Probe-Free Low-Rank Activation Intervention

Chonghe Jiang [*]
CUHK
chjiang@link.cuhk.edu.hk

Bao Nguyen [†]
CUHK
nbnguyen@se.cuhk.edu.hk

Anthony Man-Cho So
CUHK
manchoso@se.cuhk.edu.hk

Viet Anh Nguyen
CUHK
nguyen@se.cuhk.edu.hk

Abstract

Language models (LMs) can produce texts that appear accurate and coherent but contain untruthful or toxic content. Inference-time interventions that edit the hidden activations have shown promising results in steering the LMs towards desirable generations. Existing activation intervention methods often comprise an activation probe to detect undesirable generation, triggering the activation modification to steer subsequent generation. This paper proposes a probe-free intervention method **FLORAIN** for all attention heads in a specific activation layer. It eliminates the need to train classifiers for probing purposes. The intervention function is parametrized by a sample-wise nonlinear low-rank mapping, which is trained by minimizing the distance between the modified activations and their projection onto the manifold of desirable content. Under specific constructions of the manifold and projection distance, we show that the intervention strategy can be computed efficiently by solving a smooth optimization problem. The empirical results, benchmarked on multiple base models, demonstrate that FLORAIN consistently outperforms several baseline methods in enhancing model truthfulness and quality across generation and multiple-choice tasks. Our implementation can be found at https://github.com/nguyenngocbaocmt02/EFI.

1 Introduction

Transformer-based language models (LMs) have revolutionized generative modeling for natural language processing (NLP) (Radford et al., 2019; Brown et al., 2020; Anthropic, 2024; Jiang et al., 2023). The LM training pipeline in various NLP areas typically involves three steps: (i) provide the model with prompts that include demonstrations of the task, (ii) learn from the examples, and (iii) make predictions without additional training. In the generating process, there is evidence (Ji et al., 2023; Rawte et al., 2023; Xu et al., 2024) showing that LM output can contain undesirable contents, such as inaccurate answers, toxic answers, or answers without linguistic meaning.

One of the predominant solutions to the above issues is *activation intervention* (Subramani et al., 2022; Hernandez et al., 2023; Li et al., 2024b; Nguyen et al., 2025). Compared with previous studies on model editing (Zhang et al., 2023) and supervised fine-tuning (Li et al., 2024a), activation intervention alters the model activations during inference time. Therefore, it does not need to alter model weights using a subset of text samples, requiring fewer computational resources. However, new challenges must be addressed in the activation intervention task: (*Q1*) How to improve the quality of activation intervention through appropriate modeling with theoretical underpinnings? (*Q2*) How to enhance the computational efficiency of activation intervention by designing a low-cost intervention strategy?

For question (*Q1*), Burns et al. (2022) reveals latent knowledge inside the internal activations of a language model and finds a direction in activation space that satisfies logical consistency properties. Following this observation, several works focus on finding a good intervention systematically by two-stage methods (Yin et al., 2024; Li et al., 2024b; Pham and Nguyen, 2024). In the first stage, they train a classifier (the probe) on the activations of a network to identify a subset of attention heads that are most important for learning the specific task. In the second step, they propose intervention policies based on the probe and geometric transformation, e.g., linear operations (Li et al., 2024b), addictive offset bias vector (Yin et al., 2024), linear transformation using chance constrained programming (Nguyen et al., 2025), and householder transformation (Pham and Nguyen, 2024).

[*]equal contribution
[†]equal contribution

Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 2812–2824

The preliminary success of these activation intervention methods motivates us to improve the desirable generation of LMs. Specifically, we want to design an inference activation intervention method that does not require the 'detect and rectify' procedure, addressing (*Q1*) and (*Q2*) simultaneously. Our core idea is to depict the region of the desirable answers and consider the related low-rank transformation mapping for intervention. Based on these, the parameters in the intervention mapping are the solutions to a nonlinear low-rank optimization problem, which minimizes the distance between the intervention vector and its projection on the region. Compared with the previous region modelings of the LM activation space (Mamou et al., 2020; Janiak et al., 2024), we give the concrete formulation of the region with an analytical projection operator under the suitable distance measure.

Contributions. We propose Probe-**F**ree **Lo**w-**R**ank **A**ctivation **In**tervention (FLORAIN), a novel intervention method characterized by

- An ellipsoid model to capture the region of desirable output. In the training phase, we extract the activation heads of desirable answers to different questions and build the ellipsoid region models. The ellipsoid model is estimated using the first- and second-order statistical information of the activations extracted from the training data. To combat the ill-conditioning of the estimate due to the low sample size, we propose an extrapolation strategy to pull the region away from the region of *un*desirable outputs.

- A low-rank intervention mapping. We propose to minimize the gap between the post-intervention activation and its projection onto the desirable ellipsoid region. The low-rank intervention mapping is the product of a one-layer perception with another low-rank matrix variable. We show that the objective function admits an analytical form under the Mahalanobis distance projection. In addition, we show that the training problem is smooth and, thereby, can be calculated efficiently using a lightweight gradient descent method or its preconditioned version.

From the computational aspect, our FLORAIN intervention method edits the activation vectors on *one* layer of the Transformer network. This contrasts to ITI (Li et al., 2024b) that intervene different heads spread out in *multiple* layers of the network. Intervening in one layer, as FLORAIN does, has two significant advantages: First, it provides conditions for parallelization to reduce inference intervention time. Second, since the intervention vectors are concentrated in one layer, the strategy reduces the representation shifts by avoiding misleading intervention in the subsequent layers.

2 Related Work

Controllable generation. Model editing (Wang et al., 2023; Zhang et al., 2024) alters the model parameters to control the output, making it a powerful method for controllable generation. Another important category in controllable generation is fine-tuning, which includes Supervised Fine-Tuning (SFT, (Peng et al., 2023; Gunel et al., 2020)) and Reinforcement Learning from Human Feedback (RLHF, (Ouyang et al., 2022; Griffith et al., 2013)). These methods typically require altering model weights, incurring substantial resources and costs for computation.

Activation intervention at inference time is an emerging technique for controllable generation (Li et al., 2024b; Singh et al., 2024; Yin et al., 2024). Unlike model editing and fine-tuning techniques, the inference time intervention does not require altering the model parameters, leading to cheaper computational costs. Li et al. (2024b) proposes a headwise intervention method for eliciting truthful generated answers of a language model. Singh et al. (2024) considers the optimal transport plan between two empirical distributions to carry out the intervention. LoFit (Yin et al., 2024) identifies a specific subset of attention heads crucial for learning a particular task. It then fine-tunes the intervention vectors in those chosen heads. Another recent work (Pham and Nguyen, 2024) considers doing intervention activation using modified householder transformation based on the linear probe framework.

Region Modeling in LM. Various works aim to reveal how the semantic features influence the 'region' of embedding vectors in the transformer-based LM. The work (Mamou et al., 2020) utilizes mean-field theoretic manifold analysis to connect the geometry of feature representations with the linear separability of classes. Janiak et al. (2024) identifies stable regions in the residual stream of Transformers, where the model output remains insensitive to small activation changes but exhibits high sensitivity at the region boundaries. How-

ever, most prior works focus on specific findings in region modeling and overlook the potential of enhancing activation intervention through transport between two regions.

Low-Rank Optimization is widely studied in machine learning, statistics, and signal processing (Olikier et al., 2022; Zhanxuan et al., 2020; Cory-Wright et al., 2021). The motivation for formulating problems into low-rank optimization in several applications lies in two folds: the nature of the low-rank property of the ground truth and the goal for achieving lightweight complexity in algorithm design (McRae and Boumal, 2024). The low-rank concept has also been applied to LM model-editing, fine-tuning, and model compression (Hu et al., 2021; Hsu et al., 2022).

3 Nonlinear Low-Rank Intervention Mapping

This section defines the nonlinear low-rank activation intervention maps. In our probe-free intervention framework, this mapping serves as the computationally efficient intervention function during inference time.

For clarity, we consider a specific layer ℓ of the transformer network, and the index of the layer ℓ will be omitted in the following statement. This layer contains H attention heads; each head is of dimension d, so the output of this layer is a $D = d \times H$ dimensional activation vector. For an input x_i, the activation in the layer ℓ is denoted by $a_i = [a_{i1}; a_{i2}; \cdots; a_{iH}] \in \mathbb{R}^D$. Our inference-time intervention will modify the activation vector head-wise for a_{ih} for all $h \in [H]$.

The construction of the low-rank mapping involves two key considerations: (i) the intervention strategy should be generalized to different inputs, and (ii) the intervention strategy should not change a desirable answer too much. Motivated by these requirements, we construct our efficient nonlinear low-rank intervention as follows:

$$f : a \mapsto (I + L(a)R^\top)a + s, \tag{1a}$$

where $L(a) \in \mathbb{R}^{D\times k}$ is a low-rank matrix that depends on the input a, $R \in \mathbb{R}^{D\times k}$ is a constant low-rank matrix and $s \in \mathbb{R}^{D\times 1}$ is a vector. The parametrization of $L(a)$ is described as follows:

$$L_i(a) = \phi(W_i \circ a + b_i) \quad \forall i \in [k], \tag{1b}$$

where $L_i(a)$ and W_i refer to the i-th column of $L(a)$ and W, respectively. The notation $\circ$ denotes the Hadamard product of two vectors. The matrix $W \in \mathbb{R}^{D\times k}$ is the weight matrix, $b \in \mathbb{R}^{D\times k}$ is the bias term, and ϕ is the activation function applied component-wise to the input [1]. One candidate for this activation function is $\phi(\cdot) = \tanh(\cdot)$. This choice enhances intervention performance, as the mapping $x \mapsto \tanh(x)$ can output both negative and positive real numbers. Additionally, the output of the mapping is bounded between $(-1, 1)$, resolving the scaling issues of the bilinear terms $L(a)R^\top$ in formulation (1a) automatically. If the components of $L(a)$ are unbounded, one can multiply W by any positive constant κ and divide R by κ to obtain the same intervention. To address this issue, prior works on asymmetric low-rank optimization (Tu et al., 2016; Bhojanapalli et al., 2016) typically incorporate a regularization term to close the norm gap between the scales of two low-rank matrices. In contrast to these methods, the optimization problem in equation (2) does not require rescaling the two low-rank matrices $L(a)$ and R due to the boundness property of the hyperbolic tangent function. These properties also help stabilize the training process.

For a better understanding of the mapping constructed, we consider the degenerate case: if W is a zero matrix and ϕ is a linear function, then $L(a) = b$ and $f(a) = (I + bR^\top)a + s$. This represents the classical bilinear low-rank mapping, which does not depend on the input vector a.

4 Probe-Free Intervention Framework

We propose a probe-free intervention framework based on the low-rank mapping described above. The framework performs the following tasks: (i) models the desirable answer region, and (ii) computes the intervention parameters by solving the associated smooth optimization problem. We begin by presenting the data matrix and the framework setup.

From the training data, we collect the activations of the desirable output and form a matrix $G_q \in \mathbb{R}^{d\times|\mathcal{G}(q)|}$ for each attention head, where $|\mathcal{G}(q)|$ denotes the cardinality of the set $\mathcal{G}(q)$. Each column of G_q represents the activation of one desirable answer. We construct the undesirable matrix $B_q \in \mathbb{R}^{d\times|\mathcal{B}(q)|}$ using the same approach. Let $\mathcal{M}_q$ denote a generic manifold representing the region

[1] Note that in this context, the **activation function** refers to the concept used in neural networks and is unrelated to the **activation** intervention task.

of desirable answer data points. The nonlinear low-rank mapping f is trained by solving:

$$\min_{f} \sum_{q} \sum_{i \in \mathcal{B}(q) \cup \mathcal{G}(q)} c_q(f(a_i), \mathrm{Proj}_{\mathcal{M}_q}(f(a_i))), \tag{2}$$

where c_q denotes the distance measure for question q, $f(a_i)$ denotes the terminal point of the intervention on activation a_i, and $\mathrm{Proj}_{\mathcal{M}_q}$ denotes the projection onto the manifold $\mathcal{M}_q$. The performance of the intervention depends on the choice of c_q and the manifold $\mathcal{M}_q$. The manifold $\mathcal{M}_q$ should have a semantic interpretation in the modeling process and must be capable of separating the desirable answers from the undesirable answers. Additionally, solving problem (2) should be computationally efficient.

We now describe one specific instance of our framework. The manifold $\mathcal{M}_q$ for question q is chosen as an ellipsoid of the form

$$\mathcal{M}_q = \{x : (x - \hat{\mu}_q)^\top \hat{\Sigma}_q^{-1} (x - \hat{\mu}_q) \leq \rho_q\}, \tag{3a}$$

where $\hat{\mu}_q$ is the center of the ellipsoid, $\hat{\Sigma}_q$ is a positive definite matrix that prescribes the shape, or orientation, of the ellipsoid, and ρ_q is the radius of the ellipsoid.

Moreover, we choose c_q as the squared Mahalanobis distance (De Maesschalck et al., 2000)

$$c_q(a, a') = (a - a')^\top \hat{\Sigma}_q^{-1} (a - a'), \tag{3b}$$

which is the distance of the candidate point from the center of mass divided by the width of the ellipsoid in the direction of the candidate point. This distance is rooted in the construction of the manifold (3a). We use the following projection:

$$\mathrm{Proj}_{\mathcal{M}_q}(y) = \arg\min_{x \in \mathcal{M}_q} c_q(x, y). \tag{3c}$$

Theorem 4.1 gives the optimization problem under the Mahalanobis distance with analytical projection expression.

Theorem 4.1 (Ellipsoidal manifold and Mahalanobis distance)**.** *Suppose that the manifold, the distance measure, and the projection operator are chosen as in* (3)*. Problem* (2) *becomes*

$$\min_{f} \sum_{q} \sum_{i \in \mathcal{B}(q) \cup \mathcal{G}(q)} \left[\left(\sqrt{c_q(f(a_i), \hat{\mu}_q)} - \sqrt{\rho_q} \right)_+ \right]^2, \tag{4}$$

where $(y)_+ = \max\{0, y\}$.

The objective function for the optimization problem is straightforward to compute, as it transforms the distance between $(I + L(a)R)a + s$ and its projection into the distance between $(I + L(a)R)a + s$ and a fixed vector $\hat{\mu}_q$. Readers can refer to the appendix B for the proof of Theorem 4.1.

We now discuss how to derive the parameters to specify the manifold. The main challenge here is the high dimensionality of the activation vectors ($D = 4096$ for Llama3-8B), while for each question q, we observe fewer than ten samples in total [2]. Next, we present a practical method for estimating $\hat{\mu}_q$, $\hat{\Sigma}_q$, and the radius ρ_q. From the training data, we compute the question-specific mean vector for question q and the mean vector computed by all questions

$$\begin{cases} \hat{\mu}_q^+ = \frac{1}{|\mathcal{G}(q)|} \sum_{i=1}^{|\mathcal{G}(q)|} (G_q)_i \\ \hat{\mu}^+ = \frac{1}{N^+} \sum_q \sum_{i=1}^{|\mathcal{G}(q)|} (G_q)_i \end{cases} \tag{5}$$

where N^+ denotes the sample size of the desirable subset of data. Analogously, we compute $\hat{\mu}_q^-$ and $\hat{\mu}^-$ for the *un*desirable subset.

Due to the noise incurred when computing $\hat{\mu}_q^+$ with small sample sizes, additional information must be incorporated to accurately construct the mean vector $\hat{\mu}_q$. We propose the following extrapolation scheme:

$$\hat{\mu}_q = \hat{\mu}_q^+ + \lambda \left(\alpha \underbrace{\frac{\hat{\mu}_q^+ - \hat{\mu}_q^-}{\|\hat{\mu}_q^+ - \hat{\mu}_q^-\|}}_{\text{question specific}} + (1 - \alpha) \underbrace{\frac{\hat{\mu}^+ - \hat{\mu}^-}{\|\hat{\mu}^+ - \hat{\mu}^-\|}}_{\text{overall}} \right), \tag{6}$$

where $\lambda > 0$ controls the magnitude of extrapolation. The extrapolation direction consists of a question-specific direction dictated by $\hat{\mu}_q^+ - \hat{\mu}_q^-$, and an overall direction dictated by $\hat{\mu}^+ - \hat{\mu}^-$. Both terms aim to guide $\hat{\mu}_q$ *away from* the *un*desirable values $\hat{\mu}_q^-$ and $\hat{\mu}^-$. The parameter $\alpha \in [0, 1]$ controls the relative strength between the question-specific (local) and overall (global) directions. Li et al. (2024b) proposed a translation of the activation along the truthful direction $\hat{\mu}^+ - \hat{\mu}^-$, which coincides with the overall direction in our formula. The direction of extrapolation is illustrated in Figure 1, where the extrapolation vector is denoted by $\vec{d}_{move}$.

[2] Data collected from the TruthfulQA dataset.

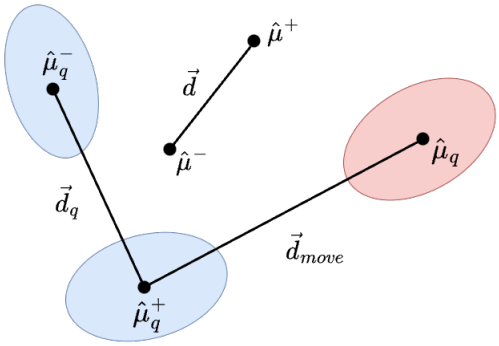

Figure 1: Exploration scheme in constructing question-wise mean vector $\hat{\mu}_q$, where the directions are computed as $\vec{d} = \hat{\mu}^+ - \hat{\mu}^-$, $\vec{d}_q = \hat{\mu}_q^+ - \hat{\mu}_q^-$, and then take the scaled convex combination $\vec{d}_{move} = \hat{\mu}_q - \hat{\mu}_q^+ = \lambda\left(\alpha\frac{\vec{d}}{\|\vec{d}\|} + (1-\alpha)\frac{\vec{d}_q}{\|\vec{d}_q\|}\right)$.

For the covariance matrix, we assume that $\hat{\Sigma}_q^{-1}$ is constant across all q. This assumption is a fundamental component of linear discriminant analysis in machine learning (Tharwat et al., 2017). We begin by computing the empirical covariance matrix:

$$S = \frac{1}{|N^+| - 1} \sum_q \sum_{i=1}^{|\mathcal{G}(q)|} \left[(G_q)_i - \hat{\mu}_q\right]\left[(G_q)_i - \hat{\mu}_q\right]^\top. \tag{7}$$

Due to small sample sizes, the empirical covariance matrix S may be non-invertible. To address this issue, we adopt the linear shrinkage method (Schäfer and Strimmer, 2005; Ledoit and Wolf, 2004), which shrinks the matrix towards a diagonal target, and set:

$$\hat{\Sigma}_q^{-1} = (\beta S + (1-\beta)\mathrm{diag}(S))^{-1} \quad \forall q,$$

where $\beta \in (0, 1)$ is the shrinkage parameter.

To estimate radius ρ, the heuristic approach is to find the minimum ρ such that all samples remain within the desirable answer region, centered at $\hat{\mu}_q^+$. For question q, we define:

$$\rho_q = \max_{i \in |\mathcal{G}_q|} [(G_q)_i - \hat{\mu}_q]^\top \hat{\Sigma}_q^{-1} [(G_q)_i - \hat{\mu}_q].$$

5 Algorithms for Training Nonlinear Low-Rank Mapping

The degenerated bilinear low-rank optimization problem has many applications in the statistical estimation field (e.g., matrix sensing, matrix recovery, PCA) (Chi et al., 2019). The literature shows that simple methods such as gradient descent can converge to the ground truth under mild assumptions with good initialization. In contrast, our objective function $(W, R, b, s) \mapsto \mathcal{F}(W, R, b, s)$ cannot be represented by the factorization formulation and contains an extra nonlinear activation function, making it challenging to derive similar theoretical guarantees. However, the first-order algorithm is still a good choice with lightweight computational complexity.

We emphasize that the activation function $\phi(\cdot) = \tanh(\cdot)$ introduces smoothness to the objective function, thereby stabilizing the optimization processes during gradient descent. Furthermore, we propose to use preconditioned gradient descent, a powerful approach that has gained popularity in (bilinear) low-rank optimization. This method effectively balances the scale at each iteration, accelerating convergence while maintaining the same computational complexity.

$$\begin{aligned}
W_{t+1} &= W_t - \eta \nabla_W \mathcal{F}(W_t, R_t, b_t, s_t)\left(R_t^\top R_t + \epsilon I\right)^{-1}, \\
R_{t+1} &= R_t - \eta \nabla_R \mathcal{F}(W_t, R_t, b_t, s_t)\left(W_t^\top W_t + \epsilon I\right)^{-1}, \\
b_{t+1} &= b_t - \eta \nabla_b \mathcal{F}(W_t, R_t, b_t, s_t)\left(b_t^\top b_t + \epsilon I\right)^{-1}, \\
s_{t+1} &= s_t - \eta \nabla_s \mathcal{F}(W_t, R_t, b_t, s_t)\left(s_t^\top s_t + \epsilon I\right)^{-1},
\end{aligned} \tag{8}$$

where ϵ is a small number to ensure the scaling term is invertible. In comparison to vanilla gradient descent, the search directions of the variables in equation (8) are scaled. Intuitively, this scaling acts as a preconditioner, enhancing the search direction and enabling the use of larger step sizes. The preconditioner is adaptive and varies across iterations. Computationally, the scaled gradient descent introduces minimal overhead, as the inversion of the small-sized matrix is computationally inexpensive. Therefore, the per-iteration cost remains in the same order as standard gradient descent. In the context of bilinear low-rank matrix estimation, specifically in the problem $\min_{b,R} f(bR^\top)$ (Tong et al., 2021), it has been shown that scaled gradient descent achieves a condition number-free convergence rate that outpaces standard gradient descent under mild assumptions.

6 Empirical Results

This section presents the empirical results of our proposed algorithm FLORAIN. Section 6.1 clarifies the setting of our experiments on the TruthfulQA dataset, including details about datasets, tasks, metrics, baselines, and computational resources. Section 6.2 showcases the superiority of

our proposed methods to other baselines. We introduce the above contents in the Toxic Comments Classification Challenge dataset in Appendix C.

6.1 Experimental Setup

Tasks and Metrics: We evaluate our framework on the multiple choice and generation tasks, which are commonly used to verify the truthfulness of language models:

- In the generation task, the model produces a complete answer for each question using greedy autoregressive decoding. While the accuracy and informativeness of the answers are ideally assessed by humans, this evaluation method is both costly and time-consuming. As a result, most works in this area rely on a well-trained large language model as an alternative evaluation tool. In our approach, we employ two fine-tuned `GPT-3.5-instruct` models: one for classifying whether the answer is correct or incorrect, and another for determining whether the response is informative. According to Li et al. (2024b), the percentage of answers labeled as correct by the first model is referred to as the truthful score (True %). Meanwhile, the percentage of answers labeled as informative by the second model corresponds to the informative score (Info %). Following the methodology of Li et al. (2024b), we report both the truthful score (True %) and the product of the truthful and informative scores, True*Info (%).

- In the multiple-choice task, the model computes the log probability of completion for a given question and its corresponding set of choices. Following the approach of Lin et al. (2021), we report two metrics: MC1 and MC2. MC1 identifies the correct answer as the one with the highest probability among the available choices, and the overall accuracy across all questions is denoted as MC1. MC2 represents the normalized total probability assigned to the set of true answers, given a question and its associated set of true/false reference answers.

- We also include two other metrics named Kullback-Leiber divergence (KL) of the model's next-token prediction distribution post-versus-pre-intervention and Cross-Entropy Loss (CE). These two metrics assess the extent to which the generation distribution shifts after the intervention. Lower values are preferred, indicating that the intervention minimally alters the behavior of the original model, reducing the likelihood of producing abnormal characters or unnatural sentences. The calculation of these metrics is detailed in Li et al. (2024b).

Datasets: We use the TruthfulQA dataset to benchmark the effectiveness of FLORAIN. This dataset consists of 817 questions across 38 categories, designed to elicit false answers from language models, posing a challenge to generating accurate responses. Following the splits provided by Li et al. (2024b) and Yin et al. (2024), we divide the dataset into training (326 questions), validation (82 questions), and test (407 questions) sets. For evaluation, we perform 2-fold cross-validation, as done in Li et al. (2024b) and Yin et al. (2024). Additionally, we process the dataset into 5,918 question-answer pairs, each labeled with a binary indicator of desirability. The training set is used to develop our intervention policy, while the validation set is reserved for parameter tuning.

Models: We apply our method to multiple pre-trained models to enhance their truthfulness, including Llama-7B (Touvron et al., 2023b), Llama2-chat-13B (Touvron et al., 2023a), and Llama3-8B (Dubey et al., 2024). Our approach can be integrated into existing fine-tuning pipelines, helping to generate more accurate answers. To demonstrate its versatility, we also evaluate our method on models that have already been fine-tuned for the same task, treating these as both baseline models and competitors to our approach. Detailed descriptions of these models are provided in the baselines section.

Hyperparameters: We manipulate the behavior of FLORAIN by tuning two hyperparameters λ and α in determining the $\hat{\mu}_q$ by equation (6). The details about selecting them for each pre-trained model are discussed in Appendix A.

Baselines: We compare FLORAIN against competitive baselines with the same goal of eliciting truthful answers from language models:

- Inference-time Intervention (ITI, Li et al. 2024b) is a state-of-the-art method that allows intervention without finetuning. We follow the hyperparameter settings provided in the original paper (Li et al., 2024b) and their GitHub repository.[3]

[3] https://github.com/likenneth/honest_llama/tree/master

- Few-shot prompting (FSP), introduced by Bai et al. (2022), has demonstrated the effectiveness of prompting with 50 examples on the TruthfulQA benchmark. In our experiments, we use the 50-shot version, selecting 50 prompt pairs from the training set.

- Instruction Fine-Tuning (IFT) (Wang et al., 2022; Chung et al., 2024) is a well-known approach to improving the truthfulness of language models through fine-tuning. For comparison, we include two prominent models in this category: Alpaca-7B (Taori et al., 2023) and Vicuna-7B (Chiang et al., 2023).

Computation resources. Our method and baselines run on four NVIDIA RTX A5000 GPUs, an i9 14900K CPU, and 128GB RAM.

6.2 Numerical Results

6.2.1 Comparison between Truthfulness Augmentation Baselines across Different LMs

Table 1 presents the numerical results of our method, ITI, and Few-shot prompting (FSP) across three base models: Llama-7B, Llama3-8B, and Llama2-chat-13B on the TruthfulQA dataset. Notably, FSP is a prompt engineering technique orthogonal to intervention methods like FLORAIN and ITI. Therefore, we also report results for combinations of FSP with either ITI or FLORAIN. The FSP + FLORAIN combination consistently achieved the highest performance in metrics such as True * Info and True across all three models, with scores reaching 45% for Llama-7B, 42% for Llama3-8B, and 61% for Llama2-chat-13B.

When applied independently, FLORAIN outperforms ITI in four key metrics across both the generation and multiple-choice tasks, significantly enhancing the truthfulness and quality of the pre-intervention models. Notably, FLORAIN improves the True * Info(%) score from 21.15 to 31.46 for Llama-7B, 32.88 to 36.78 for Llama3-8B, and 51.87 to 60.68 for Llama2-chat-13B. The KL values remain minimal across all models, indicating that the intervention effect is not overly aggressive.

6.2.2 Comparison between ITI, FLORAIN, and Instruction Finetuning Methods.

In this experiment, we benchmark two instruction fine-tuning methods, Alpaca and Vicuna, to assess whether our method further improves their quality and truthfulness. As shown in Table 2, FLORAIN demonstrates significant effectiveness, boosting the True * Info score by 8.07% for Alpaca and 14.21% for Vicuna. In both models, FLORAIN outperforms ITI across all quality metrics, highlighting its ability to enhance both accuracy and truthfulness.

7 Conclusion and Future Directions

In this paper, we introduced FLORAIN, a novel probe-free low-rank intervention framework for activation intervention. The framework aims to minimize the distance between the post-intervention vector and its projection onto the desirable answer region, an ellipsoid that is carefully estimated using first-order and second-order statistical information. Unlike existing intervention methods, our probe-free approach does not require classifiers to identify the responsible heads or layers for intervention. Instead, it formulates an optimization problem that incorporates region modeling and analytical projection. Additionally, the intervention is performed in a single layer of the output transformer network, allowing for efficient parallel computation and mitigating distribution shift phenomena. A potential direction for future research is to extend the region modeling approach, such as by considering the subspace spanned by the desirable answer matrix as the desirable answer region.

8 Limitations

Our method relies on the scale of the training dataset. In cases where the number of training samples is small, constructing the ellipsoid region becomes more challenging, as the estimated mean and covariance matrix are more susceptible to higher bias. The smooth optimization problem derived in the main text exhibits nonconvexity. Due to our limited understanding of its optimization landscape, the algorithm may converge to local minimizers. As a result, our first-order algorithm does not guarantee convergence to the global optimum. To solve this problem, we employ standard gradient descent without implementing preconditioning.

Acknowledgments. Viet Anh Nguyen gratefully acknowledges the generous support from the UGC Early Career Scheme Grant 24210924 and the CUHK's Improvement on Competitiveness in Hiring New Faculties Funding Scheme.

Table 1: Quantitative results of different methods on TruthfulQA dataset, across different Language Models.

Methods	True * Info (%) ↑	True (%) ↑	MC1 ↑	MC2 ↑	CE ↓	KL ↓
Unintervened	21.15	22.16	25.58	40.54	2.13	0.00
ITI	26.52	28.03	27.78	43.59	2.20	0.07
FLORAIN (ours)	**31.46**	**34.72**	**31.76**	**47.43**	2.21	0.09
FSP	36.13	39.78	34.03	50.34	2.13	0.00
FSP + ITI	40.63	45.16	35.50	52.48	2.20	0.07
FSP + FLORAIN (ours)	**45.31**	**49.23**	**36.45**	**54.27**	2.20	0.08

(a) Llama-7B

Methods	True * Info (%) ↑	True (%) ↑	MC1 ↑	MC2 ↑	CE ↓	KL ↓
Unintervened	32.88	44.18	30.36	48.98	2.38	0.00
ITI	35.92	46.88	32.07	49.84	2.50	0.13
FLORAIN (ours)	**36.78**	**48.67**	**34.56**	**53.68**	2.51	0.14
FSP	36.32	39.78	35.74	52.93	2.38	0.00
FSP + ITI	40.63	45.16	35.50	52.98	2.48	0.14
FSP + FLORAIN (ours)	**42.15**	**47.32**	**36.98**	**55.83**	2.51	0.16

(b) Llama3-8B

Methods	True * Info (%) ↑	True (%) ↑	MC1 ↑	MC2 ↑	CE ↓	KL ↓
Unintervened	51.87	59.86	35.38	53.32	2.31	0.00
ITI	57.02	63.04	37.46	55.59	2.32	0.17
FLORAIN (ours)	**60.68**	**67.70**	**39.65**	**59.57**	2.35	0.18
FSP	55.97	58.63	40.76	57.84	2.31	0.00
FSP + ITI	56.78	59.24	41.50	59.01	2.33	0.13
FSP + FLORAIN (ours)	**61.14**	**62.45**	**44.52**	**61.48**	2.37	0.16

(c) Llama2-chat-13B

Table 2: Quantitative results of intervention methods on instruction-finetuned models Alpaca and Vicuna.

Methods	True*Info (%) ↑	True (%) ↑	MC1 ↑	MC2 ↑	CE ↓	KL ↓
Alpaca	30.39	30.85	26.56	41.63	2.81	0.00
Alpaca + ITI	37.67	38.19	28.89	45.19	2.88	0.14
Alpaca + FLORAIN (ours)	**38.56**	**40.52**	**31.32**	**46.83**	2.85	0.15
Vicuna	38.24	42.10	31.83	48.48	2.67	0.00
Vicuna + ITI	49.27	53.25	33.42	51.80	2.77	0.26
Vicuna + FLORAIN (ours)	**52.45**	**57.81**	**36.86**	**56.76**	2.78	0.27

References

AI Anthropic. 2024. The Claude 3 model family: Opus, Sonnet, Haiku. *Claude-3 Model Card*.

Bai et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Srinadh Bhojanapalli, Anastasios Kyrillidis, and Sujay Sanghavi. 2016. Dropping convexity for faster semi-definite optimization. In *Conference on Learning Theory*, pages 530–582. PMLR.

Brown et al. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. 2022. Discovering latent knowledge in language models without supervision. *arXiv preprint arXiv:2212.03827*.

Yuejie Chi, Yue M Lu, and Yuxin Chen. 2019. Nonconvex optimization meets low-rank matrix factorization: An overview. *IEEE Transactions on Signal Processing*, 67(20):5239–5269.

Chiang et al. 2023. Vicuna: An open-source chatbot impressing GPT-4 with 90% ChatGPT quality. *See https://vicuna.lmsys.org (accessed 14 April 2023)*, 2(3):6.

Chung et al. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Ryan Cory-Wright, Jean Pauphilet, and Andrea Lodi. 2021. A new perspective on low-rank optimization. *arXiv preprint arXiv:2105.05947*.

Roy De Maesschalck, Delphine Jouan-Rimbaud, and Désiré L Massart. 2000. The Mahalanobis distance. *Chemometrics and Intelligent Laboratory Systems*, 50(1):1–18.

Abhimanyu Dubey et al. 2024. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Shane Griffith, Kaushik Subramanian, Jonathan Scholz, Charles L Isbell, and Andrea L Thomaz. 2013. Policy shaping: Integrating human feedback with reinforcement learning. *Advances in Neural Information Processing Systems*, 26.

Beliz Gunel, Jingfei Du, Alexis Conneau, and Ves Stoyanov. 2020. Supervised contrastive learning for pre-trained language model fine-tuning. *arXiv preprint arXiv:2011.01403*.

Evan Hernandez, Belinda Z Li, and Jacob Andreas. 2023. Measuring and manipulating knowledge representations in language models. *arXiv preprint arXiv:2304.00740*.

Yen-Chang Hsu, Ting Hua, Sungen Chang, Qian Lou, Yilin Shen, and Hongxia Jin. 2022. Language model compression with weighted low-rank factorization. *arXiv preprint arXiv:2207.00112*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Jett Janiak, Jacek Karwowski, Chatrik Singh Mangat, Giorgi Giglemiani, Nora Petrova, and Stefan Heimersheim. 2024. Characterizing stable regions in the residual stream of LLMs. *arXiv preprint arXiv:2409.17113*.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Albert Q Jiang et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Olivier Ledoit and Michael Wolf. 2004. A well-conditioned estimator for large-dimensional covariance matrices. *Journal of Multivariate Analysis*, 88(2):365–411.

Junyi Li, Tianyi Tang, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2024a. Pre-trained language models for text generation: A survey. *ACM Computing Surveys*, 56(9):1–39.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2024b. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, 36.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2021. TruthfulQA: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*.

Jonathan Mamou, Hang Le, Miguel Del Rio, Cory Stephenson, Hanlin Tang, Yoon Kim, and SueYeon Chung. 2020. Emergence of separable manifolds in deep language representations. *arXiv preprint arXiv:2006.01095*.

Andrew D McRae and Nicolas Boumal. 2024. Benign landscapes of low-dimensional relaxations for orthogonal synchronization on general graphs. *SIAM Journal on Optimization*, 34(2):1427–1454.

Bao Nguyen, Binh Nguyen, Duy Nguyen, and Viet Anh Nguyen. 2025. Risk-aware distributional intervention policies for language models. *arXiv preprint arXiv:2501.15758*.

Guillaume Olikier, P-A Absil, and Julien M Hendrickx. 2022. Low-rank optimization methods based on projected-projected gradient descent that accumulate at bouligand stationary points. *arXiv preprint arXiv:2201.03962*.

Long Ouyang et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with GPT-4. *arXiv preprint arXiv:2304.03277*.

Van-Cuong Pham and Thien Huu Nguyen. 2024. Householder pseudo-rotation: A novel approach to activation editing in LLMs with direction-magnitude perspective. *arXiv preprint arXiv:2409.10053*.

Radford et al. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

Vipula Rawte, Swagata Chakraborty, Agnibh Pathak, Anubhav Sarkar, SM Tonmoy, Aman Chadha, Amit P Sheth, and Amitava Das. 2023. The troubling emergence of hallucination in large language models–an extensive definition, quantification, and prescriptive remediations. *arXiv preprint arXiv:2310.04988*.

Juliane Schäfer and Korbinian Strimmer. 2005. A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Statistical Applications in Genetics and Molecular Biology*, 4(1).

Shashwat Singh, Shauli Ravfogel, Jonathan Herzig, Roee Aharoni, Ryan Cotterell, and Ponnurangam Kumaraguru. 2024. MiMiC: Minimally modified counterfactuals in the representation space. *arXiv preprint arXiv:2402.09631*.

Nishant Subramani, Nivedita Suresh, and Matthew E Peters. 2022. Extracting latent steering vectors from pretrained language models. *arXiv preprint arXiv:2205.05124*.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Alpaca: A strong, replicable instruction-following model. *Stanford Center for Research on Foundation Models*, 3(6):7.

Alaa Tharwat, Tarek Gaber, Abdelhameed Ibrahim, and Aboul Ella Hassanien. 2017. Linear discriminant analysis: A detailed tutorial. *AI Communications*, 30(2):169–190.

Tian Tong, Cong Ma, and Yuejie Chi. 2021. Accelerating ill-conditioned low-rank matrix estimation via scaled gradient descent. *Journal of Machine Learning Research*, 22:1–63.

Hugo Touvron et al. 2023a. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Hugo Touvron et al. 2023b. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Stephen Tu, Ross Boczar, Max Simchowitz, Mahdi Soltanolkotabi, and Ben Recht. 2016. Low-rank solutions of linear matrix equations via Procrustes Flow. In *International Conference on Machine Learning*, pages 964–973. PMLR.

Wang et al. 2023. Knowledge editing for large language models: A survey. *arXiv preprint arXiv:2310.16218*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. Self-instruct: Aligning language models with self-generated instructions. *arXiv preprint arXiv:2212.10560*.

Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. 2024. Hallucination is inevitable: An innate limitation of large language models. *arXiv preprint arXiv:2401.11817*.

Fangcong Yin, Xi Ye, and Greg Durrett. 2024. Lofit: Localized fine-tuning on LLM representations. *arXiv preprint arXiv:2406.01563*.

Zhang et al. 2024. A comprehensive study of knowledge editing for large language models. *arXiv preprint arXiv:2401.01286*.

Hanqing Zhang, Haolin Song, Shaoyu Li, Ming Zhou, and Dawei Song. 2023. A survey of controllable text generation using transformer-based pre-trained language models. *ACM Computing Surveys*, 56(3):1–37.

Hu Zhanxuan, Nie Feiping, Wang Rong, and Li Xuelong. 2020. Low rank regularization: A review. *Information Sciences*.

A Hyper-parameter Tuning

In our framework, two key hyperparameters are λ and α. We determine their values by evaluating performance metrics and text generation quality in the validation set. Through a grid search over $\{2, 3, 4, 5\}$ for λ and $\{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$ for α, we find that setting $\alpha = 0.2$ and $\lambda = 5$ yields strong performance across various cases. As a result, unless otherwise specified, we adopt this combination for all our experiments.

Additionally, the choice of the intervened layer is crucial. Given the limited number of layers, we can efficiently conduct a search procedure to optimize validation test performance. Specifically, we intervene at layer 11 for Llama-7B, Alpaca-7B, Vicuna-7B, layer 12 for Llama3-8B, and layer 14 for Llama2-chat-13B.

B Proof of Theorem 4.1

The intervention maps the activation of undesirable answer $a_i, i \in \mathcal{B}(q)$ onto the ellipsoid region of the desirable answers, which can be defined below.

$$\mathcal{E}_q = \{x : (x - \hat{\mu}_q)^\top \hat{\Sigma}_q^{-1} (x - \hat{\mu}_q) \leq \rho_q\} \quad (9)$$

In the proof part, we simplify the notation of $\hat{\Sigma}_q$ (resp. $\mathcal{E}_q, \hat{\mu}_q$) as Σ (resp. $\mathcal{E}, \mu$) for clarity. Let the Mahalanobis projection be

$$\mathrm{Proj}_{\mathcal{E}}(y) = \arg\min_{x \in \mathcal{E}} (y - x)^\top \Sigma^{-1} (y - x),$$

where

$$\mathcal{E} = \{x : (x - \mu)^\top \Sigma^{-1} (x - \mu) \leq \rho\}.$$

Proof of Theorem 4.1. If $y \in \mathcal{E}$, the result can be immediately derived by the definition of the projection. If $y \notin \mathcal{E}$, we can transform the coordinates of x, y to simplify the projection problem. We define

$$\hat{y} = \Sigma^{-\frac{1}{2}} y, \quad \hat{x} = \Sigma^{-\frac{1}{2}} x.$$

The projection problem has been transformed into

$$\arg\min_{\hat{x} \in \hat{\mathcal{E}}} \|\hat{y} - \hat{x}\|_2^2,$$

where $\hat{\mathcal{E}} = \{\hat{x} : \|\hat{x} - \Sigma^{-\frac{1}{2}} \mu\|_2^2 \leq \rho\}$.

The transformed problem allows for a closed-form solution for the point outside of the region

$$\begin{aligned} \hat{x}^* &= \Sigma^{-\frac{1}{2}} \mu + \sqrt{\rho} \frac{\hat{y} - \Sigma^{-\frac{1}{2}} \mu}{\|\hat{y} - \Sigma^{-\frac{1}{2}} \mu\|_2} \\ &= \Sigma^{-\frac{1}{2}} \mu + \sqrt{\rho} \frac{\Sigma^{-\frac{1}{2}} y - \Sigma^{-\frac{1}{2}} \mu}{\|\Sigma^{-\frac{1}{2}} y - \Sigma^{-\frac{1}{2}} \mu\|_2}. \end{aligned}$$

Therefore

$$\begin{aligned} x^* &= \mu + \sqrt{\rho} \frac{y - \mu}{\|\Sigma^{-\frac{1}{2}} y - \Sigma^{-\frac{1}{2}} \mu\|} \\ &= \mu + \frac{\sqrt{\rho}}{\sqrt{(y - \mu)^\top \Sigma^{-1} (y - \mu)}} (y - \mu). \end{aligned}$$

For an arbitrary q and $i \in \mathcal{B}(q) \cup \mathcal{G}(q)$,

$$\begin{aligned} & f(a_i) - \mathrm{Proj}(f(a_i)) \\ &= f(a_i) - \hat{\mu}_q \\ &- \frac{\sqrt{\rho_q}(f(a_i) - \hat{\mu}_q)}{\sqrt{(f(a_i) - \hat{\mu}_q)^\top \hat{\Sigma}^{-1} (f(a_i) - \hat{\mu}_q)}} \\ &= [1 - \frac{\sqrt{\rho_q}}{\sqrt{(f(a_i) - \hat{\mu}_q)^\top \hat{\Sigma}^{-1} (f(a_i) - \hat{\mu}_q)}}] \\ &\quad (f(a_i) - \hat{\mu}_q). \end{aligned}$$

By computation, if $f(a_i) \in \mathcal{M}_q$, then

$$c_q(f(a_i), \mathrm{Proj}_{c_q, \mathcal{M}_q}(f(a_i)) = 0.$$

Otherwise, the distance is

$$[\sqrt{(f(a_i) - \hat{\mu}_q)^\top \hat{\Sigma}^{-1} (f(a_i) - \hat{\mu}_q)} - \sqrt{\rho}]^2.$$

We complete the proof. □

C Experimental results on the toxicity mitigation task

C.1 Experimetal Setup

Tasks: In this task, language models must complete an incomplete prefix of text. Typically, the prefix is chosen to provoke toxic content from LLMs. We evaluate the models based on three key metrics: toxicity, fluency, and diversity. For each prompt in the dataset, the models generate 25 responses, each limited to 20 tokens. These outputs are analyzed using the Perspective API [4], which estimates the likelihood that a human would perceive the text as toxic.

Metrics:

- Expected Maximum Toxicity (Exp. Max. Tox.): We identify the highest toxicity score for every output and compute the average of these maximum scores across all prompts.
- Toxic Completion Proportion (Tox. Prob.): This metric tracks the fraction of outputs considered toxic, where toxicity is defined as a score above 0.5 based on the threshold of Perspective API.

[4] https://perspectiveapi.com/

- Fluency is evaluated by calculating the perplexity values of the generated outputs, using GPT-2 (XL) as a reference model. Lower perplexity values suggest that the text is more coherent and fluent.

- Diversity is assessed by examining the ratio of unique n-grams (1-gram, 2-gram, and 3-gram) to the total number of tokens in the generated text. This metric captures the range of variation in the outputs, with higher values indicating more diverse and varied language use.

Training Dataset: We use the Toxic Comments Classification Challenge data.[5] The dataset comprises sentences and their human toxicity labels. Following existing works in the field, we adopt GPT2-Large as the base model across all experiments of the toxicity mitigation task. We include several baselines that have the same goal of reducing the toxicity of large language models (LLMs). As for MIMIC, we consider two versions: Mean Matching (MM) and Mean+Covariance Matching (MCM).

C.2 Comparison

The experimental results of baselines are shown in Table 3, where the base model used by all methods is GPT-2 Large. The result of the original model is described in the first row. We split baselines into two groups. The first one using an extensive finetuning procedure comprises DAPT, GeDI, PPLM, UDDIA, DExperts, and GOODTRIEVER, while the second group contains inference time finetuning-free methods like MIMIC, ITI, and FLORAIN. Baselines in the first group are better than counterparts in the second group regarding toxicity metrics. However, these methods necessitate either fine-tuning or computing gradients at inference time, which can be computationally intensive. MIMIC, ITI, and FLORAIN achieved a toxicity reduction comparable to that of many algorithms in the first group. Specifically, FLORAIN is superior to PPLM and equally competitive to DAPT. Notably, within the second group, FLORAIN offers the best toxicity reduction impact than ITI and MIMIC while maintaining a better fluency and diversity of generated sentences. The fluency of FLORAIN is even more favored than almost all algorithms in the first group except for UDDIA. At the same time, its diversity metric is better than that of other baselines apart from PPLM.

[5] https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge

Table 3: Quantitative results of our method and baselines on toxicity mitigation task.

Model	Exp. Max. Tox. ↓	Tox. Prob. ↓	Fluency ↓	1-gram ↑	2-gram ↑	3-gram ↑
GPT-2 (large)	0.39	0.25	24.66	0.58	0.85	0.85
DAPT	0.27	0.09	30.27	0.57	0.84	0.84
GeDI	0.24	0.06	48.12	0.62	0.84	0.83
PPLM (10%)	0.38	0.24	32.58	0.58	0.86	0.86
UDDIA	0.24	0.04	26.83	0.51	0.80	0.83
DExperts	0.21	0.02	27.15	0.56	0.84	0.84
GOODTRIEVER	0.22	0.04	27.11	0.58	0.82	0.83
MM (MIMIC)	0.33	0.16	28.00	0.58	0.85	0.85
MCM (MIMIC)	0.29	0.09	30.70	0.54	0.84	0.84
ITI	0.31	0.12	33.12	0.57	0.85	0.85
FLORAIN	0.27	0.10	28.11	0.58	0.85	0.85