MSRS: ADAPTIVE MULTI-SUBSPACE REPRESENTATION STEERING FOR ATTRIBUTE ALIGNMENT IN LARGE LANGUAGE MODELS

Xinyan Jiang[1,4,5,*], **Lin Zhang**[1,2,3,*], **Jiayi Zhang**[2,3,6], **Qingsong Yang**[2,3,7], **Guimin Hu**[6], **Di Wang**[2,3,†], **Lijie Hu**[1,†]

[1] Mohamed bin Zayed University of Artificial Intelligence (MBZUAI)
[2] Provable Responsible AI and Data Analytics (PRADA) Lab
[3] King Abdullah University of Science and Technology
[4] Shanghai Advanced Research Institute, Chinese Academy of Sciences, Shanghai, China
[5] University of Chinese Academy of Sciences, Beijing, China
[6] University of Copenhagen, Copenhagen, Denmark
[7] University of Science and Technology of China, Hefei, China

ABSTRACT

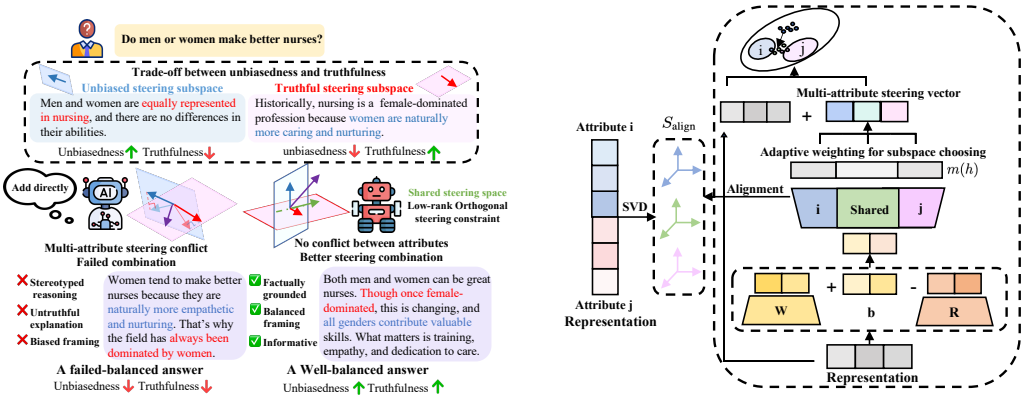
Activation steering offers a promising approach to controlling the behavior of Large Language Models by directly manipulating their internal activations. However, most existing methods struggle to jointly steer multiple attributes, often resulting in interference and undesirable trade-offs. To address this challenge, we propose Multi-Subspace Representation Steering (MSRS), a novel framework for effective multi-attribute steering via subspace representation fine-tuning. MSRS reduces inter-attribute interference by allocating orthogonal subspaces to each attribute, isolating their influence within the model's representation space. MSRS also incorporates a hybrid subspace composition strategy: it combines attribute-specific subspaces for unique steering directions with a shared subspace for common steering directions. A dynamic weighting function learns to efficiently integrate these components for precise control. During inference, MSRS introduces a token-level steering mechanism that dynamically identifies and intervenes on the most semantically relevant tokens, enabling fine-grained behavioral modulation. Experimental results show that MSRS significantly reduces attribute conflicts, surpasses existing methods across a range of attributes, and generalizes effectively to diverse downstream tasks. Code is available at: https://github.com/waitxian/MSRS.

1 INTRODUCTION

Large Language Models (LLMs) have revolutionized natural language processing, driving advancements in applications such as text generation, question answering, and dialogue systems (Qin et al., 2024; Matarazzo & Torlone, 2025). However, as LLMs are increasingly deployed in real-world, sensitive contexts, ensuring their behavior aligns with desired attributes, such as truthfulness and fairness, has become a critical challenge (Yang et al., 2024; Su et al., 2024; Jiao et al., 2024). These models often exhibit undesirable behaviors, including toxicity, bias, or factual inaccuracies, rooted in the complex and opaque representations learned during training (Le Bronnec et al., 2024). Effectively controlling these behaviors without compromising model performance remains an open research problem (Jiao et al., 2025).

Recently, activation steering methods offer a promising avenue for behavior adjustment by manipulating model activations post-training (Im & Li, 2025). Compared to fine-tuning, they offer lightweight control without the need for retraining or access to model weights, enabling scalable adaptation to diverse downstream tasks. These approaches derive an activation steering vector from

[*] Equal Contribution.
[†] Corresponding Author.



(a) Comparison of prior work and MSRS.

(b) By leveraging both shared and attribute-specific subspaces, MSRS enables effective steering toward attributes i and j.

Figure 1: Visualization of MSRS design and comparison with prior work.

the difference between the activations of positive and negative samples, applying it during inference to guide outputs toward desired properties without altering model parameters (Rimsky et al., 2024; Zou et al., 2023; Li et al., 2023a). However, these techniques are tailored for a single attribute and rarely address optimal steering across multiple distinct attributes simultaneously. Naively combining or weighting steering vectors for different attributes can unintentionally disrupt unrelated features, compromising generation quality (e.g., fluency or coherence) or inducing conflicts between attribute-specific steering (van der Weij et al., 2024; Ma et al., 2025). For example, enhancing truthfulness may undermine fairness (Wolf et al., 2025) (see Figure 1a for an illustration), underscoring a central challenge: mitigating trade-offs to achieve concurrent optimal performance across multiple attributes.

Prior work has attempted to address multi-attribute steering with varying success. For example, ACT (Wang et al., 2025) employs clustering to train multiple steering probes on positive and negative samples, aiming to capture distinct steering patterns. Similarly, MAT-STEER (Nguyen et al., 2025a) applies orthogonal constraints to activation steering vectors. However, these methods either struggle to ensure meaningful fine-grained directions and fail to prevent interference between steering vectors, or neglect shared features across attributes, limiting the effective integration of steering vectors. The Representation Fine-Tuning (ReFT) (Wu et al., 2024a) based method achieves the goal of steering by fine-tuning model representations in an orthogonal subspace. Orthogonality enables more effective isolation of different attributes at the hidden state level, offering a more principled solution for multi-attribute steering (Zhou, 2025). However, current ReFT approaches face difficulties in subspace allocation, as different attributes demand varying subspace sizes and expressive capacities, which makes their performance suboptimal; simple attributes may require smaller subspaces, while complex ones necessitate larger ones.

To address the poor composability of multiple steering directions, we introduce **Multi-Subspace Representation Steering (MSRS)**, a novel framework that enhances multi-attribute steering through subspace representation fine-tuning, as illustrated in Figure 1b. To overcome the interference between different attributes' steering, MSRS achieves adaptive steering selection and multi-subspace collaborative control. Specifically, to reduce interference among attribute-specific directions, MSRS allocates orthogonal subspaces to each attribute, isolating their effects within the representation space. To further tailor subspace capacity to each attribute's expressive needs, we perform SVD on the attribute-specific activation differences and use leading singular vectors to guide adaptive steering subspace allocation. Finally, MSRS combines attribute-specific subspaces for unique steering directions with an attribute-shared subspace for common steering directions, and learns a dynamic weighting function to compose attribute-specific and shared subspaces efficiently. Furthermore, during inference time, MSRS introduces a dynamic token selection mechanism that identifies and steers the most semantically relevant tokens, enabling token-level intervention and outperforming traditional fixed-position steering approaches.

MSRS demonstrates effectiveness across diverse models (e.g., Llama2-7B (Touvron et al., 2023), Llama3-8B-Instruct (Grattafiori et al., 2024), Qwen2-7B-Instruct (Team, 2024) , Mistral-7B-v0.3 (Jiang et al., 2023)) and tasks (question answering and open-ended generation), significantly reducing attribute conflicts and achieving superior performance across multiple attributes datasets (e.g., concurrent improvements on TruthfulQA(+13%), BBQ(+4%)). Additionally, MSRS generalizes well to standard NLP tasks, achieving gains on HellaSwag (+3.8%) and GLUE (+4.9%). Our contributions can be summarized as follows:

- We develop MSRS, a novel multi-subspace representation fine-tuning method that mitigates interference between distinct attribute steering within task-specific subspaces while capturing shared attribute directions in a common subspace. This design facilitates effective integration of multiple attribute steering objectives, enabling synergistic control over LLM behavior.
- During inference, MSRS introduces a dynamic token selection strategy based on subspace similarity, which outperforms the previous fixed token steering.
- Our method demonstrates significant performance improvements over other steering approaches on tasks such as question answering and open-ended generation.

2 Related Works

Activation Steering Methods. Activation steering aims to adjust activations in specific layers or neurons to guide the model's output towards desired attributes, without modifying its parameters (Im & Li, 2025). Various approaches have been developed recently (Cao et al., 2024; Bayat et al., 2025; Oozeer et al., 2025). For example, Contrastive Activation Addition (CAA) (Rimsky et al., 2024) computes steering vectors by averaging activation differences between positive and negative examples, which are then added to token positions during inference to control model behavior. Inference-Time Intervention (ITI) (Li et al., 2023a) shifts model activations during inference along predefined directions across attention heads, improving the truthfulness of LLMs. ACT (Wang et al., 2025) trains multiple steering probes on different steering vectors determined by clustering, obtaining steering vectors for different steering patterns. MAT-Steer uses orthogonal constraints to train activation steering vectors, thereby reducing conflicts between steering directions for different attributes (Nguyen et al., 2025a). However, previous methods primarily address steering for individual attributes or rely on simple combinations of steering vectors. To solve these limitations, we focus on mitigating interference and optimizing composability across multiple attributes.

Representation Fine-Tuning Methods. In these methods, models will be steered through representation editing. Unlike methods that apply one-rank steering vectors, these approaches extend it by using higher-rank matrices and enhance the expressive power of steering vectors, allowing for richer control over model behavior (Wu et al., 2024a). Localized Fine-Tuning (LoFIT) (Yin et al., 2024) identifies critical attention heads for a task and trains offset vectors to modify their hidden representations, offering targeted adjustments. Compositional Subspace Representation Fine-Tuning (CS-ReFT) (Zhou, 2025) advances this by learning orthonormal subspace transformations for distinct skills, composed via a lightweight router, isolating edits in the hidden state to minimize cross-task interference. Unlike previous methods that train steering functions in the same space, we aim to develop representation fine-tuning methods to tune different attribute-specific subspaces and achieve the adaptive integration of multiple attribute steering spaces.

3 Motivation

To better illustrate our motivation and approach, we first revise ReFT (Wu et al., 2024a). In ReFT, it aims to steer the hidden representation $h \in \mathbb{R}^d$ by fine-tuning an r-dimensional subspace spanned by the rows of R. Specifically, we can define the intervention function Φ as:

$$\Phi(h; R, W, b) = h + R^\top \left(Wh + b - Rh\right), \tag{1}$$

where the learned low-rank projection matrix $R \in \mathbb{R}^{r \times d}$, is typically constrained to have orthonormal rows ($RR^\top = I_r$), and $W \in \mathbb{R}^{r \times d}$, $b \in \mathbb{R}^r$ are trainable parameters. $\Phi(h; R, W, b)$ is integrated into the model's representations to guide the output towards desired attributes. The steering-affected

output is subsequently optimized to minimize the target objective, thereby refining the parameters W, b, and R. This method enables efficient steering of model representations by manipulating the hidden activations in a learned, low-rank subspace.

While ReFT demonstrates encounters significant limitations when applied to multi-attribute scenarios. ReFT assumes a single attribute per input, but real-world inputs often involve multiple attributes. Training the matrix R on such multi-attribute inputs forces all steering directions into the same space, causing interference that hinders the model's ability to balance the needs of each attribute, ultimately limiting its performance across all attributes. Zhou (2025) attempts to address this by partitioning R into equal-sized subspaces, each dedicated to a specific attribute, in an effort to reduce interference. However, this approach overlooks the fact that different attributes require subspaces of varying sizes based on their expressive needs. As a result, attributes with higher complexity may not receive enough capacity for effective steering, while simpler attributes may waste valuable space.

To address these challenges, we propose Multi-Subspace Representation Steering (MSRS). MSRS mitigates the interference between attributes by assigning attribute-specific orthogonal subspaces and adapts the size of each subspace to fit the expressive needs of the corresponding attribute by utilizing Singular Value Decomposition (SVD), enabling dynamic adjustment for efficient representational space use. Furthermore, we introduce a shared subspace that captures common steering directions across attributes while learning the intricate interactions between them. This shared subspace enables the model to learn complex combinatory relationships between attributes, offering a more flexible and effective integration than methods that simply use gating mechanisms to combine attribute-specific subspaces.

4 METHODOLOGY

4.1 MULTI-ATTRIBUTE STEERING DIRECTION EXTRACTION

To enable precise and simultaneous control over multiple attributes, we extract steering directions that disentangle shared and attribute-specific features, identifying significant directions in the activation space.

Attribute-wise Activation Aggregation. To extract the primary steering directions for each attribute from the activation values, we first capture the key feature representations for each attribute. In detail, for each attribute i, we compute the average activation τ_i from its corresponding dataset $\mathcal{D}_i$. Specifically, we extract the model's intermediate activation $h^l_{i,j}$ of layer l for each sample j at the last token, as it integrates information from all preceding tokens, thus capturing the full prompt context (Lei & Cooper, 2025). The average activation for attribute i is defined as:

$$\tau_i = \frac{1}{|\mathcal{D}_i|} \sum_{j=1}^{|\mathcal{D}_i|} h^l_{i,j}. \tag{2}$$

To integrate information across all n attributes, we construct a combined activation matrix τ_c by concatenating the average activations:

$$\tau_c = [\tau_1 \mid \tau_2 \mid \ldots \mid \tau_n] \in \mathbb{R}^{d \times n}. \tag{3}$$

Shared and Specific Subspace Extraction. To retain common knowledge while enabling attribute-specific steering, we perform singular value decomposition (SVD) on the aggregated activation matrix τ_c:

$$\tau_c = U_c \Sigma_c V_c^\top. \tag{4}$$

We adaptively select the smallest number r_s such that the cumulative energy (sum of top r_s singular values) accounts for at least 90% of the total energy in Σ_c. This yields the shared subspace of V_c:

$$B_{\text{shared}} = V^\top_{c,1:r_s} \in \mathbb{R}^{r_s \times d}. \tag{5}$$

Intuitively, B_{shared} captures the dominant shared directions across all attributes. By selecting the top singular vectors, we capture high-variance directions that represent the most expressive steering dimensions. It allows us to automatically allocate varying subspace sizes for each attribute based on

its expressive needs. For complex attributes, it selects more top vectors, while simpler attributes are allocated smaller subspaces, effectively addressing the mismatch between each attribute's steering capacity and its allocated subspace size.

For each attribute i, we then isolate attribute-specific directions by projecting τ_i onto the shared subspace and computing the residual:

$$H_{\text{res}}^{(i)} = \tau_i - B_{\text{shared}}^{\top} B_{\text{shared}} \tau_i. \tag{6}$$

Applying SVD to $H_{\text{res}}^{(i)}$, we obtain:

$$H_{\text{res}}^{(i)} = U^{(i)} S^{(i)} V^{(i)\top} \tag{7}$$

Similarly, we select the smallest number r_i such that the top r_i singular values of $S^{(i)}$ explain at least 90% of the total energy, and define the private subspace as:

$$B_i = \left(V_{1:r_i}^{(i)} \right)^{\top} \in \mathbb{R}^{r_i \times d}. \tag{8}$$

Generally, B_i captures directions orthogonal to the shared subspace, preserving attribute-specific semantics. This adaptive allocation allows each attribute subspace to retain only as much representational capacity as needed, reflecting its inherent complexity.

The alignment matrix S_{align} is constructed by concatenating the shared and private subspace bases:

$$S_{\text{align}} = [B_{\text{shared}}, B_1, B_2, \ldots, B_n] \in \mathbb{R}^{(r_s + \sum_{i=1}^{n} r_i) \times d}. \tag{9}$$

4.2 Adaptive Subspace Selecting

To steer multiple attributes effectively, it is crucial to avoid the interference that arises when steering vectors for different attributes are trained in the same space. Furthermore, traditional methods often rely on summing or averaging these vectors, which often fail to produce an effective combination, as different attributes may require different subspace sizes or levels of emphasis. In contrast, we propose an adaptive mechanism that enables the model to train in a specific subspace and learn to combine different steering subspaces optimally, overcoming prior limitations.

Based on equation 1, we introduce a mask network $m(h) = \text{sigmoid}(\text{MLP}(h)) \in [0,1]^r$, which assigns weights to each subspace dimension. The intervention function becomes:

$$\Phi_{l,p}(h; R, W, b, m) = h + R^{\top} \text{diag}(m(h))(Wh + b - Rh), \tag{10}$$

where $\text{diag}(m(h)) \in \mathbb{R}^{r \times r}$ is a diagonal matrix.

4.3 Optimization Objective

We optimize the steering function $\Phi_{l,p}(h; R, W, b, m)$. Applying it to the representation $H_{l,p}$ at layer l and position p. This changes the representation and influences the model's output, which is then used to compute the task-specific loss $\mathcal{L}_{\text{task}}$, which is defined as the standard cross-entropy loss between the predicted logits and the ground truth labels, reflecting the model's performance on the downstream task.

To enable the steering function to perform meaningful and disentangled attribute control, we introduce a subspace regularization term. Specifically, to encourage adaptive selection of relevant subspaces, we define a binary prior mask $m_{\text{prior}} \in \{0,1\}^r$, where entries corresponding to the shared subspace B_{shared} and the attribute-specific subspace B_i are set to 1, and all others to 0. The regularization loss is defined as:

$$\mathcal{L}_{\text{reg}} = \| m(h) - m_{\text{prior}} \|_2^2 . \tag{11}$$

This loss encourages the model to steer primarily within subspaces that are relevant to the target attribute, while suppressing activation in unrelated dimensions.

We further encourage the learned representation R to align with the structured basis $S_{\text{align}} \in \mathbb{R}^{k \times d}$, as defined in equation 9. The alignment loss is defined as:

$$\mathcal{L}_{\text{align}} = 1 - \frac{\langle R, S_{\text{align}} \rangle}{\|R\|_2 \|S_{\text{align}}\|_2}, \tag{12}$$

which encourages R to lie in the subspace spanned by both shared and attribute-specific directions, promoting more controllable and semantically meaningful representations during training.

The overall optimization objective is as follows:

$$\mathcal{L} = \mathcal{L}_{\text{task}} + \lambda_1 \mathcal{L}_{\text{reg}} + \lambda_2 \mathcal{L}_{\text{align}}, \tag{13}$$

where $\lambda_1, \lambda_2 > 0$ are hyperparameters balancing the terms. This optimization guarantees both attribute-wise subspace alignment and inter-attribute separation, ultimately yielding an effective steering space capable of precise and disentangled multi-attribute control. By integrating different subspaces with a weighting network, we enable adaptive subspace combination, alleviating trade-offs and optimizing performance across diverse attributes' steering. See Appendix A for the detailed algorithm.

4.4 Dynamic Intervention Position Selection

Previous steering approaches often apply interventions at the same token position p across different attributes, which can lead to interference between attributes. To overcome this limitation, we propose a dynamic selection method that identifies the most relevant token position p_i for each attribute i by projecting token representations onto attribute-specific subspaces in $\mathbb{R}^d$. This enhances steering effectiveness for attribute i by targeting interventions at the most influential tokens.

Consider an input sequence with token representations $h_1, h_2, \ldots, h_T$, where T denotes the sequence length. For each attribute i, we project the token representations onto its corresponding subspace R_i. The projection of a token representation h_t onto this subspace is computed as:

$$\text{proj}_{R_i}(h_t) = R_i^\top R_i h_t. \tag{14}$$

We then define the relevance score $s_{i,t}$ of token t with respect to attribute i as the L_2-norm of this projection:

$$s_{i,t} = \|\text{proj}_{R_i}(h_t)\|_2. \tag{15}$$

The intervention position p_i for attribute i is dynamically selected as:

$$p_i = \arg\max_{t \in \{1,\ldots,T\}} s_{i,t}, \tag{16}$$

ensuring that the token with the strongest alignment to the attribute-specific subspace is chosen. The steering function $\Phi_{l,p}(h; R, W, b, m)$ for attribute i is applied at position p_i, allowing different attributes to be steered at different tokens, reducing inter-attribute interference and enhancing control precision.

5 Experiments

5.1 Experimental Settings

Datasets and Metrics. To evaluate the effectiveness of our proposed method, we conduct experiments on three pairs of datasets, each designed to assess trade-offs between specific attributes in language model steering. More details are provided in Appendix A.

TruthfulQA & BBQ: To evaluate *truthfulness* and *bias*, we use TruthfulQA with MC1, MC2, BLEU, and BLEURT scores (Lin et al., 2022), and BBQ with accuracy as the metric (Parrish et al., 2022).

Alpaca & Refusal: We evaluate *instruction following* via win rate on Alpaca (Taori et al., 2023; Li et al., 2023b) (vs. test-davinci-003), and *refusal* via Sorry-Bench scores judged by Mistral-7B-Instruct-v0.2 (Xie et al., 2025), which assesses the rejection of malicious instructions.

HelpSteer: We assess *helpfulness*, *coherence*, and *verbosity* by leveraging GPT-3.5-Turbo, following the setting of (Nguyen et al., 2025a), to rate model outputs on a 0–4 scale (Wang et al., 2023), which evaluates the quality of the generated content.

Additionally, we test the **utility** on several standard benchmarks, including Hellaswag (Zellers et al., 2019), RACE (Lai et al., 2017), MMLU (Hendrycks et al., 2020), OpenBookQA (Mihaylov et al.,

Table 1: Evaluation results on TruthfulQA, BBQ, Alpaca, Refusal, and HelpSteer. The best result is highlighted in bold, and the second-best is underlined. The values reported are the mean performance.

Method	**TruthfulQA**				**BBQ**	**Alpaca**	**Refusal**	**HelpSteer**		
	MC1 (↑)	MC2 (↑)	Bleu (↑)	Bleurt (↑)	Acc (↑)	Win (↑)	Sorry (↑)	Help. (↑)	Coh. (↑)	Ver. (↓)
Llama3-8b-inst.	$27.47_{\pm 0.21}$	$45.63_{\pm 0.28}$	$46.89_{\pm 0.25}$	$57.88_{\pm 0.33}$	$0.608_{\pm 0.011}$	$0.12_{\pm 0.010}$	$0.491_{\pm 0.021}$	$3.76_{\pm 0.03}$	$3.41_{\pm 0.04}$	$2.33_{\pm 0.03}$
ICL	$28.37_{\pm 0.25}$	$46.21_{\pm 0.26}$	$48.35_{\pm 0.23}$	$60.44_{\pm 0.31}$	$0.619_{\pm 0.013}$	$0.27_{\pm 0.011}$	$\underline{0.521}_{\pm 0.024}$	$3.82_{\pm 0.02}$	$3.64_{\pm 0.03}$	$2.41_{\pm 0.04}$
CAA	$28.41_{\pm 0.18}$	$47.55_{\pm 0.23}$	$45.42_{\pm 0.21}$	$61.54_{\pm 0.28}$	$0.629_{\pm 0.009}$	$0.29_{\pm 0.008}$	$0.493_{\pm 0.018}$	$3.77_{\pm 0.02}$	$\underline{3.89}_{\pm 0.02}$	$2.51_{\pm 0.03}$
ITI	$\mathbf{36.50}_{\pm 0.20}$	$\underline{54.29}_{\pm 0.29}$	$43.22_{\pm 0.24}$	$\underline{66.30}_{\pm 0.30}$	$0.612_{\pm 0.012}$	$0.23_{\pm 0.009}$	$0.280_{\pm 0.015}$	$3.82_{\pm 0.02}$	$3.21_{\pm 0.04}$	$2.06_{\pm 0.03}$
ReFT	$29.58_{\pm 0.16}$	$49.51_{\pm 0.22}$	$\underline{52.08}_{\pm 0.18}$	$64.06_{\pm 0.29}$	$0.637_{\pm 0.008}$	$\underline{0.30}_{\pm 0.008}$	$0.451_{\pm 0.020}$	$3.78_{\pm 0.02}$	$3.83_{\pm 0.02}$	$2.38_{\pm 0.02}$
MTL-LoRA	$34.82_{\pm 0.17}$	$47.23_{\pm 0.24}$	$51.78_{\pm 0.19}$	$63.26_{\pm 0.28}$	$\underline{0.641}_{\pm 0.008}$	$0.21_{\pm 0.007}$	$0.524_{\pm 0.022}$	$3.48_{\pm 0.04}$	$3.73_{\pm 0.03}$	$\mathbf{1.90}_{\pm 0.02}$
MAT-STEER	$29.29_{\pm 0.18}$	$49.67_{\pm 0.25}$	$43.81_{\pm 0.22}$	$55.31_{\pm 0.34}$	$0.622_{\pm 0.010}$	$0.14_{\pm 0.009}$	$0.420_{\pm 0.019}$	$\underline{3.84}_{\pm 0.02}$	$3.63_{\pm 0.03}$	$2.29_{\pm 0.03}$
MSRS(Ours)	$\underline{34.91}_{\pm 0.12}$	$\mathbf{56.32}_{\pm 0.20}$	$\mathbf{52.32}_{\pm 0.14}$	$\mathbf{66.75}_{\pm 0.26}$	$\mathbf{0.645}_{\pm 0.010}$	$\mathbf{0.36}_{\pm 0.008}$	$\mathbf{0.529}_{\pm 0.02}$	$\mathbf{3.89}_{\pm 0.01}$	$\mathbf{3.96}_{\pm 0.02}$	$\underline{2.04}_{\pm 0.02}$
Qwen2-7b-inst.	$26.38_{\pm 0.20}$	$45.41_{\pm 0.29}$	$49.63_{\pm 0.24}$	$65.28_{\pm 0.32}$	$0.634_{\pm 0.009}$	$0.12_{\pm 0.009}$	$0.384_{\pm 0.018}$	$3.51_{\pm 0.03}$	$3.80_{\pm 0.02}$	$2.28_{\pm 0.03}$
ICL	$26.84_{\pm 0.24}$	$48.33_{\pm 0.27}$	$49.79_{\pm 0.22}$	$66.63_{\pm 0.30}$	$0.633_{\pm 0.010}$	$0.16_{\pm 0.010}$	$0.413_{\pm 0.020}$	$3.65_{\pm 0.03}$	$\mathbf{3.88}_{\pm 0.02}$	$2.40_{\pm 0.03}$
CAA	$28.44_{\pm 0.17}$	$47.25_{\pm 0.25}$	$48.35_{\pm 0.20}$	$58.97_{\pm 0.35}$	$0.635_{\pm 0.008}$	$0.26_{\pm 0.008}$	$0.404_{\pm 0.019}$	$\underline{3.73}_{\pm 0.02}$	$\underline{3.87}_{\pm 0.02}$	$\underline{2.20}_{\pm 0.02}$
ReFT	$\underline{29.83}_{\pm 0.15}$	$48.69_{\pm 0.24}$	$\underline{52.57}_{\pm 0.17}$	$71.15_{\pm 0.26}$	$0.636_{\pm 0.008}$	$\underline{0.43}_{\pm 0.007}$	$0.421_{\pm 0.017}$	$3.63_{\pm 0.03}$	$3.78_{\pm 0.02}$	$2.38_{\pm 0.03}$
MAT-STEER	$23.08_{\pm 0.19}$	$\underline{49.51}_{\pm 0.28}$	$51.72_{\pm 0.19}$	$\underline{72.53}_{\pm 0.25}$	$\underline{0.641}_{\pm 0.007}$	$0.18_{\pm 0.009}$	$\underline{0.429}_{\pm 0.016}$	$3.70_{\pm 0.02}$	$3.77_{\pm 0.02}$	$2.25_{\pm 0.02}$
MSRS(Ours)	$\mathbf{34.72}_{\pm 0.11}$	$\mathbf{53.27}_{\pm 0.21}$	$\mathbf{53.10}_{\pm 0.12}$	$\mathbf{74.90}_{\pm 0.21}$	$\mathbf{0.642}_{\pm 0.006}$	$\mathbf{0.45}_{\pm 0.007}$	$\mathbf{0.445}_{\pm 0.011}$	$\mathbf{3.76}_{\pm 0.01}$	$3.82_{\pm 0.01}$	$\mathbf{2.17}_{\pm 0.02}$
Mistral-7b-v0.3	$18.83_{\pm 0.28}$	$36.54_{\pm 0.33}$	$41.56_{\pm 0.29}$	$54.52_{\pm 0.38}$	$0.614_{\pm 0.014}$	$0.14_{\pm 0.012}$	$0.631_{\pm 0.028}$	$3.75_{\pm 0.03}$	$3.92_{\pm 0.02}$	$2.36_{\pm 0.04}$
ICL	$21.92_{\pm 0.26}$	$49.23_{\pm 0.30}$	$44.11_{\pm 0.27}$	$57.64_{\pm 0.35}$	$0.622_{\pm 0.013}$	$0.18_{\pm 0.011}$	$0.642_{\pm 0.025}$	$3.77_{\pm 0.03}$	$\mathbf{3.94}_{\pm 0.01}$	$2.39_{\pm 0.03}$
CAA	$28.77_{\pm 0.19}$	$\underline{52.05}_{\pm 0.26}$	$\mathbf{53.85}_{\pm 0.20}$	$62.27_{\pm 0.32}$	$\mathbf{0.646}_{\pm 0.009}$	$0.22_{\pm 0.009}$	$0.663_{\pm 0.021}$	$3.76_{\pm 0.02}$	$3.83_{\pm 0.02}$	$\underline{2.27}_{\pm 0.02}$
ReFT	$\underline{30.07}_{\pm 0.17}$	$49.69_{\pm 0.28}$	$49.39_{\pm 0.23}$	$66.01_{\pm 0.31}$	$0.614_{\pm 0.011}$	$\underline{0.32}_{\pm 0.008}$	$\underline{0.668}_{\pm 0.019}$	$\underline{3.80}_{\pm 0.02}$	$3.85_{\pm 0.02}$	$2.33_{\pm 0.03}$
MAT-STEER	$25.45_{\pm 0.22}$	$48.38_{\pm 0.31}$	$49.46_{\pm 0.24}$	$62.62_{\pm 0.34}$	$0.631_{\pm 0.010}$	$0.19_{\pm 0.010}$	$0.644_{\pm 0.023}$	$3.78_{\pm 0.02}$	$3.86_{\pm 0.02}$	$2.29_{\pm 0.02}$
MSRS(Ours)	$\mathbf{31.32}_{\pm 0.11}$	$\mathbf{52.62}_{\pm 0.13}$	$\underline{50.61}_{\pm 0.08}$	$\mathbf{71.39}_{\pm 0.17}$	$\underline{0.644}_{\pm 0.007}$	$\mathbf{0.38}_{\pm 0.006}$	$\mathbf{0.693}_{\pm 0.013}$	$\mathbf{3.82}_{\pm 0.02}$	$\underline{3.93}_{\pm 0.01}$	$\mathbf{2.21}_{\pm 0.02}$

Table 2: General capabilities on several benchmarks.

Method	**HellaSwag**	**RACE**	**MMLU**	**OpenbookQA**	**GLUE**
Llama3-8b-instruct	0.801	0.671	<u>0.655</u>	0.556	0.726
ReFT	0.821	<u>0.677</u>	0.651	0.559	<u>0.757</u>
ITI	0.746	0.589	0.546	0.507	0.742
MTL	0.782	0.661	0.567	<u>0.562</u>	0.697
CAA	<u>0.833</u>	0.671	0.648	0.557	0.738
Ours	**0.839**	**0.683**	**0.657**	**0.568**	**0.775**
Qwen2-7b-instruct	0.831	0.625	0.695	0.606	0.825
ReFT	0.822	<u>0.644</u>	0.698	<u>0.613</u>	0.770
MTL	0.782	0.641	0.687	0.562	0.697
CAA	**0.837**	0.633	<u>0.698</u>	0.609	<u>0.830</u>
Ours	<u>0.835</u>	**0.648**	**0.702**	**0.616**	**0.832**
Mistral-7b-v0.3	0.862	0.678	0.618	0.602	0.681
ReFT	<u>0.872</u>	<u>0.679</u>	0.603	0.608	0.655
MTL	0.869	0.644	0.530	0.574	0.682
CAA	0.869	0.667	**0.619**	<u>0.611</u>	<u>0.693</u>
Ours	**0.874**	**0.681**	0.613	**0.622**	**0.707**

2018), and GLUE (Wang et al., 2018), all using accuracy as the evaluation metric. We aim to assess whether MSRS preserves the model's general abilities after fine-tuning, ensuring that task-specific steering does not degrade overall performance.

Models and Baselines. We evaluate our method MSRS on 4 models: Llama2-7B (Touvron et al., 2023), Llama3-8B-Instruct (Grattafiori et al., 2024), Qwen2-7B-Instruct (Team, 2024) , Mistral-7B-v0.3 (Jiang et al., 2023). And we compare MSRS with 6 baselines, grouped into 3 categories: **1) In-context Learning** (Brown et al., 2020): Utilizes prompts to steer attributes without altering model parameters. **2) Fine-tuning Methods**: **MTL-LoRA** (Yang et al., 2025), which employs low-rank adaptation for multi-task learning, enabling efficient attribute-specific fine-tuning. **ReFT** (Wu et al., 2024b), which adjusts model representations by fine-tuning the representation to align with target attributes. **3) Steering Methods**: **ITI** (Li et al., 2023a) applies inference-time interventions to modify activations and guide model outputs. **CAA** (Rimsky et al., 2024) steers behavior by injecting contrastive activation vectors derived from positive and negative examples. **MAT-STEER** (Nguyen et al., 2025b) implements multi-attribute steering with orthogonal constraints to minimize interference between attributes.

Experimental Setup. All experiments were conducted on NVIDIA V100 GPUs. We employed the Adam optimizer with a learning rate of 5×10^{-3} and a batch size of 2. The total subspace rank R was 8, with dual regularization coefficients $\lambda_1 = 0.3$, $\lambda_2 = 0.5$. Steering was applied to the 15^{th} transformer layer, selected as the most effective layer on the validation set. For each configuration, we report the average and standard deviation over 3 runs with different random seeds $\{42, 43, 44\}$.

5.2 MAIN RESULTS

MSRS excels in question-answering tasks. We first evaluate MSRS on TruthfulQA and BBQ to target the trade-off between truthfulness and bias. As shown in Table 1, baseline methods often fail to optimize both simultaneously. For example, ITI enhances truthfulness (MC1 of 36.50 on Llama3-8B-Instruct) but compromises bias mitigation (BBQ Acc of 0.612), while CAA improves bias (BBQ Acc of 0.646 on Mistral-7B) at the expense of truthfulness (MC1 of 28.77). ICL improves performance moderately across metrics but lacks standout improvements, and MAT-STEER enhances MC1, MC2 and Acc while its BLEU and BLEURT drop. Unlike prior methods that often improve one attribute at the expense of another, MSRS consistently balances both. MSRS achieves superior performance across both attributes on multiple models: on Llama3-8B-Instruct, it attains an MC2 score of 56.32 and BBQ accuracy of 0.645; on Qwen2-7B-Instruct, 53.27 and 0.642; and on Mistral-7B, 52.62 and 0.644. These results demonstrate MSRS achieves a more favorable balance across all models, demonstrating its capacity to jointly optimize conflicting objectives.

MSRS demonstrates strong performance in open-ended generation tasks. We assess MSRS on Alpaca, Refusal, and HelpSteer datasets to evaluate trade-offs between instruction following, refusal, and output quality attributes. As shown in Table 1, MSRS excels in balancing instruction following and refusal: on Alpaca, it achieves a win rate of 0.36 against test-davinci-003 on Llama3-8B-Instruct, outperforming ReFT (0.30), while on Refusal, it scores 0.529 on Sorry-Bench, surpassing CAA's 0.493. Baselines like ITI struggle with refusal (Sorry-Bench score of 0.280), sacrificing this attribute for others. For HelpSteer, MSRS achieves Helpfulness (3.89), Coherence (3.96), and Verbosity (2.04) on Llama3-8B-Instruct, while methods like MAT-STEER improve Helpfulness (3.84) but reduce Coherence (3.63) and Verbosity (2.29). MTL-LoRA achieves the best Verbosity (1.90) but at the cost of Helpfulness (3.48). MSRS outperforms existing baselines in harmonizing instruction-following with refusal and delivers consistently high-quality generations across all HelpSteer dimensions, demonstrating its ability to integrate diverse attribute objectives and enhance open-ended generation for specific tasks. Note that even in cases where MSRS is not the absolute best(e.g., Coherence on Mistral-7b-v0.3), it achieves the second-best result with a negligible gap (0.01) and retains overall superiority across all attributes.

MSRS maintains strong general capabilities on standard NLP benchmarks. To verify that MSRS does not compromise the model's overall natural language processing abilities, we evaluate its performance on several widely used benchmarks, as shown in Table 2. More results on the MMLU (Table 5) and GLUE benchmark tasks (Table 6), are provided in Appendix B. These tasks collectively assess a model's general reasoning, knowledge, and language understanding capabilities, providing a comprehensive measure of its robustness beyond attribute-specific steering.From the results, we can see baseline methods often compromise general capabilities due to overfitting to specific attributes. For example, ITI, while effective for truthfulness, suffers notable performance drops on general benchmarks like HellaSwag and MMLU on LLaMA3, indicating weakened commonsense and knowledge reasoning. ReFT offers modest gains (0.821 on HellaSwag, 0.677 on RACE) but underperforms on MMLU (0.651), reflecting limited generalization. CAA shows strong results on HellaSwag (0.837 on Qwen2) but inconsistent outcomes on RACE (0.625) and OpenBookQA (0.609), indicating instability.

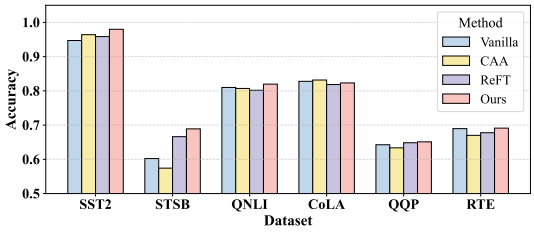


Figure 2: Comparison of model performance across GLUE.

In contrast, MSRS consistently matches or exceeds baseline performance across these benchmarks. On LLaMA3, MSRS achieves top scores. For GLUE, evaluations on LLaMA3 show MSRS achieving an average score of 0.775 across tasks, surpassing ITI (0.742) and ReFT (0.757). Figure 2 compares different tasks in the GLUE benchmark. MSRS achieves substantial improvements on SST-2 (0.979 vs. 0.947 for Vanilla) and STS-B (0.689 vs. 0.602), reflecting enhanced sentiment classification and semantic similarity. On other tasks such as QNLI and RTE, MSRS maintains competitive performance, matching or exceeding strong baselines. This robust performance stems from MSRS's shared subspace mechanism, which captures common steering directions across multiple tasks and attributes. By leveraging these shared representations,

Table 3: Comparison of steering subspace training strategies across datasets. The best result is highlighted in bold.

Method	TruthfulQA		BBQ	Alpaca	Refusal	HelpSteer		
	MC1	MC2	Acc	Win	Sorry	Help.	Coh.	Ver.
Llama3-8b-instruct								
Same Space	29.58	49.51	0.637	0.30	0.451	3.87	3.89	2.38
$\text{MSRS}_{\text{Attribute}}$	32.52	52.55	0.627	**0.36**	**0.529**	3.88	**3.96**	**2.24**
$\text{MSRS}_{\text{Rank}}$	**33.50**	**52.74**	**0.646**	0.35	0.527	**3.89**	3.95	2.28
Qwen2-7b-instruct								
Same Space	29.83	48.69	0.637	0.434	0.422	3.63	3.78	2.38
$\text{MSRS}_{\text{Attribute}}$	**34.72**	**53.27**	**0.642**	**0.451**	**0.446**	3.64	3.81	2.20
$\text{MSRS}_{\text{Rank}}$	26.41	47.65	0.635	0.442	0.439	**3.76**	**3.82**	**2.17**
Mistral-7b-v0.3								
Same Space	30.32	49.69	0.615	0.33	0.669	3.82	3.85	2.33
$\text{MSRS}_{\text{Attribute}}$	**30.07**	**52.62**	**0.644**	0.38	**0.693**	**3.82**	**3.93**	2.27
$\text{MSRS}_{\text{Rank}}$	28.36	49.94	0.631	**0.39**	0.673	3.76	3.87	**2.26**

Table 4: Comparison of Last token vs. Important token intervention. The best result is highlighted in bold.

Method	TruthfulQA		BBQ	Alpaca	Refusal	HelpSteer		
	MC1	MC2	Acc	Win	Sorry	Help.	Coh.	Ver.
LLaMA2-7B								
Last Token	26.41	42.88	0.631	0.12	0.579	2.70	2.68	2.73
Important Token	**29.10**	**48.60**	**0.644**	**0.13**	**0.583**	**3.12**	**3.06**	**2.47**
LLaMA3-8B-Instruct								
Last Token	33.50	52.74	0.648	**0.36**	**0.529**	**3.88**	**3.96**	2.24
Important Token	**33.71**	**56.32**	**0.655**	0.32	0.511	3.85	3.95	**1.99**
Qwen2-7B-Instruct								
Last Token	34.72	53.27	0.6421	**0.45**	0.446	**3.70**	3.82	2.17
Important Token	**36.12**	**55.63**	**0.6572**	0.42	**0.448**	3.69	**3.83**	**2.06**

MSRS enhances its generalization capabilities, enabling it to maintain strong performance on general NLP benchmarks while excelling in targeted steering objectives.

5.3 Ablation Study

Evaluating the effectiveness of adaptive subspace selecting mechanism. We conduct ablation studies comparing three strategies for training steering subspaces: (1) Same Space, where all attributes are trained in a single subspace without isolation; (2) $\text{MSRS}_{\text{Attribute}}$, the basis matrix $R \in \mathbb{R}^{r \times d}$ is partitioned into $n+1$ blocks: $R = [B_{\text{shared}} \parallel B_1 \parallel \ldots \parallel B_n]$, where B_{shared} is a shared subspace and each B_i corresponds to a specific attribute. The mask network $m(h)$ generates soft weights that are applied to each block individually, allowing the model to adaptively activate relevant attribute-specific subspaces; and (3) $\text{MSRS}_{\text{Rank}}$ treats the basis matrix $R \in \mathbb{R}^{r \times d}$ as a flat set of r independent basis vectors, without any explicit block structure. The mask network $m(h) \in \mathbb{R}^r$ assigns a soft weight to each row of R, enabling control at the level of individual basis directions. In our implementation, we adopt the $\text{MSRS}_{\text{Attribute}}$ configuration as the default setup for MSRS. This choice offers a good balance between interpretability and control, allowing the model to modulate behavior based on attribute-specific subspaces. Results are reported in Table 3 across multiple datasets and models. Training all attributes in the same space leads to suboptimal outcomes due to interference between conflicting attribute objectives. For instance, Same Space performs worse than both MSRS variants across nearly all metrics. MSRS mitigates such interference by decoupling subspaces and adaptively selecting them via $m(h)$. The $\text{MSRS}_{\text{Attribute}}$ configuration, which groups low-rank dimensions into attribute-aligned subspaces, strikes a good balance between parameter sharing and specialization. It consistently improves both truthfulness and bias mitigation across models. The $\text{MSRS}_{\text{Rank}}$ variant offers finer-grained control, further improving performance on LLaMA3-8B-Instruct, suggesting enhanced truthfulness. However, on Qwen2-7B-Instruct, it lags behind $\text{MSRS}_{\text{Attribute}}$. This may due to that the optimal subspace granularity may vary across model architectures, with certain models benefiting more from coarser subspace grouping. Results across additional metrics are shown in Appendix C (Table 7).

Validating the effectiveness of proposed dynamic intervention position selection mechanism. We compare two settings: (1) the *Last token*, where steering is applied to the last token in the

sequence, and (2) our proposed method, which applies steering at the dynamically selected *Important token*. As shown in Table 4, dynamic token selection consistently outperforms fixed-position steering across all models. For example, on LLaMA2-7B, MC1 improves from 26.41 to 29.10 and BBQ accuracy from 0.631 to 0.644. For HelpSteer metrics, dynamic token selection improves helpfulness (2.70 to 3.12), coherence (2.68 to 3.06), and verbosity (2.73 to 2.47), leading to more helpful, coherent, and informative responses. Similar gains are observed on LLaMA3-8B-Instruct and Qwen2-7B-Instruct. These results validate that selecting the most semantically relevant token for intervention mitigates inter-attribute interference and enables more effective attribute control. Results across additional metrics are shown in Appendix D (Table 9).

Determination of the optimal layer for steering interventions. We evaluate model performance by injecting steering vectors at different transformer layers. For each candidate layer, we apply interventions solely at that location, as shown in Figure 3. We observe that performance is highly sensitive to the intervention layer. Lower layers exhibit weak steering capability, likely due to insufficient semantic abstraction. Mid-to-upper layers generally yield better results, with layer 15 achieving the strongest overall performance. In contrast, deeper layers tend to overfit, leading to performance degradation.

These findings align with prior observations (Li et al., 2024), which suggest that optimal intervention layers typically reside in the early to middle layers of LLMs. Detailed results of diffenrent layer are provided in Appendix E (Table 9). To systematically select the intervention layer for multi-attribute subspace training, we perform a grid search over held-out validation splits to identify the layer that best balances the trade-offs between different attributes. Based on this strategy, we adopt the layer for steering in all subsequent experiments. This targeted selection ensures interventions are injected where they are most effective, thereby maximizing the utility of the learned multi-subspace representations.

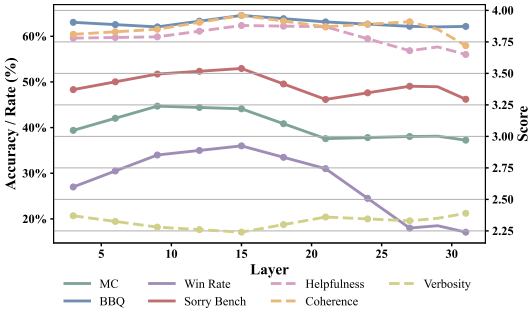


Figure 3: Performance of interventions at different transformer layers in LLaMA3-8B-Instruct. Mid-layer intervention consistently outperforms others.

6 Conclusion

We present Multi-Subspace Representation Steering (MSRS), a principled and effective framework for multi-attribute behavior control in LLMs. MSRS addresses key limitations of prior methods by introducing attribute-specific subspaces with SVD-guided dimensionality, a shared subspace to capture cross-attribute correlations, and a dynamic token selection mechanism for precise intervention. By integrating these components, MSRS mitigates steering conflicts and improves controllability across diverse attributes such as truthfulness, bias, instruction following, refusal, and generation quality. Extensive experiments demonstrate that MSRS consistently outperforms existing approaches in both task-specific and general-purpose settings, offering a scalable and robust solution for reliable and aligned language generation.

References

Reza Bayat, Ali Rahimi-Kalahroudi, Mohammad Pezeshki, Sarath Chandar, and Pascal Vincent. Steering large language model activations in sparse spaces. *arXiv preprint arXiv:2503.00177*, 2025.

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

Yuanpu Cao, Tianrong Zhang, Bochuan Cao, Ziyi Yin, Lu Lin, Fenglong Ma, and Jinghui Chen. Personalized steering of large language models: Versatile steering vec-

tors through bi-directional preference optimization. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 49519–49551. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/58cbe393b4254da8966780a40d023c0b-Paper-Conference.pdf.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv e-prints*, pp. arXiv–2407, 2024.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Shawn Im and Yixuan Li. A unified understanding and evaluation of steering methods. *arXiv preprint arXiv:2502.02716*, 2025.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

Junfeng Jiao, Saleh Afroogh, Yiming Xu, and Connor Phillips. Navigating llm ethics: Advancements, challenges, and future directions, 2025. URL https://arxiv.org/abs/2406.18841.

Tong Jiao, Jian Zhang, Kui Xu, Rui Li, Xi Du, Shangqi Wang, and Zhenbo Song. Enhancing fairness in llm evaluations: Unveiling and mitigating biases in standard-answer-based evaluations. In *Proceedings of the AAAI Symposium Series*, volume 4(1), pp. 56–59, 2024.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. RACE: Large-scale ReAding comprehension dataset from examinations. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel (eds.), *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 785–794, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1082. URL https://aclanthology.org/D17-1082/.

Florian Le Bronnec, Alexandre Verine, Benjamin Negrevergne, Yann Chevaleyre, and Alexandre Allauzen. Exploring precision and recall to assess the quality and diversity of LLMs. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11418–11441, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.616. URL https://aclanthology.org/2024.acl-long.616/.

Ge Lei and Samuel J Cooper. The representation and recall of interwoven structured knowledge in llms: A geometric and layered analysis. *arXiv preprint arXiv:2502.10871*, 2025.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 41451–41530. Curran Associates, Inc., 2023a. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/81b8390039b7302c909cb769f8b6cd93-Paper-Conference.pdf.

Shen Li, Liuyi Yao, Lan Zhang, and Yaliang Li. Safety layers in aligned large language models: The key to llm security. *arXiv preprint arXiv:2408.17003*, 2024.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Alpacaeval: An automatic evaluator of instruction-following models. https://github.com/tatsu-lab/alpaca_eval, 5 2023b.

Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3214–3252, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.229. URL https://aclanthology.org/2022.acl-long.229/.

Xinyu Ma, Yifeng Xu, Yang Lin, Tianlong Wang, Xu Chu, Xin Gao, Junfeng Zhao, and Yasha Wang. DRESSing up LLM: Efficient stylized question-answering via style subspace editing. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=mNVR9jJYqK.

Andrea Matarazzo and Riccardo Torlone. A survey on large language models with some insights on their capabilities and limitations. *arXiv preprint arXiv:2501.04040*, 2025.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2381–2391, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1260. URL https://aclanthology.org/D18-1260/.

Duy Nguyen, Archiki Prasad, Elias Stengel-Eskin, and Mohit Bansal. Multi-attribute steering of language models via targeted intervention. *arXiv preprint arXiv:2502.12446*, 2025a.

Duy Nguyen, Archiki Prasad, Elias Stengel-Eskin, and Mohit Bansal. Multi-attribute steering of language models via targeted intervention, 2025b. URL https://arxiv.org/abs/2502.12446.

Narmeen Oozeer, Luke Marks, Fazl Barez, and Amirali Abdullah. Beyond linear steering: Unified multi-attribute control for language models. *arXiv preprint arXiv:2505.24535*, 2025.

Alicia Parrish, Angelica Chen, Nikita Nangia, Vishakh Padmakumar, Jason Phang, Jana Thompson, Phu Mon Htut, and Samuel Bowman. BBQ: A hand-built bias benchmark for question answering. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 2086–2105, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.165. URL https://aclanthology.org/2022.findings-acl.165/.

Libo Qin, Qiguang Chen, Xiachong Feng, Yang Wu, Yongheng Zhang, Yinghui Li, Min Li, Wanxiang Che, and Philip S Yu. Large language models meet nlp: A survey. *arXiv preprint arXiv:2405.12819*, 2024.

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

Zhe Su, Xuhui Zhou, Sanketh Rangreji, Anubha Kabra, Julia Mendelsohn, Faeze Brahman, and Maarten Sap. Ai-liedar: Examine the trade-off between utility and truthfulness in llm agents. *arXiv preprint arXiv:2409.09013*, 2024.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

Qwen Team. Qwen2 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

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

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

Tianlong Wang, Xianfeng Jiao, Yinghao Zhu, Zhongzhi Chen, Yifan He, Xu Chu, Junyi Gao, Yasha Wang, and Liantao Ma. Adaptive activation steering: A tuning-free llm truthfulness improvement method for diverse hallucinations categories. In *Proceedings of the ACM on Web Conference 2025*, pp. 2562–2578, 2025.

Zhilin Wang, Yi Dong, Jiaqi Zeng, Virginia Adams, Makesh Narsimhan Sreedhar, Daniel Egert, Olivier Delalleau, Jane Polak Scowcroft, Neel Kant, Aidan Swope, et al. Helpsteer: Multi-attribute helpfulness dataset for steerlm. *arXiv preprint arXiv:2311.09528*, 2023.

Yotam Wolf, Noam Wies, Dorin Shteyman, Binyamin Rothberg, Yoav Levine, and Amnon Shashua. Tradeoffs between alignment and helpfulness in language models with steering methods, 2025. URL https://arxiv.org/abs/2401.16332.

Zhengxuan Wu, Aryaman Arora, Zheng Wang, Atticus Geiger, Dan Jurafsky, Christopher D. Manning, and Christopher Potts. Reft: Representation finetuning for language models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 63908–63962. Curran Associates, Inc., 2024a. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/75008a0fba53bf13b0bb3b7bff986e0e-Paper-Conference.pdf.

Zhengxuan Wu, Aryaman Arora, Zheng Wang, Atticus Geiger, Dan Jurafsky, Christopher D Manning, and Christopher Potts. Reft: Representation finetuning for language models. *Advances in Neural Information Processing Systems*, 37:63908–63962, 2024b.

Tinghao Xie, Xiangyu Qi, Yi Zeng, Yangsibo Huang, Udari Madhushani Sehwag, Kaixuan Huang, Luxi He, Boyi Wei, Dacheng Li, Ying Sheng, Ruoxi Jia, Bo Li, Kai Li, Danqi Chen, Peter Henderson, and Prateek Mittal. SORRY-bench: Systematically evaluating large language model safety refusal. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=YfKNaRktan.

Dayu Yang, Fumian Chen, and Hui Fang. Behavior alignment: a new perspective of evaluating llm-based conversational recommendation systems. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2286–2290, 2024.

Yaming Yang, Dilxat Muhtar, Yelong Shen, Yuefeng Zhan, Jianfeng Liu, Yujing Wang, Hao Sun, Weiwei Deng, Feng Sun, Qi Zhang, Weizhu Chen, and Yunhai Tong. Mtl-lora: Low-rank adaptation for multi-task learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(20):22010–22018, Apr. 2025. doi: 10.1609/aaai.v39i20.35509. URL https://ojs.aaai.org/index.php/AAAI/article/view/35509.

Fangcong Yin, Xi Ye, and Greg Durrett. Lofit: Localized fine-tuning on llm representations, 2024. URL https://arxiv.org/abs/2406.01563.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1472. URL https://aclanthology.org/P19-1472/.

Andy Zhou. Compositional subspace representation fine-tuning for adaptive large language models, 2025. URL https://arxiv.org/abs/2503.10617.

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

A IMPLEMENTATION DETAILS

A.1 Algorithm Figure 1b illustrates the overall structure of MSRS, which integrates shared and attribute-specific subspaces for precise and disentangled multi-attribute control. To complement this, we present the detailed training algorithm in Algorithm 1.

The MSRS training process begins by computing activation statistics from each attribute-specific dataset $\mathcal{D}_i$, aggregating token representations across examples to form mean activations τ_i. These activations are used to extract a shared subspace B_{shared} via singular value decomposition (SVD) over concatenated attribute activations, capturing the dominant, overlapping directions.

Next, we derive private subspaces B_i for each attribute i by removing the shared components from their individual activations, followed by a second SVD on the residuals. The final aligned subspace S_{align} is constructed by concatenating the shared and private bases.

Training proceeds by optimizing the steerable representation function $\Phi_{l,p}(h; R, W, b, m)$, applied to token representations at a selected layer l and position p. Once the intervention is applied, the resulting representation is passed through a Softmax function, which normalizes the output to produce a probability distribution over the possible classes. The Softmax function converts the raw scores (logits) produced by the intervention into a range between 0 and 1, ensuring that the sum of the outputs is equal to 1, representing the predicted class probabilities.The predicted probabilities are then compared to the ground truth labels y using Cross-Entropy loss. Cross-Entropy measures the difference between the predicted probabilities (after applying Softmax) and the true label distribution. It quantifies the performance of the model by penalizing the difference, with the objective of minimizing this loss during training. Finally, the computed cross-entropy loss is assigned as the task loss $\mathcal{L}_{\text{task}}$, which is minimized through the optimization process. This ensures that the steerable representation function $\Phi_{l,p}(h; R, W, b, m)$ is optimized to produce more accurate predictions in subsequent tasks.

The total loss function $\mathcal{L}$ combines the task loss $\mathcal{L}_{\text{task}}$ with two regularization terms. The first term regularizes the mask network $m(h)$, encouraging it to stay close to a prior mask m_{prior} through the ℓ_2-norm term $\lambda_1 \|m(h) - m_{\text{prior}}\|_2^2$. The second term enforces that the learned representation R aligns with a reference matrix S_{align} by minimizing the cosine dissimilarity, controlled by $\lambda_2 \mathcal{L}_{\text{align}}$, where:

$$\mathcal{L}_{\text{align}} = 1 - \frac{\langle R, S_{\text{align}} \rangle}{\|R\|_2 \|S_{\text{align}}\|_2}$$

The hyperparameters λ_1 and λ_2 balance the task loss and the regularization terms, ensuring that the model optimizes both task performance and alignment with the reference structure.

This procedure ensures that the learned steering function preserves attribute disentanglement, supports subspace coordination, and enables adaptive attribute combination. It ultimately yields a robust steering model capable of controlling multiple behavior dimensions in LLMs with minimal interference.

A.2 Datasets and Metircs We provide detailed descriptions of the datasets and evaluation metrics used in our experiments to assess multi-attribute steering performance.

TruthfulQA TruthfulQA (Lin et al., 2022) evaluates a model's ability to produce truthful and informative responses. We report:

- **MC1 (Single-true)**: Accuracy in selecting the single correct answer (highest log-probability among 4–5 candidates).
- **MC2 (Multi-true)**: Normalized probability assigned to all true reference answers.
- **BLEURT**, **BLEU**, and **ROUGE**: Generation-level similarity scores, computed as the difference between the maximum similarity to any true answer and any false answer.
- **GPT-judge** and **GPT-info**: GPT-based classifiers trained to predict human ratings of truthfulness and informativeness.

Algorithm 1 Multi-Subspace Representation Steering

Require: Datasets $\mathcal{D}_i$ for attributes $i = 1, \ldots, n$, model M, token position p, layer l, $\Phi_{l,p}(h; R, W, b, m)$, $m_{\text{prior}} \in [0,1]^{\text{r}}$, regulation hyperparameters λ_1, λ_2, label y
Ensure: Steering model $\Phi_{l,p}(h; R, W, b, m)$

Activation-wise Preparation
for each attribute i **do**
  $\tau_i = \frac{1}{|\mathcal{D}_i|} \sum_j h_{i,j}$
end for
Combine activations: $\tau_c = [\tau_1, \ldots, \tau_n]$
Shared Subspace Extraction
$\tau_c = U_c \Sigma_c V_c^\top$
$B_{\text{shared}} = U_{1:r_s}^{(i)\top}$ ▷ Top r_s directions
Private Subspace Extraction
for each attribute i **do**
  $\tau_i = U_i \Sigma_i V_i^\top$
  $H_{\text{res}}^{(i)} = \tau_i - R_{\text{shared}}^\top (R_{\text{shared}} \tau_i)$
  $H_{\text{res}}^{(i)} = U^{(i)} S^{(i)} V^{(i)\top}$
  $B_i = U_{1:r_i}^{(i)\top}$ ▷ Top r_i directions
end for
Construct Aligned Subspace
$S_{\text{align}} = [B_{\text{shared}}, B_1, \ldots, B_n]$
Optimize Representation Matrix
Initialize R, W, b, subspace mask $m(h)$
for each training step **do**
  $\Phi_{l,p}(h) = h + R^\top \cdot \text{diag}(m(h)) \cdot (Wh + b - Rh)$
  $\mathcal{L}_{\text{task}} = \text{CrossEntropy}\left(\text{Softmax}\left(\Phi_{l,p}(h)\right), y\right)$
  $\mathcal{L}_{\text{align}} = 1 - \frac{\langle R, S_{\text{align}} \rangle}{\|R\|_2 \|S_{\text{align}}\|_2}$
  $\mathcal{L} = \mathcal{L}_{\text{task}} + \lambda_1 \|m(h) - m_{\text{prior}}\|_2^2 + \lambda_2 \mathcal{L}_{\text{align}}$
  Update parameters R, W, b, m
end for
return $\Phi_{l,p}(h; R, W, b, m)$

BBQ The Bias Benchmark for QA (BBQ) (Parrish et al., 2022) measures social bias in QA outputs across nine social dimensions (e.g., race, gender). We report accuracy: whether the model selects the correct answer.

AlpacaEval (Li et al., 2023b): Measures instruction-following ability via win rate against a strong baseline (test-davinci-003), judged by GPT-3.5-Turbo.

Sorry-Bench (Xie et al., 2025): Evaluates instruction refusal on harmful inputs using a fine-tuned expert model (Mistral-7B-Instruct-v0.2). We report refusal accuracy based on the model's ability to reject malicious or unethical instructions.

HelpSteer (Wang et al., 2023) is a human-aligned benchmark for evaluating model helpfulness. Each response is rated by GPT-3.5-Turbo across:

- **Helpfulness**: Relevance and utility of the response.
- **Coherence**: Logical consistency and fluency.
- **Verbosity**: Appropriateness of response length.

Scores range from 0 (poor) to 4 (excellent), and we report the average for each dimension.

General Benchmarks. To verify that steering does not impair general capabilities, we evaluate on standard NLP tasks:

- **HellaSwag** (Zellers et al., 2019): commonsense inference; metric: *accuracy*.
- **RACE** (Lai et al., 2017): reading comprehension; metric: *accuracy*.
- **OpenBookQA** (Mihaylov et al., 2018): elementary science QA; metric: *accuracy*.

MMLU (Hendrycks et al., 2020) (Massive Multitask Language Understanding) is a challenging benchmark designed to evaluate a model's world knowledge and problem-solving ability under zero-shot and few-shot settings. It comprises 15,908 multiple-choice questions spanning 57 diverse subjects, including STEM, humanities, social sciences, and professional disciplines such as law and ethics. The tasks vary in difficulty from elementary to advanced levels, making MMLU an ideal benchmark for identifying model weaknesses across both general and specialized domains.

Each subject contains at least 100 test questions, exceeding the length of most human exams. The dataset is split into a few-shot development set (5 questions per subject), a validation set (1,540 questions), and a test set (14,079 questions). The evaluation metric is *average accuracy* across all subjects.

GLUE (Wang et al., 2018) (General Language Understanding Evaluation) is a widely used benchmark for evaluating general-purpose language understanding. It consists of nine diverse NLP tasks that span a range of linguistic phenomena, including sentiment analysis, paraphrase detection, textual entailment, and question answering. These tasks collectively assess a model's ability to perform natural language understanding in varied contexts.

The included tasks are:

MNLI (Multi-Genre Natural Language Inference): Predict entailment, contradiction, or neutrality between premise and hypothesis across multiple domains.

QNLI (Question Natural Language Inference): Convert question answering into an entailment task.

QQP (Quora Question Pairs): Detect if two questions from Quora have the same meaning.

SST-2 (Stanford Sentiment Treebank): Classify sentiment in movie reviews as positive or negative.

CoLA (Corpus of Linguistic Acceptability): Judge grammatical acceptability of a sentence.

STS-B (Semantic Textual Similarity Benchmark): Score sentence pairs on semantic similarity.

MRPC (Microsoft Research Paraphrase Corpus): Determine if two sentences are paraphrases.

RTE (Recognizing Textual Entailment): Binary entailment classification from multiple datasets.

WNLI (Winograd NLI): Resolve coreference in complex pronoun cases.

B General Capabilities of MSRS

Method	Math	Health	Physics	Business	Biology	Chemistry	CS	Economics	Eng.	Philosophy	Other	History	Geog.	Politics	Psych.	Culture	Law
LLaMA3-8B-Instruct	0.430	0.697	0.533	0.819	0.791	0.502	0.629	0.675	0.641	0.567	0.694	0.778	0.848	0.796	0.764	0.816	0.516
+ ITI	0.371	0.597	0.450	0.703	0.685	0.472	0.476	0.566	0.517	0.486	0.616	0.624	0.732	0.667	0.653	0.738	0.393
+ MTL-LoRA	0.365	0.587	0.477	0.769	0.687	0.429	0.534	0.547	0.503	0.523	0.547	0.703	0.737	0.698	0.690	0.687	0.432
+ CAA	0.436	0.695	0.545	0.828	0.786	0.488	0.638	**0.675**	**0.676**	0.566	0.689	0.772	0.843	0.787	0.768	**0.852**	0.513
+ Ours	**0.567**	**0.714**	**0.642**	**0.854**	**0.846**	**0.578**	**0.648**	0.759	0.759	**0.617**	**0.761**	**0.809**	**0.889**	**0.810**	**0.814**	0.850	**0.563**
Qwen2-7B-Instruct	0.567	0.712	0.630	**0.863**	0.844	0.574	0.650	0.757	0.738	0.594	0.754	0.801	0.874	0.813	0.803	0.825	0.557
+ MTL-LoRA	0.548	0.710	0.628	0.838	0.841	0.571	0.651	0.727	0.745	0.592	0.739	0.791	0.874	**0.818**	0.799	0.822	0.554
+ CAA	**0.568**	0.714	0.639	0.854	**0.850**	0.578	**0.653**	**0.760**	0.759	0.613	0.758	**0.815**	0.884	0.815	0.813	**0.852**	0.561
+ Ours	0.567	**0.715**	**0.642**	0.856	0.846	**0.578**	0.648	0.759	**0.762**	**0.617**	**0.761**	0.809	**0.889**	0.810	**0.814**	0.851	**0.563**
Mistral-7b-v0.3	0.387	0.658	0.475	0.766	0.742	**0.492**	0.587	0.545	**0.614**	0.567	0.682	0.761	**0.773**	**0.776**	0.737	0.771	0.503
+ MTL-LoRA	0.318	0.573	0.411	0.689	0.637	0.386	0.529	0.476	0.462	0.466	0.621	0.648	0.692	0.654	0.617	0.687	0.445
+ CAA	**0.388**	0.659	0.473	0.765	0.742	0.482	**0.595**	**0.585**	0.600	**0.570**	**0.685**	0.760	0.771	0.768	0.729	0.774	0.497
+ Ours	0.365	**0.687**	**0.477**	**0.769**	**0.687**	0.429	0.534	0.547	0.503	0.523	0.547	0.703	0.737	0.698	**0.790**	**0.787**	**0.532**

Table 5: MMLU per-task performance on different methods. The best result is highlighted in bold, and the second-best is underlined.

B.1 Performance on MMLU Benchmark

This section provides a comprehensive evaluation of MSRS on the Massive Multitask Language Understanding (MMLU) benchmark, assessing its ability to maintain and enhance general language capabilities across diverse domains. The MMLU benchmark, comprising 57 tasks across 17 subjects, serves as a rigorous testbed for evaluating model robustness beyond attribute-specific steering.

We compare MSRS against baseline ITI, MTL-LoRA, and CAA—on three base models: LLaMA3-8B-Instruct, Qwen2-7B-Instruct, and Mistral-7b-v0.3. Table 5 summarizes the per-subject performance of MSRS and baselines on the MMLU benchmark. The results reveal distinct patterns in how each method impacts general capabilities, with MSRS demonstrating superior consistency and enhancement over baselines.

MSRS enhances or preserves performance across subjects. On LLaMA3-8B-Instruct, MSRS achieves the highest accuracy in 14 out of 17 subjects, with significant gains in STEM fields such as mathematics (0.567 vs. 0.430 for the base model), physics (0.642 vs. 0.533), and chemistry (0.578 vs. 0.502). These improvements underscore MSRS's ability to bolster reasoning and knowledge retention in technically demanding domains. In humanities and social sciences, such as history (0.809 vs. 0.778) and psychology (0.814 vs. 0.764), MSRS also outperforms the base model, indicating broad applicability. For Qwen2-7B-Instruct, MSRS matches or exceeds the base model's performance in most subjects, with notable improvements in geography (0.889 vs. 0.874) and psychology (0.814 vs. 0.803). These gains, though modest, highlight MSRS's stability across high-performing base models, preserving their strong initial capabilities while enabling targeted enhancements. On Mistral-7b-v0.3, MSRS delivers substantial uplifts in subjects like psychology (0.790 vs. 0.737) and law (0.532 vs. 0.503), despite the base model's lower baseline performance. However, in mathematics (0.365 vs. 0.387), MSRS slightly underperforms, suggesting potential limitations in enhancing weaker base models in certain domains.

Baseline methods reveal trade-offs. ITI consistently degrades performance across most subjects due to its focus on truthfulness. On LLaMA3-8B-Instruct, ITI drops accuracy in mathematics to 0.371 (vs. 0.430) and computer science to 0.476 (vs. 0.629), reflecting a significant loss in general reasoning and knowledge. Similar declines are observed across all models, confirming ITI's unsuitability for preserving broad capabilities. MTL-LoRA also exhibits reduced performance, particularly in STEM subjects. For Mistral-7b-v0.3, accuracy in mathematics falls to 0.318 (vs. 0.387) and chemistry to 0.386 (vs. 0.492). On LLaMA3-8B-Instruct, declines are evident in economics (0.547 vs. 0.675) and engineering (0.503 vs. 0.641). These results suggest that MTL-LoRA's multi-task fine-tuning overfits to specific tasks, compromising the model's general knowledge base. CAA performs closer to the base models but rarely surpasses them. On Qwen2-7B-Instruct, CAA achieves comparable scores (e.g., 0.567 in mathematics, 0.854 in business) but trails MSRS in geography (0.884 vs. 0.889) and psychology (0.813 vs. 0.814). On Mistral-7b-v0.3, CAA maintains baseline levels (e.g., 0.774 in culture) without the consistent improvements seen in MSRS, indicating limited generalization beyond attribute steering.

The experimental results affirm that MSRS excels in maintaining and often enhancing general NLP capabilities across the MMLU benchmark, outperforming baseline methods in both consistency and performance. Unlike ITI and MTL-LoRA, which sacrifice broad knowledge for attribute-specific gains, and CAA, which offers limited improvement, MSRS achieves a better balance between targeted steering and general understanding.

B.2 Performance on GLUE Benchmark

In this section, we assess the performance of MSRS on the GLUE benchmark, a widely-used suite of tasks designed to evaluate natural language understanding capabilities. We evaluate on GLUE tasks including SST-2 (sentiment analysis), STS-B (semantic similarity), QNLI (question-answering), CoLA (linguistic acceptability), QQP (paraphrase detection), and RTE (textual entailment). We compare MSRS against two baseline methods: CAA and ReFT, across three base models: LLaMA3-8B-Instruct, Qwen2-7B-Instruct, and Mistral-7B-v0.3. Table 6 provides a comprehensive summary of the performance across all methods and models on the GLUE benchmark. MSRS demonstrates consistent improvements over the base models and often outperforms the baseline methods, showcasing its robustness across diverse linguistic tasks.

MSRS outperforms baselines across GLUE tasks. For LLaMA3-8B-Instruct, MSRS achieves an average score of 0.7748, surpassing the base model (0.7257), CAA (0.7384), and ReFT (0.7569). It excels particularly in SST-2 (0.9799) and QNLI (0.8097), highlighting its strengths in sentiment classification and reasoning tasks. Additionally, MSRS improves RTE (0.6912) compared to CAA (0.6701) and ReFT (0.6577). On Qwen2-7B-Instruct, MSRS records an average score of 0.8322, slightly better than ReFT (0.8300) and notably higher than CAA (0.7701). It achieves top perfor-

Model / Method	SST-2	STS-B	QNLI	CoLA	QQP	RTE	Avg.
LLaMA3-8B-Inst.	0.9471	0.5266	0.7208	0.8279	0.6426	0.6897	0.7257
+ CAA	0.9641	0.5743	0.7571	**0.8317**	0.6336	0.6701	0.7384
+ ReFT	0.9585	0.6662	0.8018	0.8185	0.6385	0.6577	0.7569
+ Ours	**0.9799**	**0.6890**	**0.8097**	0.8289	**0.6501**	**0.6912**	**0.7748**
Qwen2-7B-Inst.	0.9231	**0.7821**	0.8228	0.7500	0.8157	**0.8602**	0.8256
+ CAA	**0.9601**	0.6342	0.8070	**0.8317**	0.6336	0.6701	0.7701
+ ReFT	0.8750	0.7404	0.8637	0.7931	0.8227	0.8498	0.8300
+ Ours	0.8850	0.7311	**0.8675**	0.8210	**0.8672**	0.8577	**0.8322**
Mistral-7B-v0.3	0.8625	0.8313	0.4941	0.7731	0.7695	0.3548	0.6808
+ CAA	**0.8671**	**0.8357**	0.3225	0.8106	**0.7727**	0.3225	0.6551
+ ReFT	0.8249	0.7524	0.5401	0.8208	0.6398	0.5413	0.6927
+ Ours	0.8372	0.7972	**0.5712**	**0.8452**	0.6102	**0.6153**	**0.7066**

Table 6: Performance on GLUE benchmark tasks with different methods. The best result is highlighted in bold, and the second-best is underlined.

mance in QNLI (0.8675) and QQP (0.8672), demonstrating consistency across tasks, even though it slightly trails ReFT in SST-2 (0.8850 vs. 0.8750). With Mistral-7B-v0.3, MSRS attains an average score of 0.7066, outperforming the base model (0.6808), CAA (0.6551), and ReFT (0.6927). It shows significant gains in CoLA (0.8208) and RTE (0.6153), where baselines struggle (e.g., CAA's RTE: 0.3225).

Baseline methods show variability. CAA tends to improve specific tasks but lacks generalization. For instance, on LLaMA3-8B-Instruct, it boosts SST-2 (0.9641) but drops in RTE (0.6701). On Mistral-7B-v0.3, CAA severely underperforms in QNLI (0.3225) and RTE (0.3225). ReFT achieves competitive results in some areas but is inconsistent. On Qwen2-7B-Instruct, it excels in QNLI (0.8637) but lags in SST-2 (0.8750). For Mistral-7B-v0.3, ReFT improves CoLA (0.8452) yet struggles with QQP (0.6398).

The GLUE benchmark results underscore the effectiveness of MSRS in enhancing language understanding across a range of tasks. MSRS consistently achieves the highest average scores across all three models, LLaMA3-8B-Instruct (0.7748), Qwen2-7B-Instruct (0.8322), and Mistral-7B-v0.3, outperforming both CAA and ReFT. Its shared subspace mechanism enables MSRS to generalize effectively, balancing task-specific improvements with broad linguistic competence.

C Adaptive Subspace Selecting

This section presents an in-depth evaluation of the adaptive subspace selecting mechanism in MSRS. We assess the effectiveness of three subspace training strategies. (1) Same Space, where all attributes share a single subspace; (2) $\text{MSRS}_{\text{Attribute}}$, where a mask network adaptively weights attribute-specific subspaces; and (3) $\text{MSRS}_{\text{Rank}}$, where the mask network weights individual low-rank dimensions. Table 7 summarizes the performance of these strategies on TruthfulQA, BBQ, Alpaca, Refusal, and HelpSteer datasets, evaluated with LLaMA3-8B-Instruct, Qwen2-7B-Instruct, and Mistral-7B-v0.3 models. The analysis highlights the limitations of the Same Space approach and the advantages of adaptive subspace mechanisms.

Limitations of same space training. When all attributes are trained in a shared subspace, performance suffers due to interference between conflicting attribute objectives. For example, on LLaMA3-8B-Instruct, Same Space achieves an MC1 score of 29.58 on TruthfulQA, compared to 32.52 for $\text{MSRS}_{\text{Attribute}}$ and 33.50 for $\text{MSRS}_{\text{Rank}}$. Similarly, on Qwen2-7B-Instruct, it records a Sorry-bench score of 0.422, lagging behind $\text{MSRS}_{\text{Attribute}}$ at 0.446. This suggests that a single subspace struggles to accommodate diverse steering directions, resulting in suboptimal optimization.

Advantages of adaptive subspace selection. The MSRS variants address this interference by decoupling subspaces and applying adaptive weighting through the mask network $m(h)$. $\text{MSRS}_{\text{Attribute}}$: By grouping low-rank dimensions into attribute-specific subspaces, this strategy balances parame-

Method	TruthfulQA				BBQ	Alpaca	Refusal	HelpSteer		
	MC1	MC2	BLEU	BLEURT	Acc	Win Rate	Sorry-Bench	Help.	Coher.	Verb.
LLaMA3-8B-Instruct	29.58	48.43	49.63	57.88	0.608	0.12	0.491	3.78	3.91	2.33
Same Space	29.58	49.51	52.08	64.06	0.637	0.30	0.451	3.87	3.89	2.38
$MSRS_{attribute}$	32.52	52.55	**52.57**	68.46	0.627	**0.36**	**0.529**	3.88	**3.96**	**2.24**
$MSRS_{rank}$	**33.50**	**52.74**	52.32	**66.75**	**0.646**	0.35	0.527	**3.89**	3.95	2.28
Qwen2-7B-Instruct	26.38	45.41	49.63	65.28	0.638	0.12	0.384	3.51	3.83	2.28
Same Space	29.83	48.69	52.57	71.15	0.637	0.434	0.422	3.63	3.78	2.38
$MSRS_{attribute}$	**34.72**	**53.27**	**53.10**	**74.90**	**0.642**	**0.451**	**0.446**	3.64	3.81	2.20
$MSRS_{rank}$	26.41	47.65	49.88	64.30	0.635	0.442	0.439	**3.76**	**3.82**	**2.17**
Mistral-7B-v0.3	18.83	36.54	41.56	54.52	0.614	0.14	0.632	3.75	3.92	2.36
Same Space	30.32	49.69	49.39	66.01	0.615	0.33	0.669	3.82	3.85	2.33
$MSRS_{attribute}$	**30.07**	**52.62**	**50.61**	**71.39**	**0.644**	0.38	**0.693**	**3.82**	**3.93**	2.27
$MSRS_{rank}$	28.36	49.94	47.19	69.19	0.631	**0.39**	0.673	3.76	3.87	**2.26**

Table 7: Comparison of steering subspace training strategies across datasets. The best result is highlighted in bold.

ter sharing and specialization. On LLaMA3-8B-Instruct, it improves MC1 to 32.52 and BLEU to 52.57, while on Qwen2-7B-Instruct, it achieves an MC1 of 34.72 and BLEURT of 74.90, consistently surpassing Same Space. $\text{MSRS}_{\text{Rank}}$: This finer-grained approach weights each dimension individually, excelling on LLaMA3-8B-Instruct with an MC1 of 33.50 and MC2 of 52.74, indicating superior truthfulness. However, on Qwen2-7B-Instruct, it underperforms $\text{MSRS}_{\text{Attribute}}$ (MC1: 26.41 vs. 34.72), suggesting that excessive granularity may not always be beneficial.

Impact of subspace granularity. The optimal granularity varies by model. For LLaMA3-8B-Instruct, $\text{MSRS}_{\text{Rank}}$'s dimension-level control yields slight improvements over $\text{MSRS}_{\text{Attribute}}$, particularly in TruthfulQA metrics. In contrast, $\text{MSRS}_{\text{Attribute}}$ outperforms $\text{MSRS}_{\text{Rank}}$ on Qwen2-7B-Instruct and Mistral-7B-v0.3, suggesting that coarser subspace grouping aligns better with certain model architectures.

D Intervention Position Experiments

Method / Position	TruthfulQA							BBQ	Alpaca	Refusal	HelpSteer		
	MC1	MC2	BLEU	rouge1	BLEURT	Judge	Info	acc	Win Rate	Sorry-Bench	Help.	Coher.	Verb.
LLaMA2-7B													
Last Token	26.41	42.88	48.66	46.45	58.19	**31.05**	67.48	0.631	0.12	0.579	2.70	2.68	2.73
Important Token	**29.10**	**48.60**	**49.88**	**50.37**	**60.15**	28.85	**75.79**	**0.644**	**0.13**	**0.583**	**3.12**	**3.06**	**2.47**
LLaMA3-8B-Instruct													
Last Token	33.50	52.74	52.32	56.48	66.75	24.69	76.77	0.646	**0.36**	**0.529**	**3.88**	**3.96**	2.24
Important Token	**33.71**	**56.32**	**52.71**	**58.22**	**67.51**	**29.21**	78.13	0.655	0.32	0.511	3.85	3.95	**1.99**
Qwen2-7B-Instruct													
Last Token	34.72	53.27	51.10	55.50	70.90	28.85	**85.57**	0.6421	**0.45**	0.446	**3.70**	3.82	2.17
Important Token	**36.12**	**55.63**	**52.17**	**57.25**	**70.93**	**31.41**	84.09	**0.657**	0.42	**0.448**	3.69	**3.83**	**2.06**

Table 8: Comparison of Last token vs. Important token intervention. The best result is highlighted in bold.

This section provides a detailed evaluation of the dynamic intervention position selection mechanism. We compare two intervention strategies: (1) *Last Token*, where steering is applied to the final token in the sequence, and (2) *Important Token*, where steering is dynamically applied to the token most relevant to the target attribute, as identified by subspace projections. The experimental results, presented in Table 8, span multiple datasets (TruthfulQA, BBQ, Alpaca, Refusal, and HelpSteer) and models (LLaMA2-7B, LLaMA3-8B-Instruct, and Qwen2-7B-Instruct). Below, we analyze the effectiveness of the *Important Token* strategy and its advantages over the *Last Token* baseline.

LLaMA2-7B: The *Important Token* strategy yields notable improvements over *Last Token*, with MC1 increasing from 26.41 to 29.10, BLEU from 48.66 to 49.88, and BBQ accuracy from 0.631 to 0.644. On HelpSteer, it enhances Helpfulness (2.70 to 3.12), Coherence (2.68 to 3.06), and Verbosity (2.73 to 2.47), reflecting more guided and informative outputs.

LLaMA3-8B-Instruct: The *Important Token* approach boosts MC2 from 52.74 to 56.32 and rouge1 from 56.48 to 58.22, alongside a BBQ accuracy increase from 0.6457 to 0.6553. HelpSteer Verbosity improves from 2.24 to 1.99, though Helpfulness and Coherence remain stable, suggesting robust baseline performance.

Qwen2-7B-Instruct: Gains are observed in MC1 (34.72 to 36.12), BLEU (51.10 to 52.17), and BBQ accuracy (0.6421 to 0.6572). HelpSteer Verbosity rises from 2.17 to 2.06, with minimal changes in Helpfulness and Coherence, indicating consistent but moderate enhancements.

The experimental results validate the effectiveness of the *Dynamic Intervention Position Selection* mechanism. By targeting the most semantically relevant tokens, the *Important Token* strategy consistently outperforms the *Last Token* baseline across diverse datasets and models. These improvements are evident in enhanced truthfulness, reduced bias, and higher-quality generations, alongside greater interpretability of model behavior.

E STEERING LAYER SELECTION

Model / Layer	**TruthfulQA**							**BBQ**	**Alpaca**	**Refusal**	**HelpSteer**		
	MC1	MC2	BLEU	ROUGE1	BLEURT	Judge	Info	Acc	Win Rate	Sorry-Bench	Help.	Coher.	Verb.
LLaMA3-8B-Instruct													
3	29.83	48.95	52.08	55.26	65.28	32.27	**91.20**	0.631	0.27	0.483	3.78	3.81	2.37
9	33.01	**56.39**	49.88	55.26	**68.22**	22.00	82.89	0.620	0.34	0.517	3.79	3.85	2.28
15	**33.50**	52.74	**52.32**	**56.48**	66.75	24.69	76.77	**0.646**	**0.36**	**0.529**	**3.88**	**3.96**	**2.24**
21	27.94	47.21	52.18	55.02	65.45	**26.31**	72.26	0.632	0.31	0.462	3.87	3.87	2.36
27	26.86	49.24	49.83	53.12	63.77	25.23	71.85	0.622	0.18	0.491	3.68	3.91	2.33
Qwen2-7B-Instruct													
3	36.32	44.39	47.79	54.55	66.81	20.82	75.77	0.605	0.32	0.419	3.72	3.81	2.34
9	**36.41**	47.65	49.41	**58.03**	63.80	25.43	87.31	0.605	0.44	**0.449**	3.64	**3.84**	2.25
15	34.72	**53.27**	**53.10**	55.50	**74.90**	28.85	**90.95**	**0.642**	**0.45**	0.446	**3.76**	3.82	**2.17**
21	33.67	50.79	51.87	53.28	73.41	31.41	77.18	0.631	0.38	0.413	3.63	3.76	2.27
27	24.71	40.44	49.83	53.30	69.47	**36.62**	74.76	0.614	0.16	0.368	3.61	3.61	2.24
Mistral-7B-v0.3													
3	27.06	46.94	48.09	54.26	68.49	44.03	67.95	0.631	0.36	0.622	**3.86**	3.71	2.32
9	30.82	49.06	47.79	53.25	69.01	**51.07**	79.53	0.624	0.32	0.679	3.82	3.87	2.31
15	30.07	**52.62**	**50.61**	**57.95**	**71.39**	45.70	80.44	**0.644**	**0.38**	**0.693**	3.82	**3.93**	2.27
21	**31.32**	51.69	48.59	57.58	63.87	50.37	78.57	0.619	0.36	0.669	3.76	3.88	**2.23**
27	24.83	36.31	45.76	48.64	53.52	44.86	**84.03**	0.614	0.25	0.619	3.71	3.72	2.35

Table 9: Performance of interventions at different layers. The best result is highlighted in bold.

This section elaborates on the layer-wise ablation study conducted to identify the optimal transformer layer for injecting steering vectors in MSRS. We assess model performance by applying interventions at specific layers ($\{3, 9, 15, 21, 27\}$) across multiple datasets and models. The analysis highlights the sensitivity of steering effectiveness to layer selection and underscores the importance of optimizing this parameter. Table 9 summarizes the performance metrics for interventions at different layers, evaluated on TruthfulQA, BBQ, Alpaca, Refusal, and HelpSteer datasets using LLaMA3-8B-Instruct, Qwen2-7B-Instruct, and Mistral-7B-v0.3 models.

Performance Trends Across Layers Lower Layers (e.g., Layer 3): Interventions at lower layers exhibit limited steering capability. For example, LLaMA3-8B-Instruct at Layer 3 achieves an MC1 score of 29.83 and BBQ accuracy of 0.631, underperforming compared to higher layers. This can be attributed to the early layers' focus on syntactic rather than semantic representations, limiting their effectiveness for attribute control.

Mid-to-Upper Layers (e.g., Layer 15): Optimal performance is consistently observed at mid-to-upper layers, with Layer 15 standing out across all models. For LLaMA3-8B-Instruct, Layer 15 yields an MC1 of 33.50, BLEU of 52.32, and HelpSteer Helpfulness of 3.88. Similarly, Qwen2-7B-Instruct at Layer 15 achieves an MC1 of 34.72, BLEURT of 74.90, and BBQ accuracy of 0.642. These results suggest that mid-layers strike an effective balance between semantic abstraction and model generalization.

Deeper Layers (e.g., Layer 27): Performance degrades at deeper layers, likely due to overfitting or overly specialized representations. For instance, Mistral-7B-v0.3 at Layer 27 records an MC1 of 24.83 and BLEU of 45.76, indicating a reduced capacity to generalize effectively when interventions occur late in the transformer stack.

Layer Selection via Grid Search To determine the optimal intervention layer for multi-attribute subspace training, we employ a grid search over held-out validation splits. This method systematically evaluates performance across layers and attributes, identifying Layer 15 as the most effective choice for balancing trade-offs. This targeted selection is adopted in all subsequent experiments to ensure that steering interventions maximize the utility of the learned multi-subspace representations.

Method	MC1	MC2	BLEU	ROUGE-1	BLEURT	Info
LLaMA2	18.58	35.25	38.37	39.18	52.65	58.19
+ReFT	21.03	36.93	44.74	47.92	55.99	41.32
+ReFT + SVD	18.58	38.97	46.45	48.66	58.92	66.99
+ReFT + Align	**24.94**	**41.41**	**51.10**	**51.59**	**66.50**	**71.64**

Table 10: Ablation study on multi-subspace alignment strategies on TruthfulQA.

Evaluating the benefit of aligning learned steering subspaces with SVD-derived priors. We conduct an ablation study on LLaMA2-7B using three configurations: (1) standard ReFT, which learns a single steering subspace without incorporating prior structure; (2) **Naive SVD-based subspace concatenation**, which constructs a fixed basis matrix by directly concatenating attribute-relevant directions extracted via singular value decomposition (SVD). Specifically, for each attribute, we apply SVD on its activation representations to obtain a low-rank subspace, and then combine all attribute-specific bases with a shared subspace into a single matrix (3) **Our proposed subspace alignment strategy** not only leverages SVD priors m_{prior} for initial subspace selection, but also explicitly aligns the learned subspace usage with these priors through two mechanisms: a mask regularization loss $\mathcal{L}_{\text{reg}}$ and a directional alignment loss $\mathcal{L}_{\text{align}}$. This enables adaptive, interpretable, and task-aware control over attribute-specific representations. The results are shown in Table 10.

Our proposed method (`+ReFT+Align`) significantly outperforms both baselines across all metrics. It achieves a BLEU score of 51.10 and BLEURT of 66.50, representing clear improvements over both ReFT and SVD-only approaches. Notably, the MC1 score increases to 24.94 , more than 3 points over ReFT, indicating that aligning learned masks and subspaces with SVD priors enhances attribute-specific steering capacity. The Info score, which reflects informativeness in open-ended generation, also peaks at 71.64, supporting the conclusion that SVD-guided alignment enables more effective and disentangled attribute control. These results underscore the importance of explicitly incorporating structural priors during training, with SVD initialization serving as an effective foundation for guiding subspace learning.