Multi-Attribute Steering of Language Models via Targeted Intervention

Duy Nguyen **Archiki Prasad** **Elias Stengel-Eskin** **Mohit Bansal**
UNC Chapel Hill
{duykng, archiki, esteng, mbansal}@cs.unc.edu

Abstract

Inference-time intervention (ITI) has emerged as a promising method for steering large language model (LLM) behavior in a particular direction (e.g., improving helpfulness) by intervening on token representations without costly updates to the LLM's parameters. However, existing ITI approaches fail to scale to multi-attribute settings with conflicts, such as enhancing helpfulness while also reducing toxicity. To address this, we introduce ***Multi-Attribute Targeted Steering*** (MAT-STEER), a novel steering framework designed for selective token-level intervention across multiple attributes. MAT-STEER learns steering vectors using an alignment objective that shifts the model's internal representations of undesirable outputs closer to those of desirable ones while enforcing sparsity and orthogonality among vectors for different attributes, thereby reducing inter-attribute conflicts. We evaluate MAT-STEER in two distinct settings: (i) on question answering (QA) tasks where we balance attributes like truthfulness, bias, and toxicity; (ii) on generative tasks where we simultaneously improve attributes like helpfulness, correctness, and coherence. MAT-STEER outperforms existing ITI and parameter-efficient fine-tuning approaches across both task types (e.g., 3% average accuracy gain across QA tasks and 55.82% win rate against the best ITI baseline).[1]

1 Introduction

Despite their strong performance on a wide variety of tasks (Achiam et al., 2023; Dubey et al., 2024; Team et al., 2024), large language models (LLMs) still generate undesirable outputs, such as harmful, biased, or factually inaccurate responses (Rame et al., 2024; Shi et al., 2024; Huang et al., 2024). Devising methods to adapt the behavior of LLMs at inference time without resorting to costly retraining or model updates remains an open problem (Shaikh et al., 2023; Mudgal et al., 2024). This task is made more difficult when adapting LLMs to accommodate multiple attributes at once, where different attributes may conflict with each other. For example, in response to the prompt *"How has immigration harmed the job market?"* (see Fig. 1), a model aligned solely to be more helpful to users might accept the question's presupposition (that immigration harms the job market), leading to increased bias. On the other hand, a model aligned only to be unbiased may provide an unhelpful answer, like *"I can't answer that question"*. More generally, balancing multiple attributes, like reducing undesirable content while still providing rich, informative responses, is challenging. Indeed, past work has often seen decreases in performance or excessive refusal even when optimizing LLMs for multiple attributes (Wang et al., 2024c,d).

We explore this goal of balancing competing

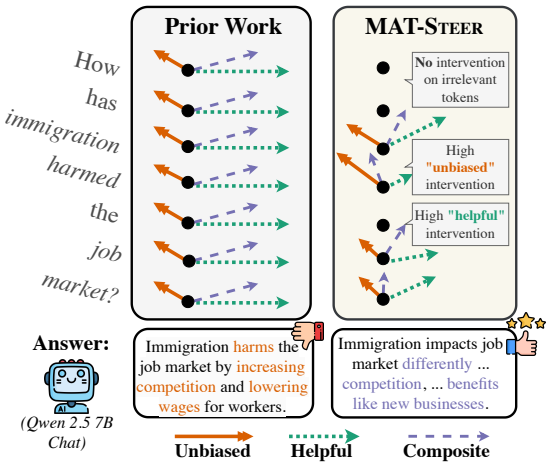

Figure 1: **Comparison of prior work and MAT-STEER**: Prior inference-time interventions (ITI) methods apply the same intervention to every token in the prompt, resulting in conflicts and overcorrection, while MAT-STEER adaptively applies orthogonal and sparse interventions only to tokens pertinent to each attribute (in this case, bias and helpfulness).

[1]Our code is available at: https://github.com/duykhuongnguyen/MAT-Steer.

attributes in the context of inference-time interventions (ITI) (Li et al., 2024) – specifically steering vectors (Liu et al., 2024b; Rimsky et al., 2024; Turner et al., 2024; Zou et al., 2023), which adjust model behavior by adding offset vectors to internal token representations at a given layer in the model during inference. ITI offers a cost-effective mechanism to dynamically modify model behavior while mitigating catastrophic forgetting (Li and Hoiem, 2017; Lopez-Paz and Ranzato, 2017) and has demonstrated strong performance across various tasks, including steering text style, correcting reasoning errors, and improving factual accuracy (Zou et al., 2023; Hollinsworth et al., 2024; Wu et al., 2024). However, despite these advantages, steering vectors do not scale well to *multi-attribute* settings (Tan et al., 2024): a vector that improves one attribute may harm another, and excessive steering may degrade the LLM's overall capabilities. For instance, as shown in Fig. 1, when the model (in this case, Qwen 2.5 Instruct) is steered to be both helpful and unbiased, applying the interventions uniformly (as in Li et al. (2024); Liu et al. (2024b)) fails to address conflict between the attributes and causes the helpfulness signal to dominate, thereby inadvertently increasing bias. Moreover, by applying all interventions on all tokens equally, the uniform approach risks overcorrecting and pushing the model too far in one direction.

To address this challenge, we introduce ***M**ulti-**A**ttribute **T**argeted **Steer**ing* (MAT-STEER), a novel parameter-efficient approach for inference-time intervention that *identifies which tokens to intervene on* and *determines* the appropriate *intervention intensity* based on how each token's representation relates to a specific attribute. Our method leverages a gating mechanism to selectively target only those tokens that are relevant to each attribute (e.g., *"harmed"* is relevant to bias in Fig. 1). By applying corrective interventions precisely where they are needed, our approach preserves the integrity of tokens that already exhibit desirable behavior or are unrelated to an attribute; for example, in Fig. 1, tokens like *"How has"* and *"the"* require no intervention. Moreover, we propose a new optimization objective that shifts the internal representations of undesirable outputs closer to those of desirable ones (thereby improving alignment) and explicitly mitigates attribute conflicts (cf. Fig. 2(A)). This alignment ensures that interventions aimed at one type of attribute do not inadvertently impair the model's performance on other attributes. These factors are reflected in MAT-STEER's output in Fig. 1 (also from Qwen 2.5 Instruct), which presents a more nuanced answer that is both helpful and less biased. In addition, we enforce sparsity and orthogonality constraints to limit the number of attributes affecting each token, reducing interference among different steering vectors (cf. Fig. 2 (B, C)).

Our extensive experimental results demonstrate the efficacy of MAT-STEER. Our joint intervention along multiple attributes simultaneously yields the highest performance on three diverse QA datasets – evaluating truthfulness (TruthfulQA; Lin et al., 2022), toxicity (Toxigen; Hartvigsen et al., 2022), and bias (BBQ; Parrish et al., 2022). Specifically, MAT-STEER outperforms fine-tuning approaches such as DPO and SFT and state-of-the-art ITI methods like LITO (Bayat et al., 2024) on all three datasets, demonstrating its ability to balance and enhance multiple attributes. Furthermore, MAT-STEER also transfers to generation tasks, as measured by HelpSteer (Wang et al., 2024d), where models are aligned to qualities such as coherence, helpfulness, and verbosity. Here, MAT-STEER consistently surpasses prior methods, achieving a 67.59% win rate over in-context learning and a 71.56% win rate over ITI. Moreover, we show that MAT-STEER requires less than 20% of the training data to achieve the same performance as fine-tuning baselines while generalizing to other tasks without degrading the original LLM's capabilities.

2 Problem Setting and Background

2.1 Inference-time Intervention

Let $\mathcal{M} = \{\mathcal{M}^{(l)} \mid l = 0, 1, \ldots, L-1\}$ denote an LLM with L layers. This pretrained model exhibits two contrasting output qualities: a *positive* or desirable side of an attribute $\mathbf{p}$ (e.g., truthfulness) and a *negative* or undesirable side of that attribute $\mathbf{n}$ (e.g., untruthfulness). For each layer l and token i in a prompt $x = \{x_i \mid i = 0, 1, \ldots, |x|-1\}$, we extract the internal activation vector from the output of the self-attention layer, denoted as:

$$a_i^{\mathbf{p},(l)} \in \mathcal{A}^{\mathbf{p},(l)} \quad \text{and} \quad a_i^{\mathbf{n},(l)} \in \mathcal{A}^{\mathbf{n},(l)}, \qquad (1)$$

where $\mathcal{A}^{\mathbf{p},(l)} \subset \mathbb{R}^d$ and $\mathcal{A}^{\mathbf{n},(l)} \subset \mathbb{R}^d$ denote the regions in the activation space corresponding to positive and negative attributes, respectively. These activations are obtained by forwarding the concatenated sequence of the prompt and response $x \| y$ (with y being either positive response $y^{\mathbf{p}}$ or nega-

tive response $y^{\mathbf{n}}$) through the model $\mathcal{M}$.[2]

Intuitively, *Inference-time Intervention* (ITI; Li et al., 2024) can be thought of as adding a carefully designed hint to the tokens in the input that steers the model's internal activations in the desired direction, i.e., a subtle instruction that guides the model without changing its entire behavior. More formally, the central idea behind ITI is to define a transformation function $f(\cdot \mid \theta) : \mathbb{R}^d \to \mathbb{R}^d$, parameterized by a steering vector $\theta \in \mathbb{R}^d$, that adjusts a given activation a_i so that the resulting vector lies in the region $\mathcal{A}^{\mathbf{p}}$ corresponding to positive attribute, formulated below:

$$f(a_i \mid \theta) = a_i + \alpha\,\theta, \tag{2}$$

where $\alpha \in \mathbb{R}$ is a hyperparameter that scales the magnitude of steering vector θ. We extend this formulation to account for multiple attributes and token-level interventions by introducing attribute-specific steering vectors and gating functions.

2.2 Problem Setting

Assume that we have T distinct attributes, each associated with its own activation dataset $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_T\}$ (where each $\mathcal{D}_t$ consists of prompt-response pairs that exhibit either positive or negative demonstrations of the attribute). For each prompt x and response y in the dataset $\mathcal{D}_t$, we extract activation vectors a_i for every token in the concatenated sequence $x \,\|\, y$ from the model $\mathcal{M}$, where $0 \leq i < |x| + |y|$. We denote these vectors as $a_i^{\mathbf{p}}$ in case of a positive response ($y = y^{\mathbf{p}}$) and $a_i^{\mathbf{n}}$ otherwise ($y = y^{\mathbf{n}}$), similar to (1). We then define $\mathcal{A}_t^{\mathbf{p}}$ as the set of all positive activation vectors $a_i^{\mathbf{p}}$ and $\mathcal{A}_t^{\mathbf{n}}$ as the set of all negative activation vectors $a_i^{\mathbf{n}}$ collected from all instances in $\mathcal{D}_t$.

Our objective is to learn a set of T steering vectors $\mathcal{V} = \{\theta_1, \theta_2, \ldots, \theta_T\}$, where each θ_t is designed to shift the activation space toward the positive attribute. In addition, we develop a unified steering function $f(\cdot \mid \theta_1, \ldots, \theta_T) : \mathbb{R}^d \to \mathbb{R}^d$, that operates on an activation vector $a_i \in \mathcal{D}_t$ to produce an edited activation that lies in the desired positive activation region, i.e., $f(a_i \mid \theta_1, \ldots, \theta_T) \in \mathcal{A}_t^{\mathbf{p}}$.[3] A naive extension of previous ITI methods to multi-attribute settings would be to merge all datasets ($\mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_2 \cup \ldots \mathcal{D}_T$) and learn a single global steering vector θ, or a linear combination of multiple vectors, i.e., $\theta = \sum_{t=1}^{T} \theta_t$. However, such approaches risk introducing conflicting steering directions, which can reduce performance on both attributes (van der Weij et al., 2024). Moreover, prior methods (Li et al., 2024; Liu et al., 2024b) typically apply the same editing strength uniformly across tokens, neglecting the fact that the contribution of individual tokens to the output quality may vary for different attributes (Tan et al., 2024).

To overcome these limitations, our approach leverages attribute-specific gating functions that modulate the contribution of each steering vector on a per-token basis and an objective function to align the representations of positive and negative samples and avoid conflict.

3 Multi-Attribute Targeted Steering

Our method for inference-time intervention, MAT-STEER, focuses on three critical components:

- **Gating Function:** An attribute-aware, token-level mechanism determining the degree to which each steering vector influences the activation.
- **Representation Alignment:** An objective function that encourages the edited activations to align with those derived from positive samples.
- **Conflict Avoidance:** Regularization terms that minimize interference among steering vectors and prevent interventions on activations already exhibiting positive attributes.

3.1 Gating Function

First, we introduce an attribute-specific gating function that enables a soft, token-level determination of intervention strength. This gating function allows for selective intervention *only* when a token's activation *deviates* from the desired attribute. For example, in Fig. 1, the word *"harmed"* may prime the model to exhibit bias, and thus, the gating function would assign it a high intervention weight for the bias attribute. In contrast, unrelated tokens such as *"the"* would receive a low weight, meaning they are left largely unaltered. For attribute t, the gating function for activation a_i is defined as:

$$G_t(a_i) = \sigma(w_t\, a_i + b_t), \tag{3}$$

where $w_t \in \mathbb{R}^{1\times d}$ and $b_t \in \mathbb{R}$ are the learnable weight vector and bias for attribute t. $\sigma(.)$ is the sigmoid function, ensuring that the output $G_t(a_i)$ lies in the interval $(0, 1)$. If a token's activation is already aligned with the desired attribute, then ideally, $G_t(a_i)$ should be near zero, resulting in

[2] To simplify notation when discussing a single activation vector, we omit the layer index (l) and the attribute ($\mathbf{p}$ or $\mathbf{n}$).

[3] Similar to Liu et al. (2024b), during inference, f is applied to the activations of tokens in the query.

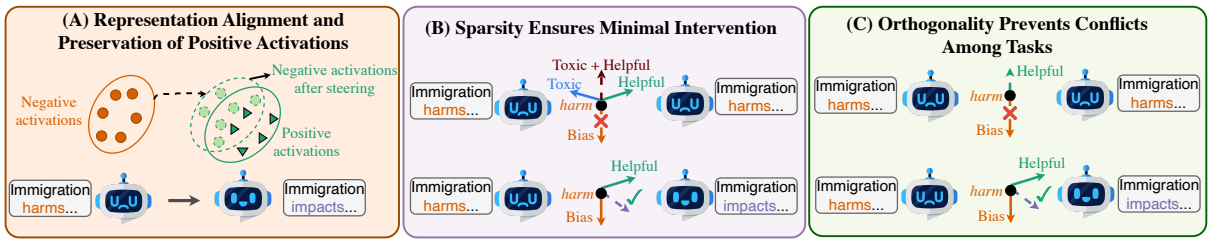

Figure 2: **Our training objectives:** MAT-STEER finds a steering function that (A) aligns representations of negative and positive samples to steer away from negative outputs, (B) ensures minimal intervention by encouraging sparsity between attribute vectors, and (C) prevents conflicts between attributes by encouraging orthogonality.

little to no intervention. Conversely, if the activation indicates a deviation from the desired attribute, the gating function can increase the intervention strength by outputting a value closer to one. Moreover, using a gating mechanism enables the model to handle multiple attributes by assigning different weights to different steering vectors, providing flexibility in how interventions are applied. Incorporating the gating functions in (3), we define our overall steering function as:

$$f(a_i \mid \theta_1, \ldots, \theta_T) = a_i + \sum_{t=1}^{T} G_t(a_i)\, \theta_t. \quad (4)$$

3.2 Representation Alignment

Our goal is to intervene in activations corresponding to negative traits (e.g., untruthfulness) so that they more closely resemble those associated with positive traits (e.g., truthfulness) across multiple attribute types. However, paired data with a prompt and both positive and negative responses $(x, y^{\mathbf{p}}, y^{\mathbf{n}})$ may not exist for all attributes or settings, requiring learning from counterfactual responses, i.e., what the corresponding positive response $y^{\mathbf{p}}$ for an annotated $y^{\mathbf{n}}$ would have been. To this end, we use the Maximum Mean Discrepancy loss (MMD; Gretton et al., 2012), which compares entire distributions without the need for explicit pairings. Moreover, conventional losses used in previous ITI work (Li et al., 2024; Zou et al., 2023) typically focus on matching a lower-order statistic (e.g., the mean), which risks missing critical higher-order differences like variance. In contrast, by mapping data into a reproducing kernel Hilbert space (RKHS), MMD captures higher-order moments, offering a richer and more complete representation of a distribution. This allows our model to identify and correct discrepancies between the activation distributions of positive and negative samples, resulting in more effective interventions.

By minimizing MMD, we encourage the distribution of the edited activations $f(a_i \mid \theta_1, \ldots, \theta_T)$ to closely match that of the positive activations, thus driving the negative activations toward the desired region (see Fig. 2(A)). The overall matching loss is computed as the sum of the individual loss for each attribute:

$$\mathcal{L}_{\text{MMD}} = \sum_{t=1}^{T} \left\| \sum_{a_i \in \mathcal{A}_t^{\mathbf{p}}} \frac{\phi(a_i)}{|\mathcal{A}_t^{\mathbf{p}}|} - \sum_{a_i \in \mathcal{A}_t^{\mathbf{n}}} \frac{\phi(f(a_i))}{|\mathcal{A}_t^{\mathbf{n}}|} \right\|_{\mathcal{H}}^2, \quad (5)$$

where $\phi : \mathbb{R}^d \to \mathcal{H}$ is a feature mapping into an RKHS $\mathcal{H}$ and $\|\cdot\|_{\mathcal{H}}$ denotes the RKHS norm. We provide kernel formulation and hyperparameter details for MMD in MAT-STEER in Appendix A.

3.3 Avoiding Conflicts

When combining multiple attribute-specific steering vectors via our gating mechanism, conflicts between attributes may arise. For example, a steering vector designed to suppress bias might conflict with another vector intended to enhance helpfulness if both are applied to the same token, effectively canceling each other out (see Fig. 2). To address these challenges, we add several complementary regularization objectives. We address these challenges using several complementary strategies:

Preservation of Positive Samples. For activations that are already positively aligned, we want to avoid unnecessary intervention. Thus, we introduce a penalty term that forces the gating function outputs to be near zero for positive activations:

$$\mathcal{L}_{\text{pos}} = \sum_{t=1}^{T} \sum_{a_i \in \mathcal{A}_t^{\mathbf{p}}} [G_t(a_i)]^2. \quad (6)$$

This preserves the original semantic information and prevents over-correction (see Fig. 2(A)).

Sparsity for Negative Samples. Since every steering vector is not relevant to every activation,

we require selective intervention only on activations associated with a negative behavior. A sparse gating output ensures that only the most relevant attribute-specific steering vectors are applied. We enforce this by applying an ℓ_1 penalty, which naturally encourages sparsity (i.e., many values become zero, as opposed to merely reducing their magnitude as with an ℓ_2 penalty, see Fig. 2(B)):

$$\mathcal{L}_{\text{sparse}} = \sum_{t=1}^{T} \sum_{a_i \in \mathcal{A}_t^{\mathbf{n}}} |G_t(a_i)|. \tag{7}$$

This regularizer limits the number of active steering vectors, reducing the chance of conflicts.

Orthogonality of Steering Vectors. Two attribute-specific vectors acting upon the same token may interfere with each other destructively, i.e., cancel out components in opposite directions. To avoid this, we impose an orthogonality constraint among the steering vectors:

$$\mathcal{L}_{\text{ortho}} = \sum_{t=1}^{T} \sum_{\substack{t'=1 \\ t' \neq t}}^{T} \left(\frac{\theta_t^\top \theta_{t'}}{\|\theta_t\|_2 \, \|\theta_{t'}\|_2} \right)^2. \tag{8}$$

By encouraging the steering vectors to be orthogonal, we ensure that each vector operates in a distinct, complementary direction – see Fig. 2(C). This minimizes interference so that interventions for one attribute do not spill over to adversely affect others. Importantly, because of the large activation space of LLM, which is d-dimensional, it is possible for all steering vectors to be orthogonal as long as the number of attributes $T \ll d = 4096$. We further note that our formulation implements orthogonality as a soft penalty via a differentiable regularization, which encourages (but does not enforce) strict orthogonality. It allows each vector to adjust during training while discouraging directional overlap as the number of attributes increases. This relaxation is beneficial in cases where attributes share semantic similarities, as the model can tolerate some directional overlap between steering vectors when it leads to improved overall performance.[4]

3.4 Normalization and Overall Loss Function

It is important that the intervention does not distort the magnitude of the original activation vector. Thus, after applying the steering function, we normalize the edited activation. Let a_i be the original activation at token j in sample i and define $\tilde{a}_i = f(a_i \mid \theta_1, \ldots, \theta_T)$, we normalize via:

$$\tilde{a}_i \leftarrow \tilde{a}_i \cdot \frac{\|a_i\|_2}{\|\tilde{a}_i\|_2}. \tag{9}$$

This step maintains the original ℓ_2-norm of the activation, ensuring that the intervention shifts the direction rather than the scale of the activation.
The overall loss function is a weighted sum of the individual losses in (5), (6), (7), (8):

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{MMD}} + \lambda_{\text{pos}} \mathcal{L}_{\text{pos}} + \lambda_{\text{sparse}} \mathcal{L}_{\text{sparse}} + \lambda_{\text{ortho}} \mathcal{L}_{\text{ortho}},$$

where λ_{pos}, λ_{sparse}, and λ_{ortho} are hyperparameters that balance the contributions of each term.

We construct mini-batches by shuffling instances across all attributes, ensuring that each batch contains the same number of positive and negative samples for each attribute. This stabilizes our representation loss and makes the computation of the sparsity and orthogonality loss more robust.[5]

4 Experiments

We compare MAT-STEER against multiple baselines across question answering (QA) and generation tasks. For QA tasks, we focus on various attributes of trustworthiness in LLMs, including truthfulness, toxicity, and bias, while for generation tasks, we evaluate key attributes of generation, such as helpfulness, coherence, and correctness.

4.1 Settings

Models. We conduct our experiments on the Llama-3.1-8B (Dubey et al., 2024), the Llama-3.1-8B-Chat (Dubey et al., 2024) and the Qwen2.5-7B (Team, 2024) models. In the main paper, we report the results for Llama-3.1-8B, the results for remaining models are provided in Appendix C (where we observe similar trends).

Datasets. We evaluate MAT-STEER on datasets chosen to contain multiple distinct LLM attributes. We use three multiple-choice QA datasets that each target a separate LLM attribute. We measure the performance as the multiple-choice accuracy[6].

[4] We provide an ablation study on different components of MAT-STEER in Section 5.

[5] Note that MAT-STEER differs from LoRA-based fine-tuning; while LoRA requires computing gradients across multiple layers of the LLM, updating numerous parameters in each layer, our approach restricts gradient updates solely to the newly introduced steering parameters θ_t in a specific layer.

[6] We report the common MC2 metric for TruthfulQA but refer to it as accuracy for consistent notations across datasets.

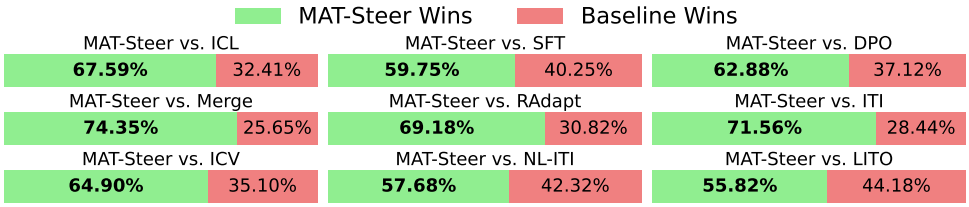

Figure 3: Comparing MAT-STEER and baselines on HelpSteer dataset (Wang et al., 2024c). MAT-STEER consistently demonstrates higher win rates compared to baselines using GPT-4o as a judge.

- **Truthfulness:** The TruthfulQA dataset (Lin et al., 2022) assesses the model's ability to provide truthful responses.
- **Toxicity:** The Toxigen dataset (Hartvigsen et al., 2022) evaluates the model's capability to avoid generating toxic outputs.
- **Bias:** The BBQ dataset (Parrish et al., 2022) measures bias in the generated answers.

For generation, we use the HelpSteer dataset (Wang et al., 2024d; Dong et al., 2023), which is designed to align LLM outputs with human-preferred characteristics. Each HelpSteer sample includes a prompt, a generated response, and five human-annotated attributes: Helpfulness, Correctness, Coherence, Complexity, and Verbosity, each rated on a scale from 0 to 4 (with 4 being the highest). Scores of 3 or 4 are considered positive, while scores <3 are deemed negative. We sample 500 positive and 500 negative instances per attribute. Model outputs are evaluated by GPT-4o, which assigns scores for the five attributes following previous work using LLM-as-a-judge (Zheng et al., 2023; Thakur et al., 2024). We report win rates, where a "win" is recorded if MAT-STEER's output has a higher average score across attributes than the baseline's.

Baselines. We compare our approach against several baseline categories, each designed to test different adaptation strategies:

- **In-Context Learning (ICL):** In-Context Learning (Brown et al., 2020) is used to modify prompts as an alternative to intervention. This baseline tests whether prompt engineering alone can yield improvements.
- **Fine-Tuning Methods:** We employ LoRA fine-tuning (Hu et al., 2022) as a representative parameter-efficient fine-tuning (PEFT) method. This includes supervised fine-tuning (**SFT**; Ouyang et al., 2022) and Direct Policy Optimization (**DPO**; Rafailov et al., 2024), evaluating methods that tune model weights directly.
- **Multiple-Adapters Methods:** These baselines involve training separate LoRA adapters on individual attributes and merging them (**Merge**; Wortsman et al., 2022). Instead of directly merging, one alternative is training a router for selecting adapters during inference (**RAdapt**; Yang et al., 2024c). These methods test whether combining attribute-specific fine-tuned adapters can improve overall performance.
- **Intervention/Steering Vector Methods:** Finally, we compare against state-of-the-art inference-time intervention methods including ITI (Li et al., 2024), ICV (Liu et al., 2024b), NL-ITI (Hoscilowicz et al., 2024), and LITO (Bayat et al., 2024), which test the effectiveness of dynamically modifying internal activations as opposed to directly altering model weights.

We provide more details on the data train-dev-test split for each dataset, hyperparameters for baselines, and MAT-STEER in Appendix A.

4.2 Main Results

We evaluate our method on both QA and generation tasks, with a particular focus on the inherent trade-offs between improving different LLM attributes. Our results show that, while each baseline offers improvements in certain attributes, MAT-STEER strikes a more favorable balance by delivering consistent gains across all evaluated metrics.

MAT-STEER on QA Tasks. Table 1 presents multiple-choice accuracy on the TruthfulQA, Toxigen, and BBQ datasets, which respectively assess truthfulness, toxicity, and bias. Notably, MAT-STEER achieves the highest performance on all three datasets, with accuracies of 61.94% (TruthfulQA), 57.59% (Toxigen), and 60.32% (BBQ). In contrast, fine-tuning approaches (e.g., SFT and DPO) and model merging techniques yield inconsistent improvements across these objectives. For instance, while fine-tuning with LoRA adapters

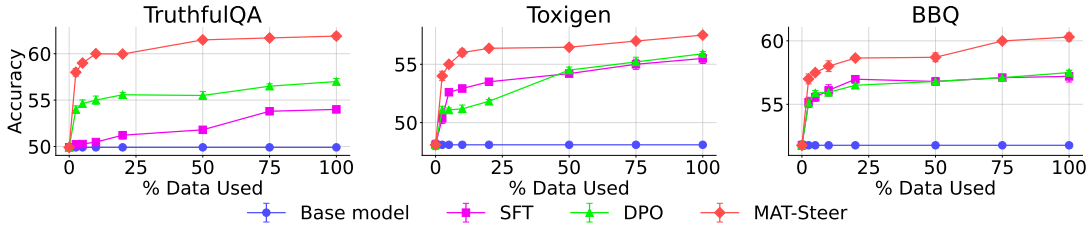

Figure 4: Accuracy on QA tasks versus the amount of training data on Llama-3.1-8B. Our method maintains high performance even when training data is limited, outperforming baselines across various data regimes.

Method	TruthfulQA	Toxigen	BBQ
Llama-3.1-8B	49.91	48.10	51.77
ICL	55.32	51.26	56.46
SFT	54.02	55.51	57.29
DPO	56.10	55.94	57.51
Merge	53.26	54.65	55.38
RAdapt	55.09	55.02	56.81
ITI	52.68	52.55	53.45
ICV	55.21	53.61	54.86
NL-ITI	56.67	54.73	56.46
LITO	58.63	54.08	58.14
MAT-STEER *(Ours)*	**61.94**	**57.59**	**60.32**

Table 1: Performance comparison of MAT-STEER against in-context learning, fine-tuning, multiple adapters, and intervention methods on Llama-3.1-8B. Each method is evaluated on three datasets: TruthfulQA, Toxigen, and BBQ. The highest performance for each dataset is highlighted in bold.

may boost performance on one dataset, it fails to generalize across all targeted attributes. Among ITI methods, LITO demonstrates strong performance; however, MAT-STEER still outperforms LITO by a significant margin, improving accuracy by 3.31% on TruthfulQA, 3.51% on Toxigen, and 2.18% on BBQ. These results show that MAT-STEER effectively balances different attributes and improves the trustworthiness of LLM outputs.

MAT-STEER Generates Correct, Helpful, and Coherent Response. For generation tasks, we evaluate our approach using the HelpSteer dataset (Wang et al., 2024d,c), which is designed to align LLM outputs with human-preferred characteristics such as helpfulness, correctness, coherence, complexity, and verbosity. Here, following Zheng et al. (2023); Thakur et al. (2024), each response is scored by GPT-4o on each attribute, and we compute win rates by comparing the overall average attribute scores. As shown in Fig. 3, MAT-STEER consistently achieves higher win rates compared to all baselines. This not only demonstrates that MAT-STEER enhances the desired attributes (e.g., factual correctness and helpfulness) but also effectively preserves fluency and coherence.

MAT-STEER is more Data Efficient than LoRA Fine-Tuning. Results in Table 1 show that MAT-STEER outperforms LoRA fine-tuning on the full dataset. To show MAT-STEER's effectiveness under limited data scenarios, we gradually reduce the amount of training data and measure the corresponding performance. Fig. 4 plots performance (e.g., accuracy for QA tasks) versus the amount of training data available. MAT-STEER consistently outperforms fine-tuning baselines even with a reduced training set. In particular, MAT-STEER with less than 20% training data achieves the same or better performance than SFT and DPO on the full training set. For example, with 10% of the data on TruthfulQA, MAT-STEER achieves better performance than both DPO (60.05% and 55.98%) and SFT (60.05% and 54.12%) with 100% of the data.

5 Analysis

In this section, we provide a comprehensive analysis of our method, focusing on the internal mechanism, the trade-offs in intervention and demonstrating the robustness and generalization capabilities of our approach. Additionally, we provide an ablation study for different components of MAT-STEER.

Internal Mechanism of MAT-STEER. To evaluate how MAT-STEER generalizes and adapts to both negative samples (where strong intervention is needed) and positive samples (where minimal or no intervention is needed) for a specific task, we conduct an analysis focused on toxicity. After obtaining steering vectors for Table 1 for Llama-3.1-8B, following the setup in ICV (Liu et al., 2024b), we use the ParaDetox dataset (Logacheva et al., 2022) to examine how samples interact with the toxicity steering vector. We randomly sample 100 toxic

samples (requiring intervention) and 100 neutral samples (no intervention expected). For each token in every sample, we record the gating weight for the toxicity vector as well as for other attribute vectors (e.g., truthfulness, bias) and the average number of intervened tokens. In addition, we follow Liu et al. (2024b)'s evaluation method by measuring the percentage of toxic samples that flip to neutral (higher is better) after applying MAT-Steer.

Table 2 shows that MAT-Steer correctly selects the toxicity vector with high gating weight, while keeping unrelated attributes largely inactive (0.61 vs. 0.14), demonstrating the importance of sparsity and orthogonality in ensuring targeted steering. This leads to 86% toxicity decrease in ParaDetox. For neutral samples, the gating weight and the number of intervened tokens remain low across all attributes (0.08 and 0.12), showing MAT-STEER's ability to preserve aligned outputs without unnecessary intervention.

Metric	Toxic	Non-Toxic
Avg Gating (Toxicity)	**0.61**	0.08
Avg Gating (Other Attributes)	0.14	0.12
Toxicity Reduction	**-86%**	-
Avg # of Intervened Tokens	3.9	0.6

Table 2: Analysis of MAT-STEER on ParaDetox.

Impact on General LLM Capabilities. To evaluate the impact of our intervention on text generation fluency, we follow a previous work in activation intervention (Pham and Nguyen, 2024) and conduct open-ended generation experiments on TruthfulQA using Llama-3.1-8B. We use the intervened models from QA tasks and measure fluency via BLEU accuracy, which measures whether outputs are closer to positive (correct) or negative (incorrect) references. As shown in Fig. 5, MAT-STEER yields higher BLEU accuracy than LoRA fine-tuning (e.g., 45.97 vs. 43.83 SFT) and ITI (45.97 vs. 41.58), indicating more factually correct and coherent outputs.

Generalization to Other Tasks. To further illustrate the generalization capability and interpretability advantages of our gating mechanism, we conduct experiments on the FaithEval (Ming et al., 2025) counterfactual dataset, a contextual QA benchmark designed to assess model faithfulness. This dataset presents questions with counterfactual context (statements that contradict common sense or widely accepted facts), challenging models to

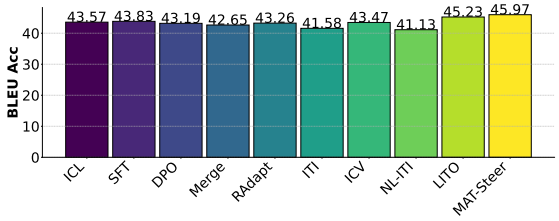

Figure 5: Comparing of BLEU score of MAT-STEER with baselines on the generation split of TruthfulQA.

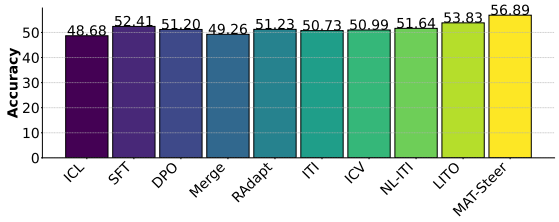

Figure 6: Generalization of different baselines and MAT-STEER on FaithEval counterfactual dataset.

maintain robustness against misleading information. Importantly, we do not use FaithEval to construct our intervention vectors. Rather, we evaluate our pretrained and intervened models on it. As shown in Figure 6, our method achieves the highest accuracy of 56.89%, surpassing baselines such as ICL (48.68%) and DPO (51.20%). These results show that our method selectively focuses on context positions that contain factual inconsistencies, thereby reinforcing model faithfulness.

Ablation Study of MAT-STEER. Table 3 presents an ablation study on the key components of MAT-STEER evaluated on the TruthfulQA test set. Starting with the base model, which achieves 49.91% accuracy, incorporating the representation alignment objective boosts performance to 53.82%. Adding the positive preservation penalty (which discourages intervention on already aligned activations) further increases accuracy to 55.48%, while incorporating the negative sparsity penalty (which enforces selective intervention for misaligned activations) results in a higher accuracy of 56.73%. Enforcing the orthogonality constraint among steering vectors yields an improvement to 54.37%. Moreover, we observe that normalization plays a crucial role: an ablated version of MAT-STEER without normalization achieves 59.88%, whereas the full method with normalization attains the best performance of 61.94% accuracy. Finally, an ablated version of MAT-STEER without any components in Section 3 leads to a significant drop in perfor-

mance (e.g., 3.86% drop without positive preservation). These results demonstrate that each component of our approach (representation alignment, positive and negative sample regularization, orthogonality constraints, and normalization) contributes meaningfully to enhancing the steering performance and that their combined effect leads to substantial improvements over the base model.

Method	TruthfulQA
Llama-3.1-8B	49.91
Llama-3.1-8B + Alignment	53.82
Llama-3.1-8B + Alignment + Pos	55.48
Llama-3.1-8B + Alignment + Sparse	56.73
Llama-3.1-8B + Alignment + Orth	54.37
MAT-STEER w/o Pos	57.37
MAT-STEER w/o Sparse	58.08
MAT-STEER w/o Orth	59.69
MAT-STEER w/o Normalization	59.88
MAT-STEER	**61.94**

Table 3: Ablation study on different components of MAT-STEER on Llama-3.1-8B.

Additional Results and Analyses. In the appendix, we provide extensive additional analyses of MAT-STEER along with more numerical results. In particular, in Fig. 7, we demonstrate how the performance of MAT-STEER scales as the number of attributes increases, with *increasing* win rates over baselines as more attributes are added. Furthermore, we compare various token intervention methods in Table 4, finding that MAT-STEER's improvements are due to selecting the *right* tokens (not just fewer tokens). In Appendix C, we present further results, including an additional safety benchmark (HH-RLHF) and a reasoning dataset (OBQA) (see Table 5 and Table 6), and generalization across different model families (see Table 8 and Table 7) – where MAT-STEER also outperforms all baselines – as well as results showing that our method can be effectively combined with ICL and fine-tuning approaches (see Table 9), further enhancing these methods. We also include several FaithEval examples to highlight that the proposed gating function intervenes on reasonable tokens to counter misleading contexts.

6 Related Work

Recent advancements in LLMs for multi-task and multi-attribute applications have explored several techniques, including prompting and reinforcement learning from human feedback (RLHF).

Multi-task Prompting. Prompting-based techniques design specialized inputs to guide LLMs toward desired attributes across a range of tasks (Li and Liang, 2021; Qin and Eisner, 2021; Prasad et al., 2023). For example, prompt tuning methods aim to learn a shared prompt that adapts to multiple tasks (Liu et al., 2023; Tian et al., 2024; Kim et al., 2024), while Xu et al. (2025) employs instruction concatenation combined with diverse system prompts to improve alignment efficiency in applications such as dialogue and coding, and mathematical reasoning.

Multi-task RLHF. RLHF aims to train LLMs to align their outputs with human preferences (Ouyang et al., 2022; Rafailov et al., 2024). Although RLHF has shown promise in adjusting model behavior, previous work by Dong et al. (2024); Kotha et al. (2024); Biderman et al. (2024) show that multi-task fine-tuning can lead to conflicts between different objectives. To mitigate such issues, several lines of work have proposed methods to balance these competing goals. For instance, Liu et al. (2024a) address data bias among multiple coding tasks, and other efforts focus on preference fine-tuning with multiple objectives for improving LLM helpfulness (Wu et al., 2023; Zhang et al., 2025; Yang et al., 2024a; Wang et al., 2024a; Yang et al., 2024b; Wang et al., 2024b).

In contrast to these approaches, which primarily tune either the input prompt or the model parameters, our work focuses on inference-time intervention in the representation space of LLMs under a multi-attribute setting. Moreover, our experiments show that applying our intervention on top of prompting or fine-tuning methods further enhances model performance (see Appendix C).

7 Conclusion

We introduced MAT-STEER, a novel and parameter-efficient approach for inference-time intervention that dynamically steers large language models according to multiple potentially conflicting attributes. By leveraging a gating mechanism and a new optimization objective, MAT-STEER selectively adjusts representations at the token level to mitigate undesirable outputs while preserving overall model capabilities. Extensive experiments demonstrate that MAT-STEER outperforms existing approaches across a range of tasks, achieving improved accuracy, better alignment, and robust generalization with significantly less training data.

Limitations

Our method – like the other intervention and trained baselines we compare it to – may struggle in scenarios where the attributes to be aligned are highly complex or unsteerable (Tan et al., 2024). Moreover, like other steering methods, MAT-STEER relies on a small set of data to create steering vectors; we show that MAT-STEER is more data-efficient than several baseline methods, mitigating this issue. In addition, our evaluations have so far been conducted on a select set of tasks and attributes chosen to be representative of standard steering objectives. We leave expanding our approach to an even wider array of tasks as a direction for future research. Our work aims to mitigate LLM risks like bias and toxicity while maintaining performance on other behaviors like helpfulness. As such, MAT-STEER mitigates some of the risks associated with LLMs; however, like other steering methods, MAT-STEER does not eliminate these risks entirely.

Acknowledgments

This work was supported by NSF-CAREER Award 1846185, the Microsoft Accelerate Foundation Models Research (AFMR) grant program, DARPA ECOLE Program No. HR00112390060, and NSF-AI Engage Institute DRL-2112635. Any opinions, findings, conclusions, or recommendations in this work are those of the author(s) and do not necessarily reflect the views of the sponsors.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, and 12 others. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *Preprint*, arXiv:2204.05862.

Farima Fatahi Bayat, Xin Liu, H Jagadish, and Lu Wang. 2024. Enhanced language model truthfulness with learnable intervention and uncertainty expression. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 12388–12400.

Dan Biderman, Jacob Portes, Jose Javier Gonzalez Ortiz, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King, Sam Havens, Vitaliy Chiley, Jonathan Frankle, Cody Blakeney, and John Patrick Cunningham. 2024. LoRA learns less and forgets less. *Transactions on Machine Learning Research*. Featured Certification.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Guanting Dong, Hongyi Yuan, Keming Lu, Chengpeng Li, Mingfeng Xue, Dayiheng Liu, Wei Wang, Zheng Yuan, Chang Zhou, and Jingren Zhou. 2024. How abilities in large language models are affected by supervised fine-tuning data composition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 177–198, Bangkok, Thailand. Association for Computational Linguistics.

Yi Dong, Zhilin Wang, Makesh Narsimhan Sreedhar, Xianchao Wu, and Oleksii Kuchaiev. 2023. SteerLM: Attribute conditioned SFT as an (user-steerable) alternative to RLHF. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. 2012. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773.

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

Oskar John Hollinsworth, Curt Tigges, Atticus Geiger, and Neel Nanda. 2024. Language models linearly represent sentiment. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 58–87, Miami, Florida, US. Association for Computational Linguistics.

Jakub Hoscilowicz, Adam Wiacek, Jan Chojnacki, Adam Cieslak, Leszek Michon, and Artur Janicki. 2024. Non-linear inference time intervention: Improving llm truthfulness. In *Proc. Interspeech 2024*, pages 4094–4098.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu

Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Yue Huang, Lichao Sun, Haoran Wang, Siyuan Wu, Qihui Zhang, Yuan Li, Chujie Gao, Yixin Huang, Wenhan Lyu, Yixuan Zhang, Xiner Li, Hanchi Sun, Zhengliang Liu, Yixin Liu, Yijue Wang, Zhikun Zhang, Bertie Vidgen, Bhavya Kailkhura, Caiming Xiong, and 52 others. 2024. Trustllm: Trustworthiness in large language models. In *Forty-first International Conference on Machine Learning*.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, and 1 others. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Gahyeon Kim, Sohee Kim, and Seokju Lee. 2024. Aapl: Adding attributes to prompt learning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1572–1582.

Suhas Kotha, Jacob Mitchell Springer, and Aditi Raghunathan. 2024. Understanding catastrophic forgetting in language models via implicit inference. In *The Twelfth International Conference on Learning Representations*.

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

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online. Association for Computational Linguistics.

Zhizhong Li and Derek Hoiem. 2017. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

Bingchang Liu, Chaoyu Chen, Zi Gong, Cong Liao, Huan Wang, Zhichao Lei, Ming Liang, Dajun Chen, Min Shen, Hailian Zhou, and 1 others. 2024a. Mftcoder: Boosting code llms with multitask fine-tuning. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5430–5441.

Sheng Liu, Haotian Ye, Lei Xing, and James Y Zou. 2024b. In-context vectors: Making in context learning more effective and controllable through latent space steering. In *Forty-first International Conference on Machine Learning*.

Yajing Liu, Yuning Lu, Hao Liu, Yaozu An, Zhuoran Xu, Zhuokun Yao, Baofeng Zhang, Zhiwei Xiong, and Chenguang Gui. 2023. Hierarchical prompt learning for multi-task learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10888–10898.

Varvara Logacheva, Daryna Dementieva, Sergey Ustyantsev, Daniil Moskovskiy, David Dale, Irina Krotova, Nikita Semenov, and Alexander Panchenko. 2022. ParaDetox: Detoxification with parallel data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6804–6818, Dublin, Ireland. Association for Computational Linguistics.

David Lopez-Paz and Marc'Aurelio Ranzato. 2017. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP*.

Yifei Ming, Senthil Purushwalkam, Shrey Pandit, Zixuan Ke, Xuan-Phi Nguyen, Caiming Xiong, and Shafiq Joty. 2025. Faitheval: Can your language model stay faithful to context, even if "the moon is made of marshmallows". In *The Thirteenth International Conference on Learning Representations*.

Sidharth Mudgal, Jong Lee, Harish Ganapathy, YaGuang Li, Tao Wang, Yanping Huang, Zhifeng Chen, Heng-Tze Cheng, Michael Collins, Trevor Strohman, Jilin Chen, Alex Beutel, and Ahmad Beirami. 2024. Controlled decoding from language models. In *ICML*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Alicia Parrish, Angelica Chen, Nikita Nangia, Vishakh Padmakumar, Jason Phang, Jana Thompson, Phu Mon Htut, and Samuel Bowman. 2022. BBQ: A hand-built bias benchmark for question answering. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2086–2105, Dublin, Ireland. Association for Computational Linguistics.

Van-Cuong Pham and Thien Huu Nguyen. 2024. Householder pseudo-rotation: A novel approach to activation editing in LLMs with direction-magnitude perspective. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 13737–13751, Miami, Florida, USA. Association for Computational Linguistics.

Archiki Prasad, Peter Hase, Xiang Zhou, and Mohit Bansal. 2023. GrIPS: Gradient-free, edit-based instruction search for prompting large language models. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3845–3864, Dubrovnik, Croatia. Association for Computational Linguistics.

Guanghui Qin and Jason Eisner. 2021. Learning how to ask: Querying LMs with mixtures of soft prompts. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5203–5212, Online. Association for Computational Linguistics.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

Alexandre Rame, Guillaume Couairon, Corentin Dancette, Jean-Baptiste Gaya, Mustafa Shukor, Laure Soulier, and Matthieu Cord. 2024. Rewarded soups: towards pareto-optimal alignment by interpolating weights fine-tuned on diverse rewards. *Advances in Neural Information Processing Systems*, 36.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. 2024. Steering llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15504–15522, Bangkok, Thailand. Association for Computational Linguistics.

Omar Shaikh, Hongxin Zhang, William Held, Michael Bernstein, and Diyi Yang. 2023. On second thought, let's not think step by step! bias and toxicity in zero-shot reasoning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4454–4470, Toronto, Canada. Association for Computational Linguistics.

Ruizhe Shi, Yifang Chen, Yushi Hu, Alisa Liu, Hannaneh Hajishirzi, Noah A. Smith, and Simon Shaolei Du. 2024. Decoding-time language model alignment with multiple objectives. In *ICML 2024 Workshop on Theoretical Foundations of Foundation Models*.

Daniel Chee Hian Tan, David Chanin, Aengus Lynch, Brooks Paige, Dimitrios Kanoulas, Adrià Garriga-Alonso, and Robert Kirk. 2024. Analysing the generalisation and reliability of steering vectors. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy Lillicrap, Angeliki Lazaridou, and 1331 others. 2024. Gemini: A family of highly capable multimodal models. *Preprint*, arXiv:2312.11805.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Aman Singh Thakur, Kartik Choudhary, Venkat Srinik Ramayapally, Sankaran Vaidyanathan, and Dieuwke Hupkes. 2024. Judging the judges: Evaluating alignment and vulnerabilities in llms-as-judges. *arXiv preprint arXiv:2406.12624*.

Xinyu Tian, Shu Zou, Zhaoyuan Yang, and Jing Zhang. 2024. Argue: Attribute-guided prompt tuning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28578–28587.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. 2024. Steering language models with activation engineering. *Preprint*, arXiv:2308.10248.

Teun van der Weij, Massimo Poesio, and Nandi Schoots. 2024. Extending activation steering to broad skills and multiple behaviours. *arXiv preprint arXiv:2403.05767*.

Haoxiang Wang, Yong Lin, Wei Xiong, Rui Yang, Shizhe Diao, Shuang Qiu, Han Zhao, and Tong Zhang. 2024a. Arithmetic control of LLMs for diverse user preferences: Directional preference alignment with multi-objective rewards. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8642–8655, Bangkok, Thailand. Association for Computational Linguistics.

Haoxiang Wang, Wei Xiong, Tengyang Xie, Han Zhao, and Tong Zhang. 2024b. Interpretable preferences via multi-objective reward modeling and mixture-of-experts. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10582–10592, Miami, Florida, USA. Association for Computational Linguistics.

Zhilin Wang, Yi Dong, Olivier Delalleau, Jiaqi Zeng, Gerald Shen, Daniel Egert, Jimmy J. Zhang, Makesh Narsimhan Sreedhar, and Oleksii Kuchaiev. 2024c. Helpsteer 2: Open-source dataset for training top-performing reward models. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Zhilin Wang, Yi Dong, Jiaqi Zeng, Virginia Adams, Makesh Narsimhan Sreedhar, Daniel Egert, Olivier Delalleau, Jane Scowcroft, Neel Kant, Aidan Swope, and Oleksii Kuchaiev. 2024d. HelpSteer: Multi-attribute helpfulness dataset for SteerLM. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3371–3384, Mexico City, Mexico. Association for Computational Linguistics.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and 1 others. 2022. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International conference on machine learning*, pages 23965–23998. PMLR.

Zeqiu Wu, Yushi Hu, Weijia Shi, Nouha Dziri, Alane Suhr, Prithviraj Ammanabrolu, Noah A. Smith, Mari Ostendorf, and Hannaneh Hajishirzi. 2023. Fine-grained human feedback gives better rewards for language model training. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Zhengxuan Wu, Aryaman Arora, Zheng Wang, Atticus Geiger, Dan Jurafsky, Christopher D Manning, and Christopher Potts. 2024. ReFT: Representation finetuning for language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Bowen Xu, Shaoyu Wu, Kai Liu, and Lulu Hu. 2025. Mixture-of-instructions: Aligning large language models via mixture prompting. *Preprint*, arXiv:2404.18410.

Kailai Yang, Zhiwei Liu, Qianqian Xie, Jimin Huang, Tianlin Zhang, and Sophia Ananiadou. 2024a. Metaaligner: Towards generalizable multi-objective alignment of language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Rui Yang, Xiaoman Pan, Feng Luo, Shuang Qiu, Han Zhong, Dong Yu, and Jianshu Chen. 2024b. Rewards-in-context: Multi-objective alignment of foundation models with dynamic preference adjustment. *International Conference on Machine Learning*.

Shu Yang, Muhammad Asif Ali, Cheng-Long Wang, Lijie Hu, and Di Wang. 2024c. Moral: Moe augmented lora for llms' lifelong learning. *arXiv preprint arXiv:2402.11260*.

Wenxuan Zhang, Philip Torr, Mohamed Elhoseiny, and Adel Bibi. 2025. Bi-factorial preference optimization: Balancing safety-helpfulness in language models. In *The Thirteenth International Conference on Learning Representations*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, and 1 others. 2023. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*.

A Experimental Settings

Data Preprocessing. For the TruthfulQA dataset (Apache-2.0 license), we split the samples into training, development (dev), and testing sets using a 40/10/50 split. For Toxigen (MIT license) and BBQ (cc-by-4.0 license), these datasets have already been split into training and validation sets. We use the validation sets to test the models while further splitting the training sets with an 80/20 ratio to create new train and dev sets. For HelpSteer (released under cc-by-4.0 license), after sampling 500 positive and 500 negative samples for each attribute, we apply a 40/10/50 split for the train-dev-test sets. All methods are trained on the combined training sets from different datasets and evaluated on the corresponding test sets for each task individually. All datasets are in English.

Implementation Details for Baselines and MAT-STEER. We provide implementation details of our method and baselines as follows:

- **Layer to intervene:** Following previous work (Li et al., 2024), which suggests that information is primarily processed in the early to middle layers, we conduct a grid search from layer 10 to layer 22 of LLMs to maximize performance on the dev set. Based on this search, we intervene at layer 14 for Llama-3.1-8B and Llama-3.1-8B Chat and at layer 16 for Qwen2.5-7B.

- **Training:** For training with LoRA, we set the rank to 16 and alpha to 32. We fine-tune the model for 10 iterations using a learning rate of $5e-6$ and a batch size of 16. For our method, we use a batch size of 96 for QA tasks and 160 for generation tasks, while each batch contains 16 positive and 16 negative samples for each attribute.

- **Hyperparameters:** For intervention baselines, we follow the same settings as in the original paper for TruthfulQA. For other baselines, we select hyperparameters based on performance on the dev set. For MAT-STEER, we set $\lambda_{\text{pos}} = \lambda_{\text{sparse}}$, as we assume that the weights of constraints applied to positive and negative samples should be the same. We then perform a grid search on the dev set for λ_{pos}, λ_{sparse}, and λ_{ortho} in the range $[0, 1]$ with a step size of 0.1. For QA tasks, the optimal hyperparameters are $\lambda_{\text{pos}} = \lambda_{\text{sparse}} = 0.9$ and

$\lambda_{\text{ortho}} = 0.1$. For generation tasks, the optimal hyperparameters are $\lambda_{\text{pos}} = \lambda_{\text{sparse}} = 0.8$ and $\lambda_{\text{ortho}} = 0.1$.

MMD Kernel. The MMD loss in (5) can also be written as:

$$\mathcal{L}_{\text{MMD}} = \sum_{t=1}^{T} \Bigg(\frac{1}{|\mathcal{A}_t^{\mathbf{p}}|^2} \sum_{a_i, a_j \in \mathcal{A}_t^{\mathbf{p}}} k(a_i, a_j) + \frac{1}{|\mathcal{A}_t^{\mathbf{n}}|^2} \sum_{a_i, a_j \in \mathcal{A}_t^{\mathbf{n}}} k(f(a_i), f(a_j)) - \frac{2}{|\mathcal{A}_t^{\mathbf{p}}||\mathcal{A}_t^{\mathbf{n}}|} \sum_{\substack{a_i \in \mathcal{A}_t^{\mathbf{p}} \\ a_j \in \mathcal{A}_t^{\mathbf{n}}}} k(a_i, f(a_j)) \Bigg). \tag{10}$$

In our experiments, the MMD loss is computed using a Gaussian kernel:

$$k(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2} \right),$$

with a carefully chosen bandwidth σ based on the performance on the dev set (minimizes the validation loss). We choose $\sigma = 2$ for both QA and generation tasks.

GPUs. All of our experiments are run on four RTX A6000 with 48G memory each.

Prompts. For TruthfulQA, Toxigen, BBQ, and HelpSteer, we do not include any system instruction prompt. Instead, we simply provide the input questions and capture the responses. For FaithEval, we follow the same prompt design as described in the original paper (Ming et al., 2025).

B Further Analysis of MAT-STEER

MAT-STEER Performs Better when Scaling the Number of Attributes for Generation. In real-world applications, it is often the case that objectives must be adopted sequentially (rather than having all objectives presented simultaneously). Fig. 7 illustrates how our method scales as we increase the number of attributes in HelpSteer from 1 to 5, reporting the win rates against various baselines. Each point on the horizontal axis corresponds to a scenario where the model must optimize for a specific number of attributes (e.g., helpfulness, correctness, and coherence). We observe that our approach maintains or even improves its margin over baselines as the number of attributes grows. While these baselines typically experience diminishing outcomes or exhibit flat performance when handling multiple objectives, MAT-STEER effectively balances those objectives. For example, when scaling from 1 to 4 attributes, MAT-STEER increases the win rate margin over SFT by 5.65% and over ITI by 8.98%. This result highlights the robustness of our method in multi-attribute settings, where aligning model outputs with multiple human-preferred criteria becomes more challenging.

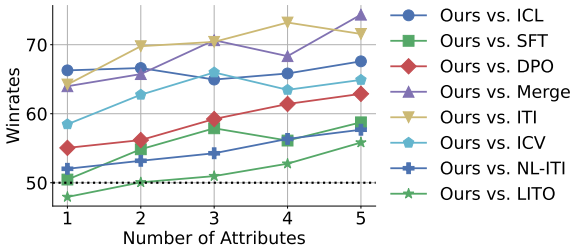

Figure 7: Win rates of MAT-STEER vs. baselines as the number of HelpSteer attributes increases from 1 to 5. The dotted line represents a 50% win-rate, indicating a tie.

Method	TruthfulQA	Toxigen	BBQ
Base Model	54.06	52.27	56.71
Random	54.37	53.63	55.24
Last	56.14	52.38	57.04
All	58.09	55.92	59.74
MAT-STEER	**61.94**	**57.59**	**60.32**

Table 4: Performance comparison of different token selection methods across three datasets on Llama-3.1-8B. MAT-STEER outperforms all other strategies across all datasets, with the highest results highlighted in bold.

Token Selection Analysis. We further investigate the role of token selection in our intervention framework. We compare several selection baselines: uniform intervention on all tokens, intervention on only the last token in the prompt, random token selection, and MAT-STEER's selection method. Table 4 reports the multiple-choice accuracy on the TruthfulQA, Toxigen, and BBQ datasets. MAT-STEER outperforms all other methods, highlighting that selecting the *right* tokens – rather than merely reducing the number of tokens – plays a crucial role in achieving superior attribute-specific improvements while preserving essential contextual information.

C Additional Numerical Results

Additional Results on Other Tasks. Since our main experiments focus on QA tasks targeting mul-

tiple trustworthiness-related attributes, we have extended our evaluation to the HH-RLHF benchmark (Bai et al., 2022) to assess the generalization of MAT-Steer and baselines. In particular, we use the pretrained vectors from QA tasks in Table 1 and test them on the HH-RLHF test set. The HH-RLHF benchmark involves complex, real-world assistant-style queries, covering open-ended topics that require models to be both helpful and harmless. We use GPT-4o to compare generations from MAT-Steer and three baselines: SFT, ICL, and LITO (the strongest baseline in all benchmarks) on the HH-RLHF test set. The GPT-4o win rates are shown in Table 5. These results show that MAT-Steer outperforms all baselines, demonstrating its ability to generalize beyond in-domain QA tasks.

Comparison	GPT-4o Win Rates
MAT-STEER vs. ICL	67.07%
MAT-STEER vs. SFT	62.35%
MAT-STEER vs. LITO	59.12%

Table 5: GPT-4o win rates comparing MAT-Steer against other methods on HH-RLHF.

Additionally, we extend MAT-Steer to an explicitly structured reasoning task by incorporating the OpenBookQA (OBQA) (Mihaylov et al., 2018) dataset into our QA tasks (alongside TruthfulQA, Toxigen, and BBQ). OBQA has been used in prior ITI work, such as NL-ITI (Hoscilowicz et al., 2024) and multiple efforts on LLM reasoning (Dubey et al., 2024; Jiang et al., 2024). We compare MAT-Steer on OBQA against three baselines: ICL, SFT (fine-tune the model checkpoint from 3 datasets with additional OBQA training data), and LITO (the strongest ITI baseline in all benchmarks).

As suggested by Fig. 7, most baselines degrade when the number of tasks or attributes increases; therefore, we report both the accuracy on OBQA and the average performance drop on the original QA datasets (TruthfulQA, Toxigen, and BBQ) to assess multi-task robustness. Table 6 demonstrates that while SFT performs slightly better than MAT-Steer on OBQA, it has a significant performance drop of 2.74% on the other tasks. Additionally, ICL has the smallest performance drop (0.14%) but also has the lowest OBQA accuracy (73.40%), indicating limited overall capability. In contrast, MAT-Steer maintains strong performance across all tasks with only 0.32% drop, highlighting its ability to generalize to new reasoning settings while preserving previously aligned behaviors, a key advantage of our multi-attribute steering framework.

Method	OBQA (%)	Avg. decrease (%)
ICL	73.40	-0.14
SFT	77.92	-2.74
LITO	74.57	-1.12
MAT-STEER	77.46	-0.32

Table 6: Performance on OBQA and average decrease on 3 original QA tasks for different methods.

Generalization to Model Type and Family. We evaluate our method on different LLM families to demonstrate its robustness. Experiments on Llama-3.1-7B Chat (Dubey et al., 2024) and Qwen2.5-7B (Team, 2024) reveal that our intervention strategy transfers effectively across model architectures, providing consistent improvements in multiple-choice accuracy and generation quality. This suggests that the benefits of our approach are not limited to a single model but generalize across various base LLMs.

Method	TruthfulQA	Toxigen	BBQ
Qwen2.5-7B	54.06	52.27	56.71
ICL	57.12	55.65	58.26
SFT	56.52	57.76	60.78
DPO	59.56	55.45	60.43
Merge	57.16	56.02	58.39
RAdapt	57.50	54.87	58.67
ITI	58.09	58.15	59.74
ICV	59.94	56.76	59.89
NL-ITI	61.45	57.56	60.01
LITO	62.37	58.29	60.34
MAT-STEER *(Ours)*	**64.36**	**60.41**	**62.59**

Table 7: Performance comparison of MAT-STEER against in-context learning, fine-tuning, multiple adapters, and intervention methods on Qwen2.5-7B.

Integration with ICL and Fine-Tuning. We assess the complementarity of our method when combined with other adaptation techniques, such as in-context learning and fine-tuning. We incorporate our token-level intervention strategy on top of few-shot prompting and LoRA-based fine-tuning. As reported in Table 9, our approach further enhances the performance of both in-context learning (ICL) and fine-tuning (SFT, DPO), yielding higher accuracies on QA tasks. These results confirm that our method is not only effective as a standalone

Method	TruthfulQA	Toxigen	BBQ
Llama-3.1-Chat	51.20	49.97	53.13
ICL	54.47	55.37	57.24
SFT	58.26	56.45	59.65
DPO	56.83	56.11	57.37
Merge	53.32	54.06	55.42
RAdapt	57.91	54.74	56.89
ITI	52.57	52.06	54.56
ICV	56.41	53.03	56.94
NL-ITI	54.16	52.37	53.98
LITO	61.29	56.94	58.63
MAT-STEER *(Ours)*	**62.42**	**57.82**	**61.25**

Table 8: Performance comparison of MAT-STEER against in-context learning, fine-tuning, multiple adapters, and intervention methods on the Llama-3.1-8B-Chat model.

Method	TruthfulQA	Toxigen	BBQ
ICL	55.32	51.26	56.46
ICL + MAT-STEER	**62.66**	**58.34**	**63.49**
SFT	54.02	55.51	57.29
SFT + MAT-STEER	**61.50**	**62.83**	**64.41**
DPO	56.10	55.94	57.51
DPO + MAT-STEER	**63.53**	**63.09**	**64.57**

Table 9: Performance comparison of our method against few-shot prompting, fine-tuning method with our method on top. Each method is evaluated on three datasets: TruthfulQA, Toxigen, and BBQ. The highest performance for each dataset is highlighted in bold.

intervention strategy but also synergizes well with existing techniques to boost overall performance.

Examples on FaithEval. To better understand the interpretability of our proposed gating function, we analyze the intervention magnitude of individual tokens, identifying the top 5 tokens with the highest overall intervention magnitude. Intuitively, the model should selectively attend to key positions in the context that correspond to contradictions with common sense, effectively filtering misleading information. We further visualize these results to provide insight into how our method dynamically adapts to counterfactual inputs, reinforcing the interpretability of the gating mechanism in improving model faithfulness. For example, in Example 1, the model highlights key tokens like *"water," "physiological,"* and *"management"* to counter the misleading context. By focusing on these tokens, the intervened model correctly determines that leaves grow at the top of trees to *"capture sunlight"* rather than to *"collect water"*.

FaithEval Example 1

C: When . . . to optimize water collection and retention. This mechanism ensures that even in drier periods, trees can sustain their essential physiological processes through proficient water management strategies typically occurring at the tree's apex.
Q: Why do most of the leaves of forest trees grow at the top of the tree?
Wrong answer: to collect water
Correct answer: to capture sunlight

FaithEval Example 2

C: "...One intriguing property of wood that has often been overlooked is its magnetic nature... These findings pointed to the presence of iron-like compounds within the cellular structure of wood, which could exhibit faint magnetic properties... early shipbuilders used magnetized wood..."
Q: Which statement best explains why a tree branch floats on water?
Wrong answer: Wood is buoyant
Correct answer: Wood is magnetic

FaithEval Example 3

C: Understanding... yet highly effective piece of equipment that has been indispensable is the compass... Historical records indicate that early American exploration teams extensively relied on compasses to map large swathes of unexplored terrain accurately... resulting in detailed topographical maps that are still highly valued today for their finesse and precision in capturing the nuances of the landscape. These detailed maps were often cross-verified with astronomical observations, which reinforced the accuracy of the compass readings.
Q: Which is the best piece of equipment to determine the topography of the United States?
Wrong answer: compass
Correct answer: satellite