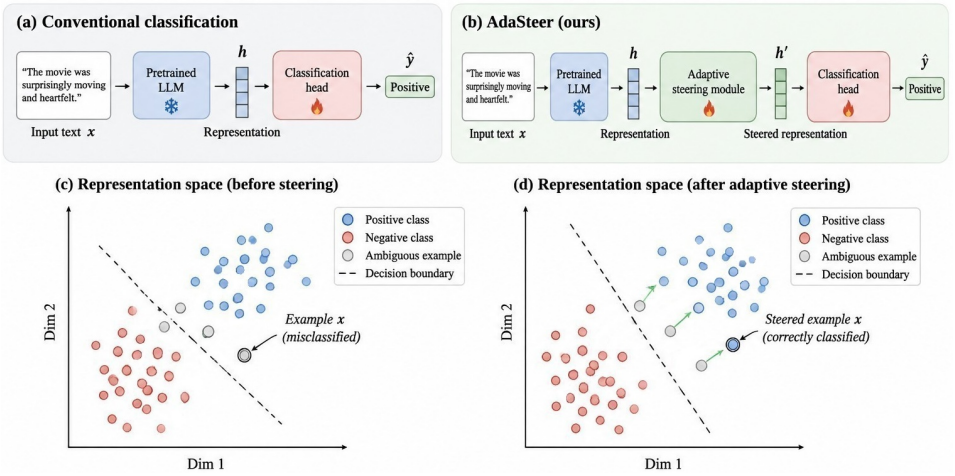

Figure 1: From passive representations to adaptive steering for text classification. (a) Conventional classification uses a pretrained LLM to obtain a fixed representation h and predicts a label with a classification head. (b) AdaSteer learns an input-dependent transformation that steers the representation to h' before classification. (c) In the original representation space, some examples (gray) lie near the decision boundary and may be misclassified. (d) AdaSteer moves such examples toward more discriminative regions, improving class separability and classification accuracy.

fine-tuning or parameter-efficient adaptation, so that its representations become better aligned with a downstream classification objective. The *second* keeps most or all of the language model fixed and learns a mapping from its representations to the target labels (Jiao et al., 2024; Liu et al., 2024).

A common feature of most existing classification approaches, however, is that the representation of an input document is ultimately treated as a fixed object, and a classifier, probe, or decoder determines a label from the representations produced by the underlying model. In this paper, we investigate a novel alternative paradigm: *rather than only learning a better decision function over a fixed representation, can we improve classification by actively adjusting the representation itself?* Intuitively, examples close to a decision boundary may already contain the information required for correct classification, but that information may not be expressed in a form that is sufficiently separable by a conventional classifier. Instead of requiring the classification head to compensate entirely for this geometry, we ask whether the internal representation of an input document can be moved toward a region of representation space that better expresses its task-relevant properties. This perspective connects text classification to a rapidly growing literature on activation and representation steering. Representation engineering (Zou et al., 2025) studies high-level concepts encoded in neural representations and demonstrates that manipulating these representations can systematically influence model behavior.

Existing steering methods have primarily been developed to control generative behavior, for example, truthfulness, sentiment, safety, style, or instruction following (Stolfo et al., 2025), rather than to improve supervised discriminative classification and mostly treat all documents the same way (Braun et al., 2026). Classification poses a different challenge. *The appropriate intervention may depend on the individual example: an input already located far from the decision boundary may require little or no modification, whereas an ambiguous example may benefit from a stronger or qualitatively different intervention.* This motivates our novel document-specific adaptive representation steering for text classification. In this work, we introduce AdaSteer, a framework for text classification that dynamically intervenes on the internal representations of a pretrained language model. Rather than treating the representation produced by the backbone as fixed, AdaSteer learns to transform each example in representation space using a specific weight before the final classification decision. The central intuition is illustrated in Figure 1: given an initial representation $\mathbf{h}$, conventional classification learns a decision boundary over $\mathbf{h}$, whereas AdaSteer additionally learns an input-dependent intervention that moves $\mathbf{h}$ toward a representation that is more discriminative with respect to the target classes. Importantly, the intervention is inferred entirely from the input and learned training signal; it does not require access to the ground-truth label at inference time.

This allows AdaSteer to retain the knowledge encoded by the pretrained model while actively reorganizing how that knowledge is exposed to the classifier.

Our contributions are threefold: (1) For the first time, we introduce representation steering as a mechanism for supervised text classification, shifting the role of internal LLM representations from passive features used by a classifier to representations that can be actively modified to improve discriminative decisions. (2) We propose AdaSteer, an adaptive steering method that learns input-dependent interventions in representation space without requiring ground-truth labels at inference time. (3) through experiments across six datasets and 20 models, we show that AdaSteer consistently outperforms or matches conventional classifiers initialized from the same pretrained LLM. Our codebase will be released for reproducibility.

2 Related Work

Text classification in the post-Transformer (Vaswani et al., 2023) era originally relied on supervised fine-tuning of pretrained encoder models such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019). With the emergence of LLMs, classification has increasingly been approached through prompting (Wei et al., 2023; Zhou et al., 2023; Brown et al., 2020), in-context learning (Brown et al., 2020), and instruction tuning (Wei et al., 2022). Recent empirical studies show that LLMs can perform zero- and few-shot classification across a wide range of tasks, reducing the need for task-specific training data. However, research shows that their performance does not consistently surpass that of smaller fine-tuned models. In particular, large-scale evaluations across diverse classification datasets find that smaller fine-tuned models can remain competitive with, and sometimes outperform, few-shot LLMs (Edwards & Camacho-Collados, 2024b; Lepagnol et al., 2024). More recent studies similarly show that the relative performance of general-purpose LLMs and specialized classifiers depends strongly on the dataset, task, and amount of labeled data available (Pecher et al., 2025). These findings motivate approaches that adapt the capabilities of LLMs to classification without requiring full parameter updates.

Activation Steeting adapts LLMs at inference time, modifying their behaviour for a given task or input without changing their underlying parameters by directly manipulating a model's internal representations. Contrastive Activation Addition (CAA) (Rimsky et al., 2024b) constructs a steering vector by contrasting activations from examples exhibiting different behaviours and adds this vector to the residual stream to steer the model towards the desired behaviour. Recent work has explored different ways of constructing steering directions (Brumley et al., 2024; Dunefsky & Cohan, 2025; Oozeer et al., 2026; Singh et al., 2025), selecting intervention layers (Sun et al., 2025), and applying representation-level interventions to control model behaviour.

Inference-Time Intervention (Li et al., 2023) shows that shifting activations along directions associated with truthfulness can substantially alter model responses without updating the model parameters. Similarly, Contrastive Activation Addition (Rimsky et al., 2024a) derives steering vectors from differences between contrasting examples and adds these vectors to intermediate activations during inference. More recent work has demonstrated that activation steering can improve instruction following (Stolfo et al., 2025) and can be extended to more targeted and adaptive interventions. These findings establish that internal representations are not merely diagnostic signals. They provide an effective interface through which model behavior can be changed at inference time.

However, conventional activation steering typically applies a fixed steering direction and intervention strength across inputs, treating the intervention itself as independent of the individual example. In this paper, we introduce a novel LLM steering approach for document-specific text classification. To the best of our knowledge, AdaSteer is the first approach to learn a document-specific steering strength for classification by using the model's own classification behavior to determine the magnitude of the representation-level intervention.

3 Methods

We introduce *AdaSteer*, a novel document-specific activation steering framework for text classification using decoder-based LLMs. The key idea is to replace a fixed steering strength with a *document-specific adaptive* steering strength that is predicted from the model's classification be-

havior. For each input document, AdaSteer estimates the magnitude of intervention that is most appropriate for that document and applies the corresponding activation shift during inference.

AdaSteer builds on CAA (Rimsky et al., 2024b), which provides the steering direction, while a lightweight predictor determines the steering magnitude. The resulting pipeline consists of two main components: (i) constructing a task-specific steering vector using CAA and (ii) learning a predictor that estimates a document-specific steering strength α from the model's classification logits. Specifically, for each document, we obtain the classifier logits under three conditions: without steering ($\alpha = 0$), with positive steering ($\alpha = +3$), and with negative steering ($\alpha = -3$). These logits provided as input to a small Multilayer Perceptron, which predicts the steering strength α to apply to the document. The predicted α is used to intervene on the hidden representation at inference time.

3.1 CAA Steering Vector

CAA serves as the steering component of AdaSteer. Given a binary classification task, we construct a task-specific steering vector by computing the mean difference between hidden-state representations associated with the two classes. We intervene at layer $l = n - 2$, where n denotes the total number of hidden layers. Since the classification datasets do not provide naturally paired contrastive examples, we construct contrastive pairs using sequence length as a matching criterion. We first oversample the minority class with replacement (Chawla et al., 2002) and then greedily pair each example with the closest unmatched example from the opposite class in terms of sequence length (Rosenbaum & Rubin, 1985). Specifically, for sample i, its contrastive counterpart is selected as $j_i = \arg\min_{j \in \mathcal{U}} |s_j - s_i|$, where s_i denotes the sequence length of sample i and $\mathcal{U}$ is the set of currently unmatched samples from the opposite class. The steering vector is then computed as the average difference between the paired hidden representations, and added to the hidden representation with steering strength α during inference:

$$\mathbf{v} = \frac{1}{N} \sum_{i=1}^{N} \left(\mathbf{h}_i^{(l)} - \mathbf{h}_{j_i}^{(l)} \right), \quad \mathbf{h}^{(l)} \leftarrow \mathbf{h}^{(l)} + \alpha \mathbf{v}.$$

Thus, CAA determines the *direction* of the intervention, whereas α determines its *magnitude*. AdaSteer focuses on learning the latter in a document-specific manner.

3.2 Document-Specific Steering Strength

A fixed steering strength applies the same intervention magnitude to every document, although different documents may require different degrees of steering. AdaSteer learns a function $\hat{\alpha} = f(\mathbf{x})$ that predicts a document-specific steering strength from the model's classification behavior.

3.2.1 Ground-Truth Steering Strength

To train the α predictor, we first construct a continuous target for each training document. For each document, we evaluate the classifier under eight steering strengths $\mathcal{A} = \{\alpha_1, \ldots, \alpha_8\} \subset [-1, 1]$, and obtain the corresponding probability assigned to the target class c:

$$p_c(\alpha_k) = \frac{\exp(z_c(\alpha_k))}{\sum_{j=1}^{C} \exp(z_j(\alpha_k))}.$$

Rather than selecting the single steering strength that produces the highest probability, we use a soft weighting scheme to obtain a continuous target. We compute the weights and define the target steering strength as the weighted average,

$$w_k = \frac{\exp(\tau p_c(\alpha_k))}{\sum_{i=1}^{K} \exp(\tau p_c(\alpha_i))}, \quad \alpha^* = \sum_{k=1}^{K} w_k \alpha_k.$$

This produces a smooth, document-specific target that reflects the model's classification behavior across multiple steering strengths.

3.2.2 DOCUMENT REPRESENTATIONS FOR STEERING PREDICTION

At inference, AdaSteer must estimate the appropriate steering strength without evaluating the document over the full set of candidate α values. We use the classifier logits obtained from three forward passes with $\alpha = 0, +3, -3$ as a compact representation of the document's response to steering.

For binary classification, concatenating the logits from these three conditions yields a six-dimensional feature vector $\mathbf{x}_{\text{dyn}} = [\mathbf{z}_0, \mathbf{z}_+, \mathbf{z}_-]$.

We additionally investigate an augmented representation that includes the element-wise mean of the three logit vectors, $\mathbf{z}_{\text{avg}} = \frac{1}{3}(\mathbf{z}_0 + \mathbf{z}_+ + \mathbf{z}_-), \quad \mathbf{x}_{\text{avg}} = [\mathbf{z}_0, \mathbf{z}_+, \mathbf{z}_-, \mathbf{z}_{\text{avg}}]$.

The two feature configurations allow us to assess whether explicitly summarizing the model's average response to steering provides additional information for predicting the appropriate intervention strength.

3.2.3 STEERING STRENGTH PREDICTION

We predict the document-specific steering strength using a lightweight Multilayer Perceptron. For the augmented feature configuration, the input dimension is increased to account for the additional averaged-logit features. Training details and hyperparameters are provided in Appendix A.2.

At inference, the predictor produces $\hat{\alpha} = f(\mathbf{x})$, which is applied to the CAA steering vector $\mathbf{h}^{(l)} \leftarrow \mathbf{h}^{(l)} + \hat{\alpha}\mathbf{v}$. Thus, CAA determines the task-specific steering direction, while the AdaSteer predictor determines the intervention magnitude for each document.

3.3 ADASTEER VARIANTS

We evaluate four AdaSteer configurations obtained by combining two predictor feature representations with two model adaptation settings. The first dimension determines whether the predictor uses only the three steering-dependent logit representations or additionally includes their element-wise average. The second dimension determines whether the underlying classifier uses a frozen LLM or LoRA adaptation. Here are the four variations:

- **AdaSteer-Dynamic**: Uses the standard six-dimensional feature representation $[\mathbf{z}_0, \mathbf{z}_+, \mathbf{z}_-]$ with a frozen LLM.
- **AdaSteer-Average**: Extends AdaSteer-Dynamic by additionally providing $\mathbf{z}_{\text{avg}}$ to the α predictor.
- **AdaSteer-Dynamic-LoRA**: Uses the standard six-dimensional feature representation together with LoRA adaptation of the underlying LLM.
- **AdaSteer-Average-LoRA**: Combines the augmented feature representation with LoRA adaptation.

Across all variants, the overall procedure remains unchanged: CAA provides the steering direction, the classifier provides the steering-dependent predictions used by the α predictor, and the predictor produces a document-specific steering magnitude for the final activation intervention.

4 EVALUATION

4.1 EVALUATION METRICS

Model Performance Metrics: We evaluate classification performance using accuracy, F1 score, ROC and AUROC. Additionally, we run statistical significance test to compare the baseline and the AdaSteer variants. In order to examine the statistical significance of the results, we use McNemar's test (McNemar, 1947), which is the preferred standard test to determine if the gains of AdaSteer predictions over baselines are statistically significant.

Latency Tests: We measure the computational needs of AdaSteer relative to the baseline using total inference time, latency per sample, and throughput $T_{\text{total}} = T_{\text{baseline}} + T_{\text{vector}} + T_{\text{AdaSteer}}$.

Latency and throughput are computed as $L_{\text{sample}} = \frac{T_{\text{total}}}{N} \times 1000, \text{Throughput} = \frac{N}{T_{\text{total}}}$, reported in seconds per sample and samples per second, respectively.

Activation Patching: We conduct an interpretability study of AdaSteer with the baseline model to better understand their internal representations and how steering influences the models' internal structures. Activation patching measures the direct Normalized Indirect Effect (NIE) of internal components by swapping hidden states between clean (x_{clean}) and corrupted (x_{corrupt}) execution passes. Comparing NIE matrices across unsteered ($\mathbf{M}_{\text{Baseline}}$), standard steered ($\mathbf{M}_{\text{AdaSteer-Dynamic}}$), and averaged steered ($\mathbf{M}_{\text{AdaSteer-Average}}$) passes isolates the specific attention heads responsible for propagating steering-induced behavioral shifts.

4.2 Datasets

We evaluate AdaSteer on 6 text classification datasets covering Taarof, politeness, sentiment, irony, and toxicity detection. For (1) Taarof, we use the *We Politely Insist: Your LLM Must Learn the Persian Art of Taarof* dataset (Sadr et al., 2025), where each instance contains a context and utterance labelled as Taarof or non-Taarof; both are used as input. For (2) Politeness, we use the *Stanford Politeness Corpus* (Danescu-Niculescu-Mizil et al., 2013), converting continuous politeness scores into binary polite/impolite labels. For (3) Sentiment, we use Zhang et al. (Zhang et al., 2024), removing neutral samples to obtain a binary positive/negative task and oversampling the negative class during training. For (4) Irony, we use the *SemEval-2018 Task 3: Irony Detection in English Tweets* dataset (Van Hee et al., 2018), which contains balanced ironic and non-ironic tweets. Finally, for toxicity, we use the (5) *Toxicity General Domain* dataset (Surge AI, 2021) and (6) the *Multilingual Toxicity Detection Dataset* (Bevendorff et al., 2024), restricting the latter to English samples.

4.3 Preprocessing

For the Taarof dataset, the Context and Utterance fields are concatenated with a space separator, while all other datasets use their text field directly. Inputs are tokenised using each model's tokenizer and the EOS token is used for padding when no dedicated padding token is available. We use right padding for Gemma and left padding for Llama, Qwen, and GPT-2, following their respective implementations (Vaswani et al., 2023; Radford et al., 2019; Touvron et al., 2023; Bai et al., 2023). Since the classifier uses the hidden representation of the final non-padding token, this padding strategy ensures that the selected representation corresponds to the end of the actual input sequence.

4.4 Experimental Results

Classification Performance Experiment: We experimented on six datasets and 20 decoder-based LLMs (124M–9.2B parameters) from 4 different model families (Llama 3.2, Gemma 2, GPT-2, and Qwen 2.5) Table 5. We present the results on the Taarof dataset in the main text and move the other results to the Appendix A.3 due to space constraints. Tables 1 and 2 report Accuracy and F1 for the baseline, AdaSteer-Dynamic, and AdaSteer-Average variants with and without LoRA fine-tuning across all evaluated models on the Taarof dataset, together with the statistical significance of the observed differences from the baseline. The results for the other datasets are illustrated in Tables 9, 4,11, 13, 15, 8, 3,10, 12 and 14. The ROC curves are illustrated in Figures 4, 6, 9, 12, 15 and, 18.

Number of Samples for CAA Vector Computation Experiment: To investigate the effect of the amount of data used to construct the steering direction, we compute separate steering vectors using 20%, 40%, 60%, 80%, and 100% of the training data. Figure 2 shows the performance results on the Taarof dataset with a broader evaluation of the No-LoRA setting, including performance across different CAA scales as well as inference latency and throughput. The results for the other datasets are illustrated in Figures 7, 10, 13, 16 and 19.

Interpretability Study: In order to analyse the behavior of AdaSteer, we perform activation patching. Figure 3 provides an interpretability analysis using activation patching, averaged across all datasets, illustrating how the contribution of different model components changes under steering. As seen, AdaSteer-Dynamic and AdaSteer-Average have more positive weights closer to 1 than the baseline, indicating that steering increases the contribution of these model components to the model's classification behavior. This suggests that AdaSteer does not merely shift the final decision, but instead modifies the internal representations used by the model to make its predictions. The in-

creased contribution of these components under steering provides evidence that the learned steering directions alter the model's internal computation in a task-relevant manner. The activation patching results for the other datasets are illustrated in Figures 5,8, 14, 17 and 20

Model	Baseline	Dynamic-LoRA	Average-LoRA
Gpt-2	67.4 / 62.9	**68.9** / 61.5	37.8 / 37.0
Gpt-2-Medium	31.9 / 30.2	**50.4 / 45.6***	**62.2 / 52.9***
Gpt-2-Large	40.0 / 39.2	35.6 / 33.5	**65.9 / 63.2***
Gpt-2-XL	63.0 / 58.8	**71.9*** / 58.3	58.5 / 54.7
Gemma-2B	80.7 / 76.2	80.0 / 75.8	80.7 / **76.9**
Gemma-2B-IT	68.9 / 67.6	**83.0 / 79.4***	**83.0 / 79.1***
Gemma-9B	54.8 / 54.8	**85.2 / 82.0***	**85.9 / 83.3***
Gemma-9B-IT	65.9 / 65.2	**86.7 / 83.3***	**88.1 / 85.1***
Llama-1	67.4 / 65.8	**77.0 / 72.7***	**77.8 / 74.7**
Llama-1-IT	70.4 / 68.5	28.1 / 22.0	**71.1 / 69.2**
Llama-3	72.6 / 70.5	**84.4 / 79.6***	**83.0 / 78.4**
Llama-3-IT	65.2 / 63.6	**82.2 / 77.3***	**85.2 / 79.9***
Qwen-0.5	43.0 / 42.8	**76.3 / 66.5***	30.4 / 25.4
Qwen-0.5-IT	76.3 / 68.5	69.6 / 63.8	71.1 / 65.6
Qwen-1.5	79.3 / 73.9	**81.5** / 71.5	**83.0 / 78.0***
Qwen-1.5-IT	76.3 / 73.6	**83.0 / 77.6***	**83.7 / 77.9***
Qwen-3	78.5 / 73.7	75.6 / 70.0	55.6 / 55.1
Qwen-3-IT	70.4 / 69.5	**71.9** / 69.3	**72.6 / 70.2**
Qwen-7B	85.2 / 81.4	74.1 / 72.5	73.3 / 71.6
Qwen-7B-IT	82.2 / 79.8	80.7 / 73.4	80.0 / 71.4

Table 1: Performance on **Taarof** (LoRA configuration, 100% training data) reported as Accuracy / Macro F1 (%). Baseline uses no steering; Dynamic-LoRA and Average-LoRA apply adaptive and average steering. **Bold** indicates improvement over baseline; * denotes significance via exact two-sided McNemar test ($p < 0.05$).

Model	Baseline	Dynamic	Average
Gpt-2	72.6 / 65.8	28.1 / 22.0	28.1 / 22.0
Gpt-2-Medium	66.7 / 54.4	**68.9 / 55.0**	**67.4** / 51.7
Gpt-2-Large	51.8 / 51.7	**61.5 / 59.3***	**62.2 / 60.0***
Gpt-2-XL	70.4 / 57.1	65.2 / **60.6**	62.2 / **57.7**
Gemma-2B	78.5 / 73.7	77.8 / **74.7**	76.3 / 73.3
Gemma-2B-IT	64.4 / 63.2	**80.7 / 76.9***	**80.0 / 75.8***
Gemma-9B	83.0 / 79.7	83.0 / **80.0**	**88.1 / 85.1**
Gemma-9B-IT	64.4 / 63.9	**88.1 / 85.1***	**88.9 / 85.9***
Llama-1	80.7 / 76.2	54.1 / 53.9	59.3 / 58.7
Llama-1-IT	72.6 / 70.5	31.9 / 27.5	30.4 / 26.0
Llama-3	83.0 / 79.7	**83.7** / 79.2	**84.4 / 80.6**
Llama-3-IT	61.5 / 60.7	**80.7 / 76.6***	**80.7 / 76.2***
Qwen-0.5	71.9 / 70.3	31.1 / 26.5	28.1 / 22.7
Qwen-0.5-IT	65.9 / 65.1	28.1 / 22.0	42.2 / 41.8
Qwen-1.5	70.4 / 68.7	**82.2 / 75.4***	**80.7 / 72.8***
Qwen-1.5-IT	65.9 / 65.1	51.1 / 51.1	52.6 / 52.6
Qwen-3	84.4 / 79.6	81.5 / 77.9	80.0 / 77.1
Qwen-3-IT	67.4 / 65.3	42.2 / 42.2	**73.3 / 70.3**
Qwen-7B	82.2 / 76.9	82.2 / 74.9	**83.0 / 78.8***
Qwen-7B-IT	80.0 / 77.4	64.4 / 62.7	75.6 / 70.9

Table 2: Performance on **Taarof** (No-LoRA configuration, 100% training data) reported as Accuracy / Macro F1 (%). Baseline uses no steering; Dynamic and Average apply adaptive and average steering. **Bold** indicates improvement over baseline; * denotes significance via McNemar test ($p < 0.05$).

Model	Base	Dynamic-LoRA	Average-LoRA
Gpt-2	52.1 / 37.4	51.8 / 36.3	**52.4 / 38.0***
Gpt-2-Medium	50.7 / 33.7	**52.6 / 46.5***	**52.5 / 41.5***
Gpt-2-Large	50.7 / 34.0	50.7 / 33.8	**50.8 / 34.1**
Gpt-2-XL	75.7 / 75.0	71.0 / 69.1	60.2 / 53.5
Gemma-2B	72.2 / 65.4	**73.7 / 70.7**	**75.1 / 71.8**
Gemma-2B-IT	72.6 / 65.2	**78.1 / 75.1***	**77.2 / 74.2***
Gemma-9B	73.7 / 67.0	**79.7 / 77.1***	**79.3 / 76.7***
Gemma-9B-IT	73.3 / 66.1	**79.1 / 75.5***	**79.7 / 76.4***
Llama-1	63.8 / 46.9	60.3 / 37.6	60.3 / 37.6
Llama-1-IT	66.3 / 52.9	60.3 / 37.6	60.3 / 37.6
Llama-3	66.2 / 51.9	58.3 / **56.9**	56.9 / **55.4**
Llama-3-IT	68.5 / 57.2	**77.2 / 76.2***	**75.0 / 73.2***
Qwen-0.5	60.3 / 37.6	59.1 / **44.6**	58.7 / **44.1**
Qwen-0.5-IT	60.0 / 39.2	57.9 / **56.6**	58.8 / **57.4**
Qwen-1.5	60.8 / 40.4	39.7 / 28.4	39.7 / 28.4
Qwen-1.5-IT	61.0 / 39.4	60.3 / 37.6	**63.5 / 48.2***
Qwen-3	60.8 / 39.3	**67.1 / 55.5***	**66.6 / 55.4***
Qwen-3-IT	61.7 / 41.6	**67.5 / 57.7***	**64.4 / 50.9***
Qwen-7B	64.9 / 50.5	**67.9 / 64.9**	**65.2 / 54.3**
Qwen-7B-IT	62.5 / 45.0	61.2 / **51.8**	62.2 / **52.3**

Table 3: Performance on **Irony** (LoRA configuration, 100% training data) reported as Accuracy / Macro F1 (%). Baseline uses no steering; Dynamic-LoRA and Average-LoRA apply adaptive and average steering. **Bold** indicates improvement over baseline; * denotes significance via exact two-sided McNemar test ($p < 0.05$).

Model	Base	Dynamic	Average
Gpt-2	61.0 / 59.8	**62.1** / 59.6	**61.5** / 59.0
Gpt-2-Medium	51.2 / 36.5	51.1 / 35.5	51.1 / 35.5
Gpt-2-Large	59.7 / 56.7	**61.3 / 57.6***	**61.0 / 57.4***
Gpt-2-XL	69.0 / 68.5	**71.7 / 71.3***	**72.3 / 71.9***
Gemma-2B	65.8 / 52.4	**71.3 / 66.2***	**68.2 / 60.2**
Gemma-2B-IT	67.3 / 55.9	**74.5 / 70.7***	**73.2 / 69.0***
Gemma-9B	68.6 / 59.4	**74.9 / 71.7***	**74.6 / 71.4***
Gemma-9B-IT	69.6 / 59.7	**76.1 / 71.5***	**76.7 / 72.2***
Llama-1	61.6 / 41.0	60.3 / 37.6	60.3 / 37.6
Llama-1-IT	66.2 / 52.5	60.3 / 37.6	60.3 / 37.6
Llama-3	62.5 / 43.3	49.9 / **45.5**	49.1 / **44.5**
Llama-3-IT	64.2 / 47.3	**67.7 / 67.7**	**67.7 / 67.7**
Qwen-0.5	61.9 / 55.2	46.4 / 46.0	46.1 / 45.6
Qwen-0.5-IT	63.1 / 50.6	60.0 / **52.2**	59.7 / **51.8**
Qwen-1.5	60.8 / 39.0	39.9 / 28.9	**66.4 / 62.6***
Qwen-1.5-IT	62.6 / 44.1	60.3 / 37.6	60.3 / 37.6
Qwen-3	61.7 / 42.4	**69.0 / 67.7***	**68.8 / 67.5***
Qwen-3-IT	62.1 / 42.4	**67.0 / 58.3***	**63.0 / 61.5**
Qwen-7B	67.0 / 56.8	60.7 / 39.0	60.7 / 38.7
Qwen-7B-IT	69.4 / 60.0	61.0 / **60.6**	60.8 / **60.5**

Table 4: Performance on **Irony** (No-LoRA configuration, 100% training data) reported as Accuracy / Macro F1 (%). Baseline uses no steering; Dynamic and Average apply adaptive and average steering. **Bold** indicates improvement over baseline; * denotes significance via McNemar test ($p < 0.05$).

5 Discussion and Key Takeaways

Based on 480 experiment runs, we draw the following key takeaways:

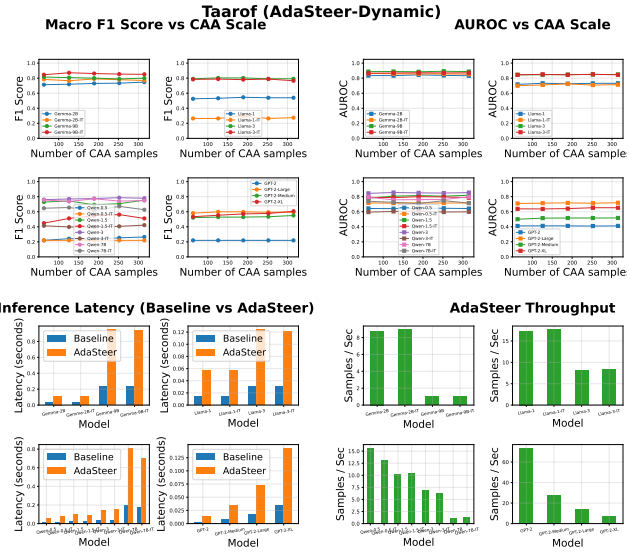

Figure 2: Results for the Taarof (No-LoRA) dataset. The top row illustrates F1 scores and AUROC metrics plotted across varying Contrastive Activation Addition (CAA) scales for the Gemma, Llama, Qwen, and Gpt-2model families. The bottom-left section illustrates latency in seconds between unsteered baselines and AdaSteer interventions across architectures, while the bottom-right displays system throughput measured in samples per second for each evaluated model configuration.

- **Comparison between Base and Instruction-Tuned Models:** Instruction-tuned models generally benefit more from dynamic steering than the baselines as seen in Tables 2, 9, 4,11, 13 and 15.
- **Effect of Model Size on Steering Performance:** Larger models benefit more consistently from AdaSteer as seen in Tables 2, 9, 4,11, 13 and 15, this may be due to more structured representation spaces.
- **Task-Specific Analysis of Steering Effectiveness:** Tasks such as irony and taarof benefit most from intervention due to contextual cues, whereas sentiment and toxicity show smaller gains from strong lexical signals and safety-oriented post-training as seen in Tables 2, 9, 4,11, 13 and 15.
- **Comparison Across Model Families:** GPT-2, Gemma, and Llama exhibit stable steering improvements, whereas Qwen shows higher variability as seen in Tables 2, 9, 4,11, 13 and 15.
- **Effect of LoRA Fine-Tuning on Performance:** LoRA fine-tuning improves dynamic steering for complex tasks, though smaller models benefit less due to limited representational capacity as shown in 1, 8, 3,10, 12 and 14.
- **Effect of Number of Samples for CAA Vector Computation:** Steering performance remains stable across different sample sizes, indicating robust CAA vector estimation as shown in Figures 2,7, 10, 13, 16 and 19.
- **Comparison between AdaSteer-Dynamic and AdaSteer-Average:** Training the alpha predictor with averaged features reduces noise sensitivity and improves dynamic steering reliability across multiple datasets and configurations Figures 5,8, 11, 14, 17 and 20 and Tables 2, 9, 4,11, 13 and 15.
- **Latency Tests:** Dynamic steering introduces a small computational overhead, resulting in a modest increase in latency as shown in Figures 2,7, 10, 13, 16 and 19.
- **Interpretability Tests:** Activation patching shows that steering systematically shifts internal representations in directions associated with improved predictions as seen in Figures 5,8, 14, 17 and 20.
- **Overall Analysis of Results**: Based on F1, accuracy, ROC curves and AUROC, AdaSteer-Dynamic and AdaSteer-Average outperform the corresponding model baselines for the Taarof, Irony, Sentiment, Toxicity General Domain and Multilingual Toxicity datasets for both the LoRA

and no-LoRA settings. AdaSteer matches the performance in the Politeness dataset, where the baseline performance is already high and there is small room for improvement.

- **Comparison to Existing Baselines**: AdaSteer via LoRA fine-tuning yields competitive results: sentiment accuracies match lower-end zero-shot baselines (though remaining below fully fine-tuned models like T5-large (Zhang et al., 2024)), irony detection scores (such as Gemma-9 achieving an accuracy of 0.7883 and F1 of 0.759) rival recent transformer systems, and multilingual toxicity performance exceeds 97% accuracy, nearing fine-tuned XLM-RoBERTa-large (Van Hee et al., 2018). Furthermore, while state-of-the-art LLMs like GPT-4o struggle below 42% on Taarof-Bench (Sadr et al., 2025), Gemma-9 with AdaSteer variants reaches 88.89% accuracy, highlighting activation steering's effectiveness in improving open-weight models and enhancing culturally nuanced understanding.

6 Conclusions and Future Work

In this paper, we propose AdaSteer, a novel text classification framework that adjusts the LLM representation dynamically and customizes it for each input document that is to be classified. This novel method that we introduce for the first time is a departure from passive and fixed document representations and presents a new view of building representation that are document-specific and can dynamically change to better represent each document such that ambiguous texts can be correctly classified. Our experimental results on six datasets and four model families, namely Llama 3.2, Qwen 2.5, Gemma 2 and GPT-2 and on 20 different models (i.e. varying in size) reveal that our proposed AdaSteer model outperforms (i.e. on the Sentiment, Irony, Taarof, Toxicity General Domain and Multilingual Toxicity datasets) or matches (i.e. on the Politeness dataset) the baselines when both categories of models are initialized using the same model. Based on these results, we conclude that dynamic Activation Steering via our proposed AdaSteer can greatly enhance document classification performance. Furthermore, through an activation patching interpretability study we find that on average AdaSteer is more inclined to shift the LLM activations to more neutral and slight positive weights rather than the baseline models which mostly keep the activation weights in negative values.

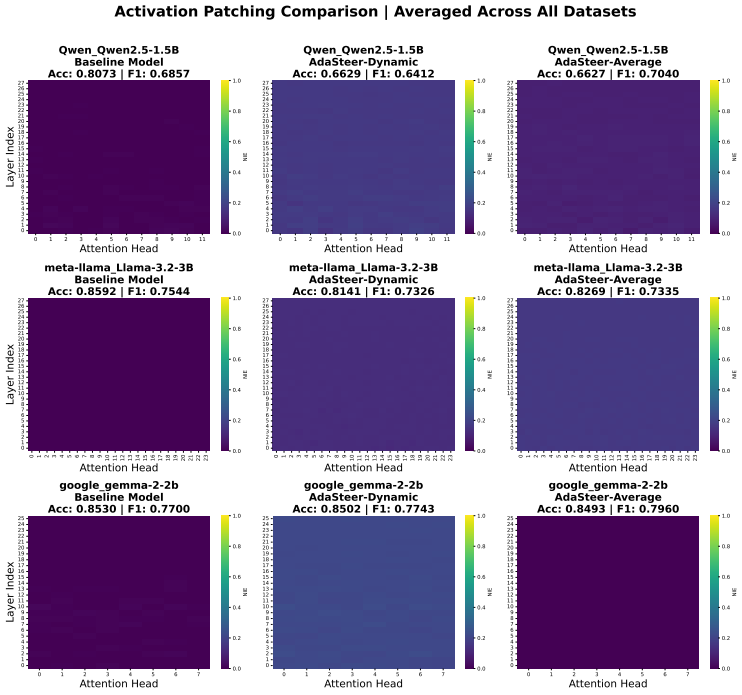

Figure 3: Activation patching analysis comparing the baseline, AdaSteer-Dynamic, and AdaSteer-Average methods across three models: Qwen-1.5, Llama-3, and Gemma-2 averaged over all six datasets.

One possible direction for future work is to explore alternative steering approaches including controlling the steering strength via a learned distribution. Another interesting future direction, is to extend the interpretability study of AdaSteer to reveal the information flow differences as compared with baselines.

AI USE STATEMENT

In this work, we used generative AI tools for minor coding assistance and minor spelling and grammar checking. We did not use generative AI tools to generate research ideas, design the methodology, conduct experiments, analyze or interpret results, generate figures or tables, or write the scientific content of the manuscript. All AI-assisted code was reviewed, manually inspected, and tested by the authors to verify its correctness. Any AI-assisted language corrections were also reviewed by the authors.

REPRODUCIBILITY STATEMENT

We provide details to facilitate the reproduction of our experiments throughout the paper and supplementary materials. The main paper describes the AdaSteer method and experimental setup, while the appendix provides additional implementation details, model configurations, hyperparameters, and experimental results. We use publicly available datasets and describe the corresponding dataset processing and evaluation protocols.

REFERENCES

Seyed-Ali Bahrainian and Andreas Dengel. Sentiment analysis using sentiment features. In *2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, volume 3, pp. 26–29. IEEE, 2013.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. Qwen technical report, 2023. URL https://arxiv.org/abs/2309.16609.

Janek Bevendorff, Xavier Bonet Casals, Berta Chulvi, Daryna Dementieva, Ashaf Elnagar, Dayne Freitag, Maik Fröbe, Damir Korencic, Maximilian Mayerl, Animesh Mukherjee, Alexander Panchenko, Martin Potthast, Francisco Rangel, Paolo Rosso, Alisa Smirnova, Efstathios Stamatatos, Benno Stein, Mariona Taulé, Dmitry Ustalov, Matti Wiegmann, and Eva Zangerle. Overview of PAN 2024: Multi-author writing style analysis, multilingual text detoxification, oppositional thinking analysis, and generative AI authorship verification - extended abstract. In Nazli Goharian, Nicola Tonellotto, Yulan He, Aldo Lipani, Graham McDonald, Craig Macdonald, and Iadh Ounis (eds.), *Advances in Information Retrieval - 46th European Conference on Information Retrieval, ECIR 2024, Glasgow, UK, March 24-28, 2024, Proceedings, Part VI*, volume 14613 of *Lecture Notes in Computer Science*, pp. 3–10. Springer, 2024. doi: 10.1007/978-3-031-56072-9_1. URL https://doi.org/10.1007/978-3-031-56072-9_1.

Joschka Braun, Carsten Eickhoff, and Seyed Ali Bahrainian. Beyond multiple choice: Evaluating steering vectors for summarization. In Vera Demberg, Kentaro Inui, and Lluís Marquez (eds.), *Findings of the Association for Computational Linguistics: EACL 2026*, pp. 3849–3884, Rabat, Morocco, March 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.findings-eacl.200. URL https://aclanthology.org/2026.findings-eacl.200/.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. URL https://arxiv.org/abs/2005.14165.

Madeline Brumley, Joe Kwon, David Krueger, Dmitrii Krasheninnikov, and Usman Anwar. Comparing bottom-up and top-down steering approaches on in-context learning tasks, 2024. URL https://arxiv.org/abs/2411.07213.

Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. SMOTE: synthetic minority over-sampling technique. *J. Artif. Intell. Res.*, 16:321–357, 2002. doi: 10.1613/JAIR.953. URL https://doi.org/10.1613/jair.953.

Cristian Danescu-Niculescu-Mizil, Moritz Sudhof, Dan Jurafsky, Jure Leskovec, and Christopher Potts. A computational approach to politeness with application to social factors. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 250–259, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL https://aclanthology.org/P13-1025/.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pp. 4171–4186. Association for Computational Linguistics, 2019. doi: 10.18653/V1/N19-1423. URL https://doi.org/10.18653/v1/n19-1423.

Jacob Dunefsky and Arman Cohan. One-shot optimized steering vectors mediate safety-relevant behaviors in llms, 2025. URL https://arxiv.org/abs/2502.18862.

Aleksandra Edwards and Jose Camacho-Collados. Language models for text classification: Is in-context learning enough? In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pp. 10058–10072. ELRA and ICCL, 2024a. URL https://aclanthology.org/2024.lrec-main.879/.

Aleksandra Edwards and Jose Camacho-Collados. Language models for text classification: Is in-context learning enough? In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pp. 10058–10072, Torino, Italy, May 2024b. ELRA and ICCL. URL https://aclanthology.org/2024.lrec-main.879/.

Allen L. Edwards. Note on the "correction for continuity" in testing the significance of the difference between correlated proportions. *Psychometrika*, 13(3):185–187, 1948. ISSN 1860-0980. doi: 10.1007/BF02289261. URL https://doi.org/10.1007/BF02289261.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. URL https://arxiv.org/abs/2106.09685.

Difan Jiao, Yilun Liu, Zhenwei Tang, Daniel Matter, Jürgen Pfeffer, and Ashton Anderson. SPIN: Sparsifying and integrating internal neurons in large language models for text classification. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 4666–4682. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-acl.277. URL https://aclanthology.org/2024.findings-acl.277/.

Pierre Lepagnol, Thomas Gerald, Sahar Ghannay, Christophe Servan, and Sophie Rosset. Small language models are good too: An empirical study of zero-shot classification, 2024. URL https://arxiv.org/abs/2404.11122.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, 2023.

Chun Liu, Hongguang Zhang, Kainan Zhao, Xinghai Ju, and Lin Yang. LLMEmbed: Rethinking lightweight LLM's genuine function in text classification. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7994–8004. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.433. URL https://aclanthology.org/2024.acl-long.433/.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing, 2021. URL https://arxiv.org/abs/2107.13586.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019. URL http://arxiv.org/abs/1907.11692.

Quinn McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, June 1947. doi: 10.1007/BF02295996. URL https://doi.org/10.1007/BF02295996.

Narmeen Oozeer, Luke Marks, Shreyans Jain, Fazl Barez, and Amirali Abdullah. Beyond linear steering: Unified multi-attribute control for language models, 2026. URL https://arxiv.org/abs/2505.24535.

Branislav Pecher, Ivan Srba, and Maria Bielikova. Comparing specialised small and general large language models on text classification: 100 labelled samples to achieve break-even performance. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 165–184. Association for Computational Linguistics, 2025.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019. URL https://api.semanticscholar.org/CorpusID:160025533.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks, 2019. URL https://arxiv.org/abs/1908.10084.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15504–15522. Association for Computational Linguistics, 2024a. doi: 10.18653/v1/2024.acl-long.828. URL https://aclanthology.org/2024.acl-long.828/.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. Steering llama 2 via contrastive activation addition. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 15504–15522. Association for Computational Linguistics, 2024b. doi: 10.18653/V1/2024.ACL-LONG.828. URL https://doi.org/10.18653/v1/2024.acl-long.828.

Paul R. Rosenbaum and Donald B. Rubin. Constructing a control group using multivariate matched sampling methods that incorporate the propensity score. *The American Statistician*, 39(1):33–38, 1985. ISSN 00031305, 15372731. URL http://www.jstor.org/stable/2683903.

Nikta Gohari Sadr, Sahar Heidariasl, Karine Megerdoomian, Laleh Seyyed-Kalantari, and Ali Emami. We politely insist: Your LLM must learn the persian art of taarof. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025, Suzhou, China, November 4-9, 2025*, pp. 1819–1838. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.EMNLP-MAIN.94. URL https://doi.org/10.18653/v1/2025.emnlp-main.94.

Shashwat Singh, Shauli Ravfogel, Jonathan Herzig, Roee Aharoni, Ryan Cotterell, and Ponnurangam Kumaraguru. Representation surgery: Theory and practice of affine steering, 2025. URL https://arxiv.org/abs/2402.09631.

Alessandro Stolfo, Vidhisha Balachandran, Safoora Yousefi, Eric Horvitz, and Besmira Nushi. Improving instruction-following in language models through activation steering. In *International Conference on Learning Representations*, 2025. URL https://mlanthology.org/iclr/2025/stolfo2025iclr-improving/.

Hao Sun, Huailiang Peng, Qiong Dai, Xu Bai, and Yanan Cao. Layernavigator: Finding promising intervention layers for efficient activation steering in large language models. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 101058–101080. Curran Associates, Inc., 2025. doi: 10.52202/085713-3383. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/9280bfb61f159c768dfdeb729a41ccd0-Paper-Conference.pdf.

Surge AI. The toxicity dataset. https://github.com/surge-ai/toxicity, 2021. GitHub repository. Accessed: February 1, 2026.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023. URL https://arxiv.org/abs/2302.13971.

Cynthia Van Hee, Els Lefever, and Véronique Hoste. SemEval-2018 task 3: Irony detection in English tweets. In Marianna Apidianaki, Saif M. Mohammad, Jonathan May, Ekaterina Shutova, Steven Bethard, and Marine Carpuat (eds.), *Proceedings of the 12th International Workshop on Semantic Evaluation*, pp. 39–50, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/S18-1005. URL https://aclanthology.org/S18-1005/.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. URL https://arxiv.org/abs/1706.03762.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned language models are zero-shot learners, 2022. URL https://arxiv.org/abs/2109.01652.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023. URL https://arxiv.org/abs/2201.11903.

Wenxuan Zhang, Yue Deng, Bing Liu, Sinno Jialin Pan, and Lidong Bing. Sentiment analysis in the era of large language models: A reality check. In Kevin Duh, Helena Gómez-Adorno, and Steven Bethard (eds.), *Findings of the Association for Computational Linguistics: NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, volume NAACL 2024 of *Findings of ACL*, pp. 3881–3906. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.FINDINGS-NAACL.246. URL https://doi.org/10.18653/v1/2024.findings-naacl.246.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. Large language models are human-level prompt engineers, 2023. URL https://arxiv.org/abs/2211.01910.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, J. Zico Kolter, and Dan Hendrycks. Representation engineering: A top-down approach to ai transparency, 2025. URL https://arxiv.org/abs/2310.01405.

A Appendix

A.1 Limitations

This work has a few limitations. First, contrastive pairs are constructed by matching input lengths rather than semantic relatedness, which may capture unintended variations; future work could use semantic similarity or LLM-generated pairs. Second, evaluation is restricted to English, leaving multilingual robustness and cross-cultural generalizability in models like Qwen unexplored.

A.2 Implementation Details

A.2.1 Model Initialization

We evaluate 20 decoder-based LLMs (124M–9.2B parameters) from four families: GPT-2, Gemma 2, Llama 3.2, and Qwen2.5, including base and instruction-tuned variants. For classification, we use masked mean pooling over token representations (Reimers & Gurevych, 2019):

$$\overline{\mathbf{h}} = \frac{\sum_t \mathbf{m}_t \mathbf{h}_t}{\sum_t \mathbf{m}_t}.$$

We additionally evaluate LoRA fine-tuning (Hu et al., 2021) on the query, key, value, and output projections. A two-layer MLP classifier maps the pooled representation to binary logits:

$$\text{Linear}(h \to h/2) \to \text{ReLU} \to \text{Linear}(h/2 \to 2).$$

A.2.2 Classification and Model Adaptation

The classifier is trained using cross-entropy loss:

$$\mathcal{L}_{\text{CE}} = -\frac{1}{N} \sum_{i=1}^{N} \log \frac{\exp(z_{i,y_i})}{\sum_{j=1}^{C} \exp(z_{i,j})},$$

where $z_{i,j}$ denotes the logit assigned to class j for sample i.

In the base setting, the LLM parameters remain frozen and only the classification head is trained, using a learning rate of 10^{-3}. We additionally consider LoRA adaptation (Hu et al., 2021), where the attention projection parameters are adapted while the remaining pretrained parameters remain frozen. For the LoRA configuration, we use a learning rate of 2×10^{-5} for the LoRA parameters and 10^{-3} for the classification head.

A.2.3 Models

Table 5 illustrates the decoder models used in this paper.

A.2.4 Hyperparameters

Table 6 shows the hyperparameters used for our experiments.

A.2.5 Statistical Test

We use the McNemar Test to check for statistical significance between the AdaSteer variants and the baseline. The McNemar Test is a statistical test for paired nominal data (McNemar, 1947). The test is applied to a 2×2 contingency table as illustrated in Table 7 (McNemar, 1947). For each test sample, we record whether the prediction was correct or incorrect for the base method (without steering) and the dynamic steering method.

McNemar's test focuses only on the number of samples that are discordant between the two methods, i.e., those incorrectly classified by one method but correctly classified by the other. Thus, the null hypothesis is

$$H_0 : P(b) = P(c),$$

where b denotes the number of samples classified correctly by the base model but incorrectly by the steered model, and c denotes the number of samples classified correctly by the steered model but incorrectly by the base model.

Under the null hypothesis, the number of samples in one discordant category follows a binomial distribution (Edwards, 1948),

Model Family	Model Name	Parameters	Type / Alignment
GPT-2	GPT-2	124M	Base
	GPT-2 Medium	355M	Base
	GPT-2 Large	774M	Base
	GPT-2 XL	1.5B	Base
Gemma 2	Gemma-2-2B	2.6B	Base
	Gemma-2-2B-IT	2.6B	Instruction-tuned
	Gemma-2-9B	9.2B	Base
	Gemma-2-9B-IT	9.2B	Instruction-tuned
Llama 3.2	Llama-3.2-1B	1B	Base
	Llama-3.2-1B-Instruct	1B	Instruction-tuned
	Llama-3.2-3B	3B	Base
	Llama-3.2-3B-Instruct	3B	Instruction-tuned
Qwen 2.5	Qwen2.5-0.5B	0.5B	Base
	Qwen2.5-0.5B-Instruct	0.5B	Instruction-tuned
	Qwen2.5-1.5B	1.5B	Base
	Qwen2.5-1.5B-Instruct	1.5B	Instruction-tuned
	Qwen2.5-3B	3B	Base
	Qwen2.5-3B-Instruct	3B	Instruction-tuned
	Qwen2.5-7B	7B	Base
	Qwen2.5-7B-Instruct	7B	Instruction-tuned

Table 5: Language Models Used to Evaluate Dynamic Steering for Classification

Hyperparameter	Value
Quantisation	8-bit integer quantisation
Learning rate without LoRA	1×10^{-3}
LoRA rank r	8
LoRA scaling α	16
LoRA dropout	0.1
LoRA targets	q, k, v, o projections
LoRA targets Gpt-2	c_attn, c_proj
Max sequence length	128 tokens
Batch size	8
LoRA learning rate	2×10^{-5}
Classifier learning rate	1×10^{-3}
Optimiser	Adam8bit
Steering layer l	$L - 2$
Pooling	Masked mean pool
CAA method	Paired (length-matched)
Sharpening temperature τ	5.0
Alpha sweep Ω	$\mathrm{linspace}(-1, 1, 8)$
Alpha predictor hidden	[64, 64]
Alpha predictor epochs	1000
Alpha predictor learning rate	1×10^{-3} to 0 (cosine)
Inference α bound	$\tanh(\hat{\alpha})$
Significance threshold	0.05

Table 6: Summary of all hyperparameters and design choices.

$$X \sim \text{Binomial}(b + c, 0.5).$$

	Steering Correct	Steering Incorrect
Base Correct	a	b
Base Incorrect	c	d

Table 7: Paired 2×2 contingency table for the McNemar Test. The table illustrates the number of correct and incorrect predictions from the base (no steering) model and the dynamic steering model. 'a' refers to the number of samples which have been classified correctly by both methods, 'b' refers to the number of samples classified correctly by the base model but incorrectly by the dynamic steering method, 'c' refers to the number of samples classified correctly by the steered model but incorrectly by the base model, and 'd' refers to the number of samples which have been incorrectly classified by both methods. (McNemar, 1947)

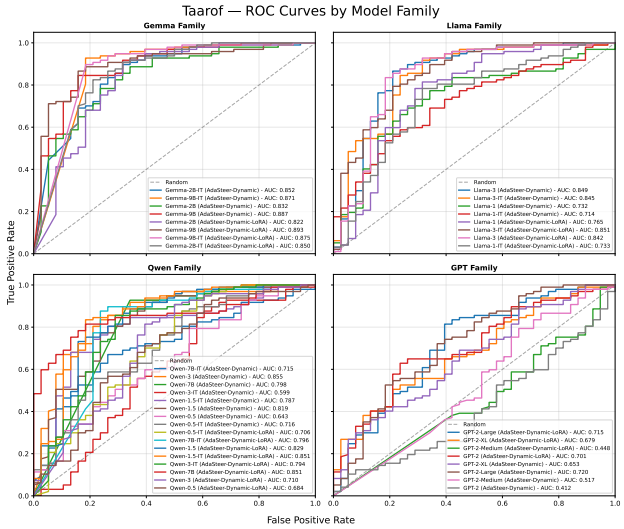

Figure 4: Receiver Operating Characteristic (ROC) curves and Area Under the Curve (AUC) scores for the Taarof dataset grouped by model family. The subplots display performance across various configurations using AdaSteer-Dynamic and AdaSteer-Dynamic-LoRA methods for the Gemma, Llama, Qwen, and Gpt-2model families. Each curve illustrates the true positive rate against the false positive rate.

The exact two-sided p-value is computed based on the binomial formulation of McNemar's test, evaluating how extreme the observed imbalance between b and c is under the null hypothesis (Edwards, 1948).

This formulation avoids the chi-square approximation and is particularly appropriate when the number of discordant pairs is small (Edwards, 1948).

A.3 Results

This section illustrates additional results from our experiments for all the datasets.

A.3.1 Taarof

Figure 5 illustrates the activation patching results for the Taarof dataset. Figure 4 illustrates the ROC curves for AdaSteer for the Taarof dataset for the model families.

A.3.2 Sentiment

Table 8 and Table 9 illustrate the results of the AdaSteer variants on the Sentiment dataset. Figure 7 illustrates the results of AdaSteer on the Sentiment dataset and Figure 8 shows the results of acti-

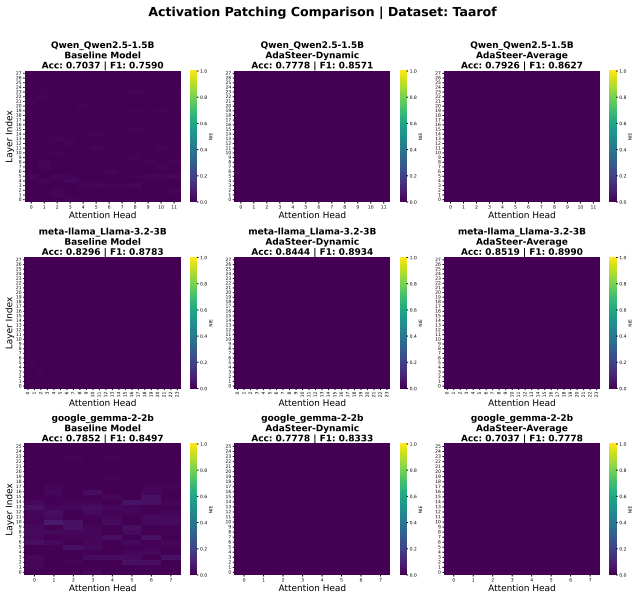

Figure 5: Activation patching analysis comparing the baseline, AdaSteer-Dynamic, and AdaSteer-Average methods on the Taarof dataset across three models Qwen-1.5, Llama-3, and Gemma-2.

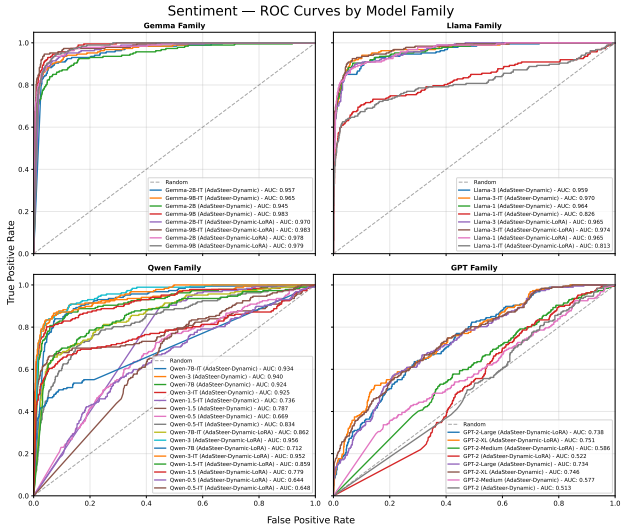

Figure 6: Receiver Operating Characteristic (ROC) curves and Area Under the Curve (AUC) scores for the Sentiment dataset grouped by model family. The subplots display performance across various configurations using AdaSteer-Dynamic and AdaSteer-Dynamic-LoRA methods for the Gemma, Llama, Qwen, and Gpt-2model families. Each curve illustrates the true positive rate against the false positive rate.

vation patching on the Sentiment dataset. Figure 6 illustrates the ROC curves for AdaSteer for the Sentiment dataset for the model families.

A.3.3 IRONY

Table 3 and Table 4 illustrate the results of the AdaSteer variants on the Irony dataset. Figure 10 illustrates the results of AdaSteer on the Irony dataset and Figure 11 shows the results of activation

Model	Baseline	AdaSteer-Dynamic-LoRA	AdaSteer-Average-LoRA
Gpt-2	84.3 / 56.1	35.5 / 34.0	53.8 / 44.8
Gpt-2-Medium	31.3 / 31.0	**43.3 / 39.5***	**40.8 / 38.1***
Gpt-2-Large	86.8 / 60.5	80.1 / **60.8**	84.2 / **63.6**
Gpt-2-XL	87.3 / 58.9	82.3 / **65.1**	74.9 / **60.6***
Gemma-2B	95.8 / 90.1	95.8 / 89.2	95.2 / 87.7
Gemma-2B-IT	94.8 / 88.1	**95.4 / 89.6**	**95.5 / 89.5***
Gemma-9B	95.2 / 89.0	**95.6 / 90.0**	95.0 / 87.5
Gemma-9B-IT	95.5 / 89.3	**95.6** / 88.8	**96.0 / 89.9**
Llama-1	92.1 / 75.0	86.7 / **77.5**	86.8 / **77.6**
Llama-1-IT	91.5 / 72.2	**92.1 / 75.8**	**93.5 / 83.0***
Llama-3	95.1 / 87.3	92.1 / 84.6	92.0 / 84.3
Llama-3-IT	95.3 / 88.9	**95.4 / 89.4**	95.3 / **89.0**
Qwen-0.5	89.8 / 74.1	53.1 / 46.3	86.8 / 50.2
Qwen-0.5-IT	83.9 / 71.2	44.1 / 41.0	59.0 / 49.2
Qwen-1.5	92.1 / 81.3	91.9 / 78.8	91.5 / 79.2
Qwen-1.5-IT	90.4 / 79.8	88.7 / 76.9	60.3 / 54.2
Qwen-3	94.2 / 87.1	**94.6 / 87.7**	**95.0 / 88.2**
Qwen-3-IT	94.9 / 88.1	**95.0** / 87.9	**95.0 / 88.3**
Qwen-7B	93.8 / 83.8	90.6 / 71.4	90.5 / 71.4
Qwen-7B-IT	93.4 / 82.7	91.1 / 76.8	92.3 / 79.0

Table 8: Performance comparison on the **Sentiment** test set using steering vectors computed from 100% of the training data under the **LoRA** configuration. Results are reported as Accuracy / Macro F1 (%). The baseline uses no steering, while AdaSteer-Dynamic-LoRA and AdaSteer-Average-LoRA apply document-specific adaptive and average steering, respectively. **Bold** values indicate improvements over the corresponding baseline metric. * indicates a statistically significant difference from the baseline according to the exact two-sided McNemar test ($p < 0.05$). Models are grouped by model family and ordered by model size.

Model	Baseline	AdaSteer-Dynamic	AdaSteer-Average
Gpt-2	74.0 / 55.9	24.7 / 24.7	24.8 / 24.8
Gpt-2-Medium	69.7 / 53.7	62.7 / 49.7	62.1 / 49.8
Gpt-2-Large	76.7 / 61.0	76.5 / 61.0	**77.9 / 62.5***
Gpt-2-XL	43.6 / 41.1	**63.8 / 54.0***	**67.4 / 55.4***
Gemma-2B	93.1 / 84.8	**94.4 / 87.0***	**95.5 / 88.8***
Gemma-2B-IT	94.6 / 86.9	**95.0 / 89.0**	94.5 / **87.2**
Gemma-9B	95.9 / 90.6	**96.3 / 91.6**	**96.6 / 92.3***
Gemma-9B-IT	94.6 / 87.4	**95.4 / 88.9***	**95.4 / 88.9***
Llama-1	94.8 / 87.7	94.2 / 87.7	94.6 / **88.4**
Llama-1-IT	94.5 / 85.6	91.8 / 74.5	91.7 / 73.5
Llama-3	93.2 / 80.6	**95.2 / 88.9***	**93.8 / 86.3**
Llama-3-IT	93.1 / 79.4	**95.2 / 88.0***	**95.0 / 87.0**
Qwen-0.5	84.0 / 72.4	64.2 / 53.4	83.7 / 69.6
Qwen-0.5-IT	92.2 / 79.9	86.8 / 72.7	71.7 / 59.8
Qwen-1.5	93.9 / 85.8	92.7 / 80.4	92.3 / 78.9
Qwen-1.5-IT	91.4 / 83.2	43.9 / 41.7	**94.0 / 85.3***
Qwen-3	93.3 / 81.3	**94.8 / 88.0***	**94.8 / 87.1**
Qwen-3-IT	94.2 / 86.9	**94.4** / 85.5	93.8 / 84.7*
Qwen-7B	94.5 / 87.5	94.0 / 85.8	93.2 / 82.8
Qwen-7B-IT	90.5 / 82.5	**91.7 / 83.0**	**91.7 / 83.2**

Table 9: Performance comparison on the **Sentiment** test set using steering vectors computed from 100% of the training data under the **No-LoRA** configuration. Results are reported as Accuracy / Macro F1 (%). The baseline uses no steering, while AdaSteer-Dynamic and AdaSteer-Average apply document-specific adaptive and average steering, respectively. **Bold** values indicate improvements over the corresponding baseline metric. * indicates a statistically significant difference from the baseline according to the exact two-sided McNemar test ($p < 0.05$). Models are grouped by model family and ordered by model size.

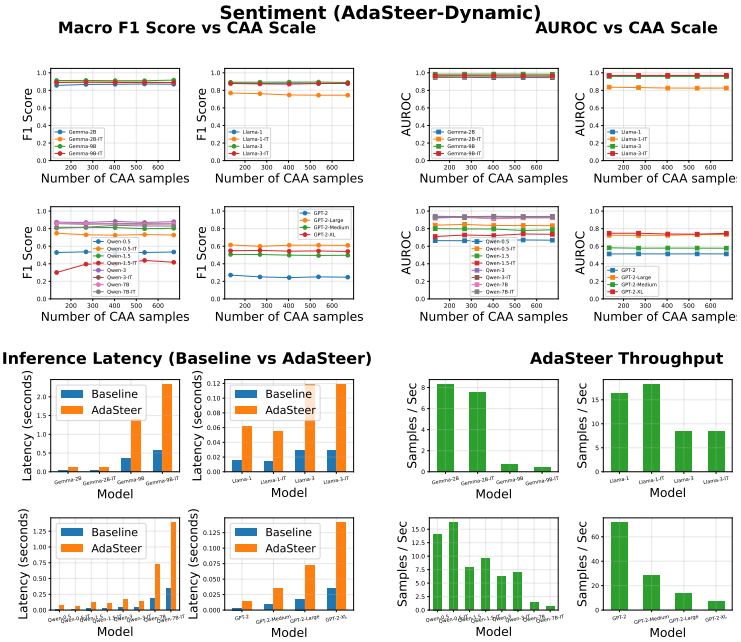

Figure 7: Results for the Sentiment (No-LoRA) dataset. The top row illustrates F1 scores and AUROC metrics plotted across varying Contrastive Activation Addition (CAA) scales for the Gemma, Llama, Qwen, and Gpt-2model families. The bottom-left section illustrates latency in seconds between unsteered baselines and AdaSteer interventions across architectures, while the bottom-right displays system throughput measured in samples per second for each evaluated model configuration.

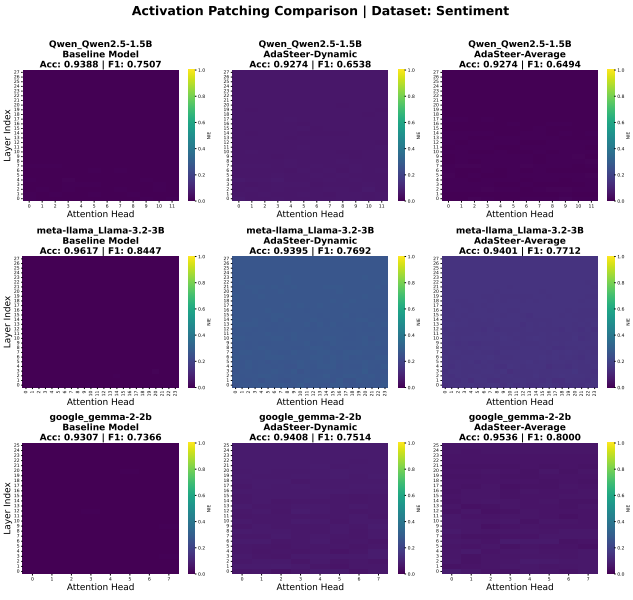

Figure 8: Activation patching analysis comparing the baseline, AdaSteer-Dynamic, and AdaSteer-Average methods on the Sentiment dataset across three models Qwen-1.5, Llama-3, and Gemma-2.

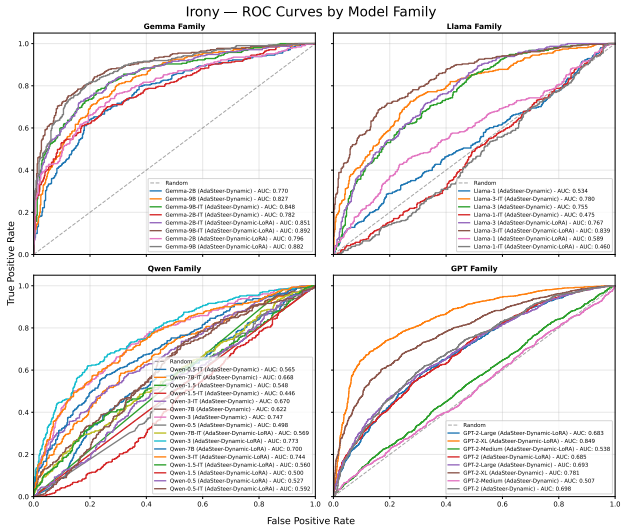

Figure 9: Receiver Operating Characteristic (ROC) curves and Area Under the Curve (AUC) scores for the Irony dataset grouped by model family. The subplots display performance across various configurations using AdaSteer-Dynamic and AdaSteer-Dynamic-LoRA methods for the Gemma, Llama, Qwen, and Gpt-2model families. Each curve illustrates the true positive rate against the false positive rate.

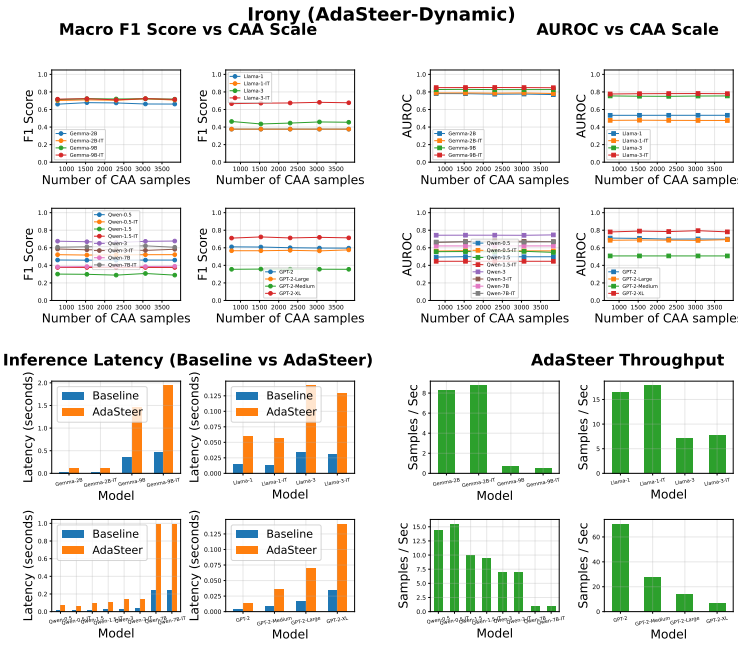

Figure 10: Results for the Irony (No-LoRA) dataset. The top row illustrates F1 scores and AUROC metrics plotted across varying Contrastive Activation Addition (CAA) scales for the Gemma, Llama, Qwen, and Gpt-2model families. The bottom-left section illustrates latency in seconds between unsteered baselines and AdaSteer interventions across architectures, while the bottom-right displays system throughput measured in samples per second for each evaluated model configuration.

patching on the Irony dataset. Figure 9 illustrates the ROC curves for AdaSteer for the Irony dataset for the model families.

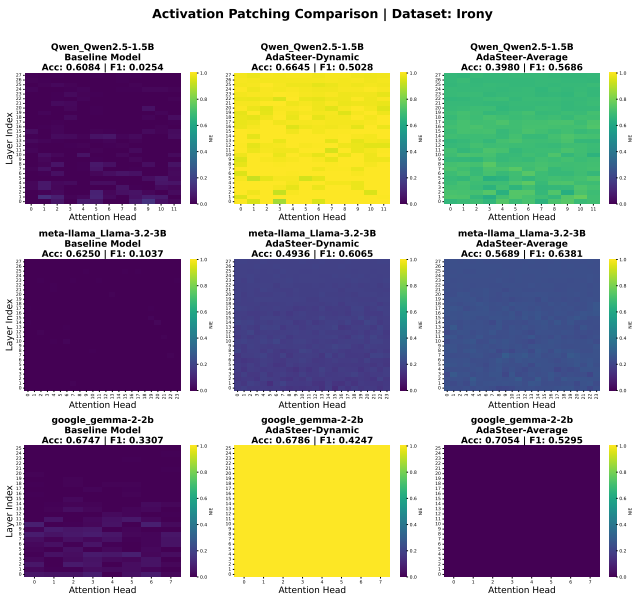

Figure 11: Activation patching analysis comparing the baseline, AdaSteer-Dynamic, and AdaSteer-Average methods on the Irony dataset across three models Qwen-1.5, Llama-3, and Gemma-2.

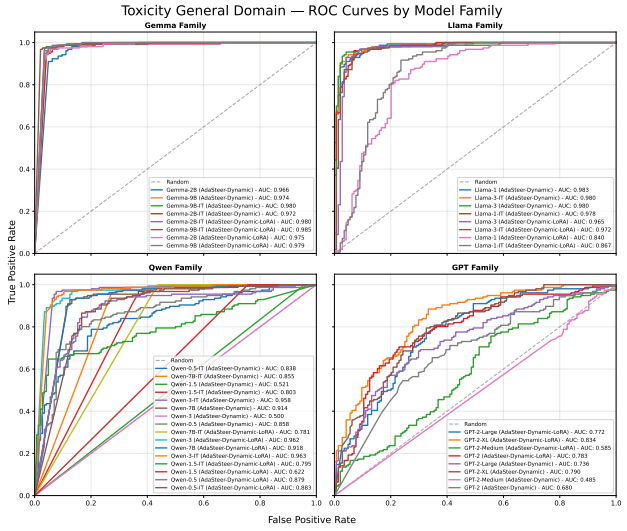

Figure 12: Receiver Operating Characteristic (ROC) curves and Area Under the Curve (AUC) scores for the Toxicity General Domain dataset grouped by model family. The subplots display performance across various configurations using AdaSteer-Dynamic and AdaSteer-Dynamic-LoRA methods for the Gemma, Llama, Qwen, and Gpt-2model families. Each curve illustrates the true positive rate against the false positive rate.

A.3.4 Toxicity General Domain

Table 10 and Table 11 illustrate the results of the AdaSteer variants on the Toxicity General Domain dataset. Figure 13 illustrates the results of AdaSteer on the Toxicity General Domain dataset and Figure 14 shows the results of activation patching on the Toxicity General Domain dataset. Figure 12 illustrates the ROC curves for AdaSteer for the General Domain Toxicity dataset for the model families.

Model	Baseline	AdaSteer-Dynamic-LoRA	AdaSteer-Average-LoRA
Gpt-2	68.0 / 67.3	**71.7 / 71.6**	57.7 / 46.9
Gpt-2-Medium	54.3 / 47.7	53.3 / 45.1	49.7 / 39.5
Gpt-2-Large	68.0 / 67.9	**70.3 / 70.3**	**69.3 / 68.6**
Gpt-2-XL	69.3 / 68.5	**71.0 / 70.7**	63.7 / 60.8
Gemma-2B	87.3 / 87.0	**95.3 / 95.3***	**94.0 / 94.0**
Gemma-2B-IT	89.7 / 89.5	**96.3 / 96.3***	**96.3 / 96.3***
Gemma-9B	95.0 / 95.0	**95.7 / 95.7**	**95.7 / 95.7**
Gemma-9B-IT	93.0 / 92.9	**97.3 / 97.3***	**97.3 / 97.3***
Llama-1	92.0 / 91.9	57.7 / 46.1	57.3 / 45.5
Llama-1-IT	94.0 / 94.0	62.7 / 55.0	62.3 / 54.4
Llama-3	90.3 / 90.2	**91.7 / 91.6**	**92.7 / 92.6**
Llama-3-IT	90.7 / 90.5	**92.3 / 92.3**	**92.7 / 92.6**
Qwen-0.5	87.3 / 87.3	83.0 / 83.0	83.0 / 83.0
Qwen-0.5-IT	86.3 / 86.3	82.7 / 82.5	83.3 / 83.2
Qwen-1.5	87.3 / 87.1	59.0 / 48.6	**90.0 / 89.9**
Qwen-1.5-IT	89.7 / 89.6	74.3 / 73.5	71.3 / 69.9
Qwen-3	91.3 / 91.3	**92.0 / 92.0**	91.3 / 91.3*
Qwen-3-IT	90.3 / 90.2	**93.0 / 93.0***	**92.7 / 92.6***
Qwen-7B	88.3 / 88.1	**88.7 / 88.6**	88.0 / 87.9
Qwen-7B-IT	90.7 / 90.5	75.3 / 73.1	73.7 / 71.1

Table 10: Performance comparison on the **Toxicity General Domain** test set using steering vectors computed from 100% of the training data under the **LoRA** configuration. Results are reported as Accuracy / F1 (%). The baseline uses no steering, while AdaSteer-Dynamic-LoRA and AdaSteer-Average-LoRA apply document-specific adaptive and average steering, respectively. **Bold** values indicate improvements over the corresponding baseline metric. * indicates a statistically significant difference from the baseline according to the exact two-sided McNemar test ($p < 0.05$). Models are grouped by model family and ordered by model size.

Model	Baseline	AdaSteer-Dynamic	AdaSteer-Average
Gpt-2	69.0 / 67.4	54.0 / 41.6	55.3 / 43.8
Gpt-2-Medium	57.3 / 57.3	52.3 / 40.2	52.7 / 40.4
Gpt-2-Large	65.0 / 62.8	**67.3 / 66.8**	**67.0 / 66.3**
Gpt-2-XL	66.0 / 60.5	**72.3 / 72.0***	**70.3 / 69.6**
Gemma-2B	88.0 / 87.8	**92.3 / 92.3***	**92.3 / 92.3***
Gemma-2B-IT	91.3 / 91.2	**94.7 / 94.7***	**95.0 / 95.0***
Gemma-9B	92.3 / 92.3	**95.3 / 95.3***	**94.0 / 94.0**
Gemma-9B-IT	84.3 / 83.7	**96.7 / 96.7***	**96.3 / 96.3***
Llama-1	93.3 / 93.3	**93.7 / 93.6**	93.0 / 93.0
Llama-1-IT	93.7 / 93.6	88.7 / 88.5	88.3 / 88.1
Llama-3	93.7 / 93.6	**95.0 / 95.0**	**95.3 / 95.3**
Llama-3-IT	91.7 / 91.6	**92.3 / 92.3**	**93.3 / 93.3**
Qwen-0.5	88.3 / 88.3	78.3 / 78.1	77.0 / 76.8
Qwen-0.5-IT	87.7 / 87.5	75.0 / 74.8	81.7 / 81.7
Qwen-1.5	89.0 / 88.8	52.0 / 34.2	77.0 / 76.7
Qwen-1.5-IT	91.7 / 91.6	71.0 / 67.8	70.3 / 66.9
Qwen-3	90.0 / 89.8	52.0 / 34.2	52.0 / 34.2
Qwen-3-IT	86.7 / 86.3	**91.3 / 91.2***	**93.3 / 93.3**
Qwen-7B	90.3 / 90.2	89.0 / 89.0	84.3 / 84.3
Qwen-7B-IT	91.7 / 91.6	83.0 / 82.4	79.3 / 78.2

Table 11: Performance comparison on the **Toxicity General Domain** test set using steering vectors computed from 100% of the training data under the **No-LoRA** configuration. Results are reported as Accuracy / F1 (%). The baseline uses no steering, while AdaSteer-Dynamic and AdaSteer-Average apply document-specific adaptive and average steering, respectively. **Bold** values indicate improvements over the corresponding baseline metric. * indicates a statistically significant difference from the baseline according to the exact two-sided McNemar test ($p < 0.05$). Models are grouped by model family and ordered by model size.

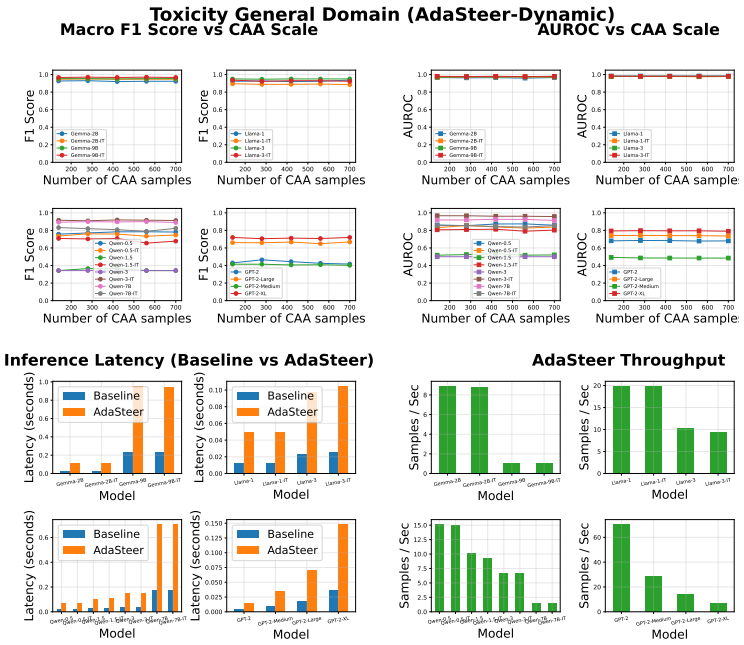

Figure 13: Results for the Toxicity General Domain (No-LoRA) dataset. The top row illustrates F1 scores and AUROC metrics plotted across varying Contrastive Activation Addition (CAA) scales for the Gemma, Llama, Qwen, and Gpt-2model families. The bottom-left section illustrates latency in seconds between unsteered baselines and AdaSteer interventions across architectures, while the bottom-right displays system throughput measured in samples per second for each evaluated model configuration.

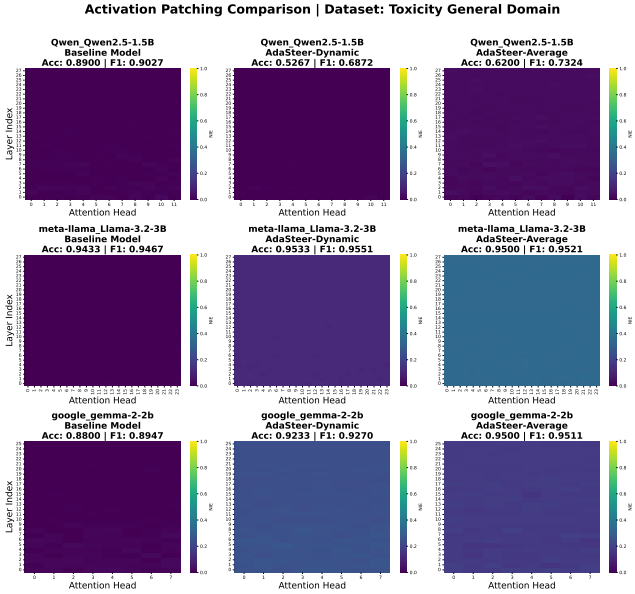

Figure 14: Activation patching analysis comparing the baseline, AdaSteer-Dynamic, and AdaSteer-Average methods on the Toxicity General Domain dataset across three models Qwen-1.5, Llama-3, and Gemma-2.

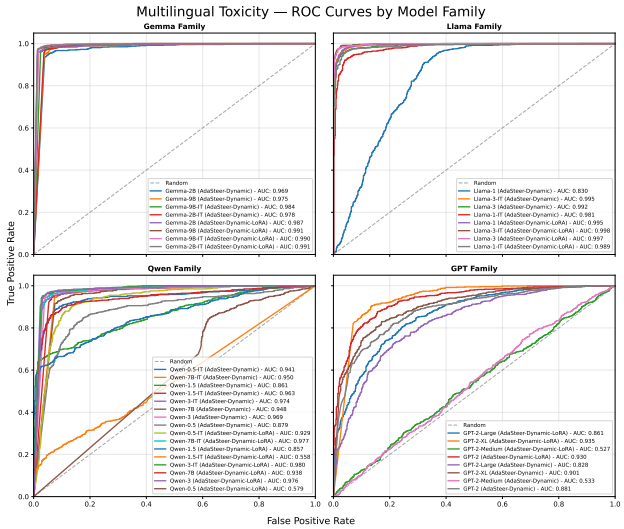

Figure 15: Receiver Operating Characteristic (ROC) curves and Area Under the Curve (AUC) scores for the Multilingual Toxicity dataset grouped by model family. The subplots display performance across various configurations using AdaSteer-Dynamic and AdaSteer-Dynamic-LoRA methods for the Gemma, Llama, Qwen, and Gpt-2model families. Each curve illustrates the true positive rate against the false positive rate.

A.3.5 Multilingual Toxicity

Table 12 and Table 13 illustrate the results of the AdaSteer variants on the Multilingual Toxicity dataset. Figure 16 illustrates the results of AdaSteer on the Multilingual Toxicity dataset and Figure 17 shows the results of activation patching on the Multilingual Toxicty dataset. Figure 15 illustrates the ROC curves for AdaSteer for the Multilingual Toxicity dataset for the model families.

A.3.6 Politeness

Table 14 and Table 15 illustrate the results of the AdaSteer variants on the Politeness dataset. Figure 19 illustrates the results of AdaSteer on the Politeness dataset and Figure 20 shows the results of activation patching on the Politeness dataset. Figure 18 illustrates the ROC curves for AdaSteer for the Politeness dataset for the model families.

Model	Baseline	AdaSteer-Dynamic-LoRA	AdaSteer-Average-LoRA
Gpt-2	82.5 / 82.2	**83.4 / 83.1**	**84.1 / 84.1**
Gpt-2-Medium	56.3 / 44.4	52.5 / 42.2	52.7 / 52.7
Gpt-2-Large	72.5 / 71.1	**76.3 / 76.0***	**75.1 / 74.3***
Gpt-2-XL	86.5 / 86.5	**87.3 / 87.3***	**88.9 / 88.9**
Gemma-2B	96.7 / 96.7	**97.0 / 97.0**	**97.3 / 97.3***
Gemma-2B-IT	97.7 / 97.7	97.5 / 97.5	**98.1 / 98.1***
Gemma-9B	97.5 / 97.5	97.5 / 97.5	97.3 / 97.3
Gemma-9B-IT	97.7 / 97.7	97.7 / 97.7	**97.8 / 97.8**
Llama-1	96.6 / 96.6	96.5 / 96.5	96.6 / 96.6
Llama-1-IT	96.3 / 96.3	90.6 / 90.5	90.7 / 90.5
Llama-3	97.0 / 97.0	**97.5 / 97.5**	**97.6 / 97.6**
Llama-3-IT	95.7 / 95.7	**97.9 / 97.9***	**97.1 / 97.1***
Qwen-0.5	91.5 / 91.4	59.4 / 56.6	59.9 / 57.1
Qwen-0.5-IT	91.1 / 91.1	88.2 / 88.2	89.2 / 89.2
Qwen-1.5	94.9 / 94.9	73.9 / 71.6	73.7 / 71.4
Qwen-1.5-IT	95.0 / 95.0	51.4 / 34.1	51.4 / 34.1
Qwen-3	95.2 / 95.2	**95.6 / 95.6**	**95.6 / 95.6**
Qwen-3-IT	94.9 / 94.9	**95.7 / 95.7***	**95.6 / 95.6**
Qwen-7B	92.1 / 92.1	89.5 / 89.4	**94.1 / 94.1**
Qwen-7B-IT	90.7 / 90.7	**94.7 / 94.7***	**94.4 / 94.4***

Table 12: Performance comparison on the **Multilingual Toxicity** test set using steering vectors computed from 100% of the training data under the **LoRA** configuration. Results are reported as Accuracy / F1 (%). The baseline uses no steering, while AdaSteer-Dynamic-LoRA and AdaSteer-Average-LoRA apply document-specific adaptive and average steering, respectively. **Bold** values indicate improvements over the corresponding baseline metric. * indicates a statistically significant difference from the baseline according to the exact two-sided McNemar test ($p < 0.05$). Models are grouped by model family and ordered by model size.

Model	Baseline	AdaSteer-Dynamic	AdaSteer-Average
Gpt-2	79.1 / 79.0	72.3 / 71.4	71.8 / 70.7
Gpt-2-Medium	62.7 / 57.7	51.2 / 36.0	51.3 / 36.1
Gpt-2-Large	70.2 / 69.1	**73.9 / 73.8***	**74.1 / 73.9***
Gpt-2-XL	71.2 / 68.3	**82.5 / 82.3***	**82.1 / 81.9***
Gemma-2B	94.3 / 94.3	**94.5 / 94.5**	**95.1 / 95.1***
Gemma-2B-IT	95.9 / 95.9	**96.4 / 96.4**	95.7 / 95.7
Gemma-9B	95.5 / 95.5	95.5 / 95.5	95.4 / 95.4
Gemma-9B-IT	95.7 / 95.7	**96.7 / 96.7***	**96.8 / 96.8***
Llama-1	92.2 / 92.1	49.3 / 34.2	89.9 / 89.9
Llama-1-IT	89.1 / 89.0	80.9 / 80.4	**94.2 / 94.2***
Llama-3	93.6 / 93.6	**95.7 / 95.7***	**95.7 / 95.7***
Llama-3-IT	95.9 / 95.9	**96.4 / 96.4**	**96.5 / 96.5**
Qwen-0.5	90.7 / 90.7	81.7 / 81.5	67.5 / 63.3
Qwen-0.5-IT	90.0 / 90.0	**90.1 / 90.1**	66.2 / 65.8
Qwen-1.5	94.5 / 94.5	74.3 / 72.3	73.9 / 71.7
Qwen-1.5-IT	95.1 / 95.1	92.5 / 92.5	92.5 / 92.5
Qwen-3	94.7 / 94.7	**95.1 / 95.1**	94.7 / 94.7
Qwen-3-IT	94.4 / 94.4	94.2 / 94.2	**95.0 / 95.0**
Qwen-7B	93.3 / 93.3	91.6 / 91.6	91.7 / 91.7
Qwen-7B-IT	92.3 / 92.2	90.8 / 90.8	90.4 / 90.4

Table 13: Performance comparison on the **Multilingual Toxicity** test set using steering vectors computed from 100% of the training data under the **No-LoRA** configuration. Results are reported as Accuracy / F1 (%). The baseline uses no steering, while AdaSteer-Dynamic and AdaSteer-Average apply document-specific adaptive and average steering, respectively. **Bold** values indicate improvements over the corresponding baseline metric. * indicates a statistically significant difference from the baseline according to the exact two-sided McNemar test ($p < 0.05$). Models are grouped by model family and ordered by model size.

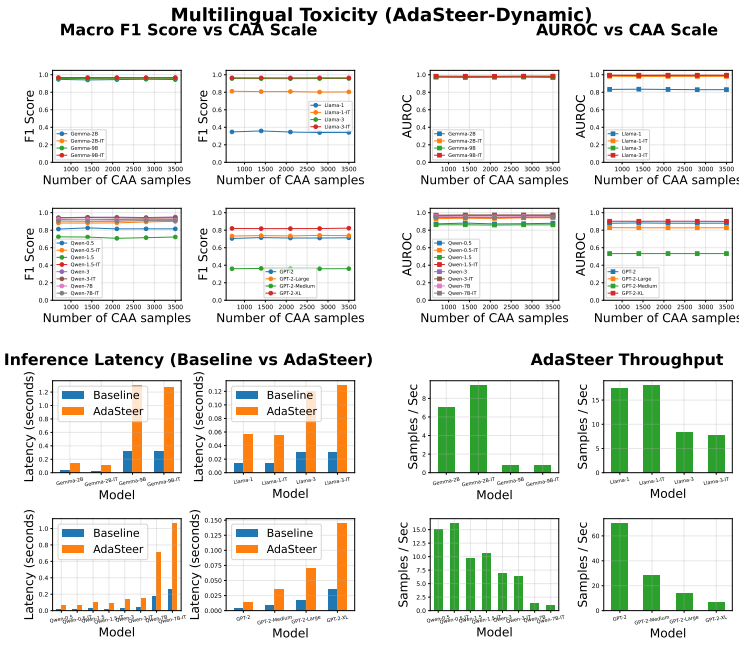

Figure 16: Results for the Multilingual Toxicity (No-LoRA) dataset. The top row illustrates F1 scores and AUROC metrics plotted across varying Contrastive Activation Addition (CAA) scales for the Gemma, Llama, Qwen, and Gpt-2model families. The bottom-left section illustrates latency in seconds between unsteered baselines and AdaSteer interventions across architectures, while the bottom-right displays system throughput measured in samples per second for each evaluated model configuration.

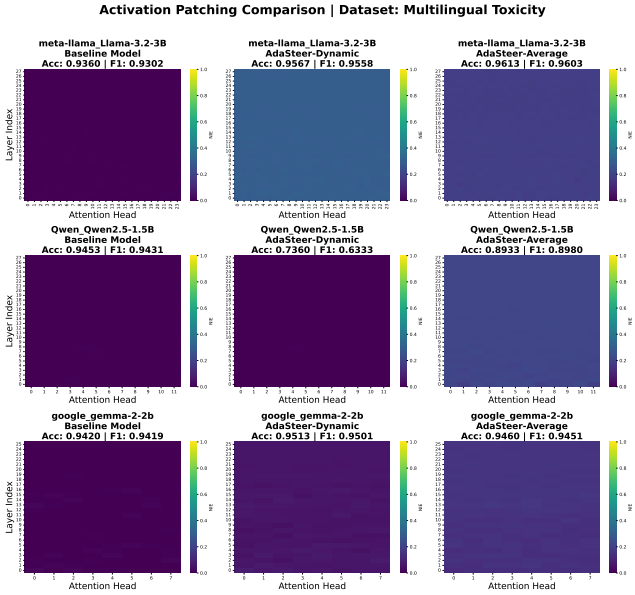

Figure 17: Activation patching analysis comparing the baseline, AdaSteer-Dynamic, and AdaSteer-Average methods on the Multilingual Toxicity dataset across three models Qwen-1.5, Llama-3, and Gemma-2.

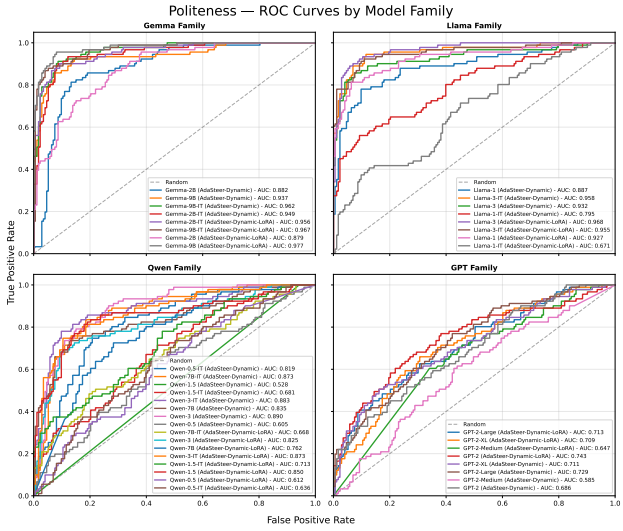

Figure 18: Receiver Operating Characteristic (ROC) curves and Area Under the Curve (AUC) scores for the politeness dataset grouped by model family. The subplots display performance across various configurations using AdaSteer-Dynamic and AdaSteer-Dynamic-LoRA methods for the Gemma, Llama, Qwen, and Gpt-2model families. Each curve illustrates the true positive rate against the false positive rate.

Model	Baseline	AdaSteer-Dynamic-LoRA	AdaSteer-Average-LoRA
Gpt-2	72.0 / 57.5	70.5 / 53.3	69.3 / 51.1
Gpt-2-Medium	39.0 / 33.7	**50.8 / 50.7***	**51.9 / 51.9***
Gpt-2-Large	65.5 / 39.6	65.5 / 39.6	65.5 / 39.6
Gpt-2-XL	65.5 / 39.6	65.5 / 39.6	65.5 / 39.6
Gemma-2B	89.8 / 87.9	73.5 / 61.3	72.4 / 57.8
Gemma-2B-IT	90.5 / 89.4	90.5 / 89.2	90.1 / 88.6
Gemma-9B	92.0 / 91.0	**92.8 / 91.9**	**94.3 / 93.6**
Gemma-9B-IT	92.8 / 92.0	91.7 / 90.5	92.0 / 90.9
Llama-1	90.5 / 89.9	84.1 / 79.8	83.3 / 78.6
Llama-1-IT	89.8 / 89.0	65.5 / 39.6	65.5 / 39.6
Llama-3	88.6 / 88.0	87.9 / 85.6	**93.6 / 92.6**
Llama-3-IT	88.3 / 87.5	**90.5 / 88.8**	85.2 / 81.5
Qwen-0.5	72.7 / 72.3	65.1 / 39.5	65.1 / 39.5
Qwen-0.5-IT	71.2 / 70.8	65.5 / 53.3	65.1 / 52.6
Qwen-1.5	82.6 / 81.9	74.6 / 74.1	74.2 / 73.8
Qwen-1.5-IT	89.8 / 88.7	66.7 / 43.0	65.5 / 39.6
Qwen-3	83.0 / 82.2	82.6 / 79.9	81.1 / 80.4
Qwen-3-IT	86.4 / 85.6	79.9 / 74.4	84.1 / 81.6
Qwen-7B	76.1 / 67.0	71.2 / 69.6	**77.3 / 72.5**
Qwen-7B-IT	81.4 / 75.8	69.3 / 51.8	70.8 / 58.2

Table 14: Performance comparison on the **Politeness** test set using steering vectors computed from 100% of the training data under the **LoRA** configuration. Results are reported as Accuracy / F1 (%). The baseline uses no steering, while AdaSteer-Dynamic-LoRA and AdaSteer-Average-LoRA apply document-specific adaptive and average steering, respectively. **Bold** values indicate improvements over the corresponding baseline metric. * indicates a statistically significant difference from the baseline according to the exact two-sided McNemar test ($p < 0.05$). Models are grouped by model family and ordered by model size.

Model	Baseline	AdaSteer-Dynamic	AdaSteer-Average
Gpt-2	75.0 / 66.7	67.8 / 47.2	68.9 / 49.4
Gpt-2-Medium	34.8 / 26.3	**65.1 / 39.5***	**65.1 / 39.5***
Gpt-2-Large	71.6 / 57.2	**72.0 / 58.1**	**72.0 / 58.1**
Gpt-2-XL	65.5 / 39.6	**71.6 / 59.8***	**72.0 / 60.1***
Gemma-2B	90.5 / 89.6	84.9 / 83.5	65.5 / 39.6
Gemma-2B-IT	78.8 / 78.4	**87.5 / 85.5***	**86.4 / 83.9**
Gemma-9B	88.3 / 87.5	86.4 / 83.5	84.9 / 81.4
Gemma-9B-IT	89.8 / 88.9	89.4 / 87.6	**90.1 / 88.6**
Llama-1	82.6 / 82.1	65.5 / 39.6	65.5 / 39.6
Llama-1-IT	70.1 / 70.1	65.5 / 39.6	65.5 / 39.6
Llama-3	86.0 / 85.4	69.7 / 51.4	68.9 / 49.4
Llama-3-IT	84.5 / 83.9	71.2 / 55.1	76.5 / 66.6
Qwen-0.5	71.2 / 70.4	65.5 / 39.6	65.5 / 39.6
Qwen-0.5-IT	61.0 / 60.9	**75.8 / 71.9***	**77.6 / 74.5***
Qwen-1.5	75.8 / 75.6	34.5 / 25.6	72.7 / 59.2
Qwen-1.5-IT	79.9 / 79.6	65.5 / 39.6	35.2 / 27.3
Qwen-3	86.4 / 85.5	78.8 / 77.7	80.7 / 79.7
Qwen-3-IT	86.4 / 85.1	83.7 / 82.4	83.0 / 81.7
Qwen-7B	85.6 / 84.0	78.8 / 72.3	78.0 / 70.7
Qwen-7B-IT	84.1 / 83.3	82.2 / 80.3	82.6 / 80.9

Table 15: Performance comparison on the **Politeness** test set using steering vectors computed from 100% of the training data under the **No-LoRA** configuration. Results are reported as Accuracy / F1 (%). The baseline uses no steering, while AdaSteer-Dynamic and AdaSteer-Average apply document-specific adaptive and average steering, respectively. **Bold** values indicate improvements over the corresponding baseline metric. * indicates a statistically significant difference from the baseline according to the exact two-sided McNemar test ($p < 0.05$). Models are grouped by model family and ordered by model size.

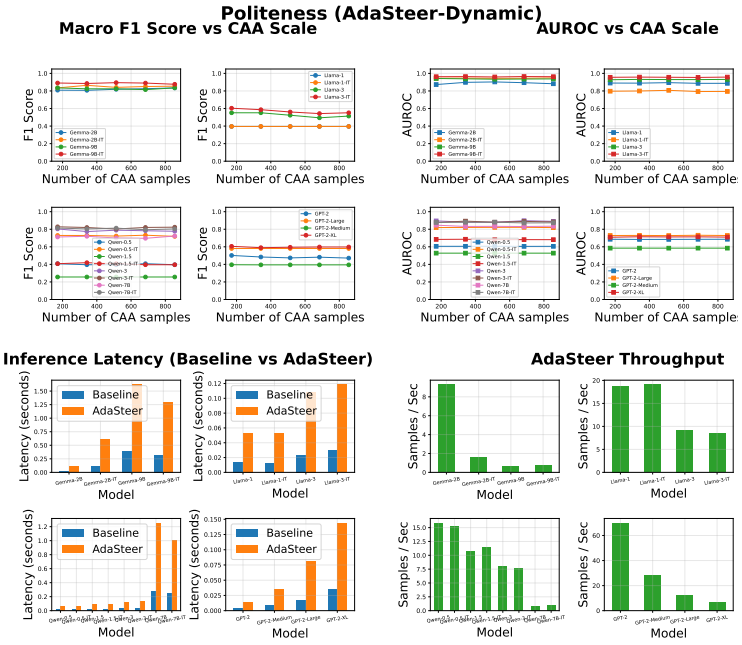

Figure 19: Results for the Politeness (No-LoRA) dataset. The top row illustrates F1 scores and AUROC metrics plotted across varying Contrastive Activation Addition (CAA) scales for the Gemma, Llama, Qwen, and Gpt-2model families. The bottom-left section illustrates latency in seconds between unsteered baselines and AdaSteer interventions across architectures, while the bottom-right displays system throughput measured in samples per second for each evaluated model configuration.

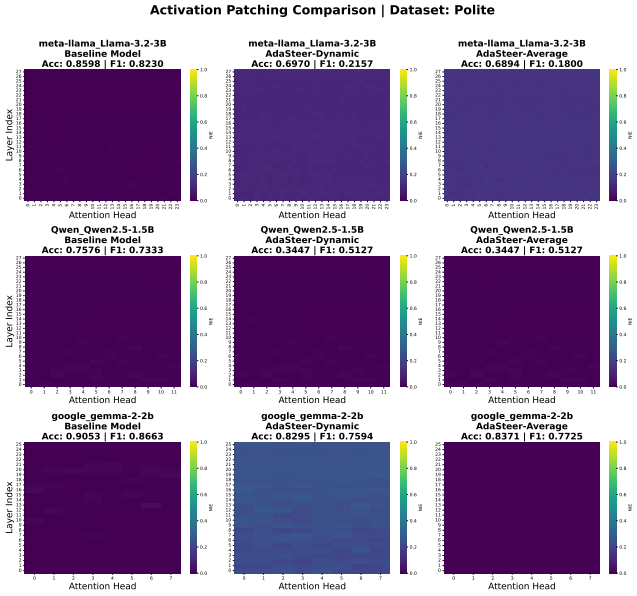

Figure 20: Activation patching analysis comparing the baseline, AdaSteer-Dynamic, and AdaSteer-Average methods on the Politeness dataset across three models Qwen-1.5, Llama-3, and Gemma-2.