BRIDGE: BI-SUBSPACE RESIDUAL GUIDANCE FOR SPECTRAL REPRESENTATION EDITING TOWARDS TRUTHFUL LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Transformer-based Large Language Models (LLMs) sometimes produce "hallucinations" by generating content that deviates from facts seen during pretraining. Interpretability studies have shown that the representations within LLMs encode rich semantic information. Representation editing techniques can intervene in LLM representations at low cost, thereby alleviating hallucinations. Existing training-based representation editing methods freeze the base model and learn interventions for hidden representations specific to the task. However, these methods still suffer from heavy parameter optimization and high computational overhead. Furthermore, these methods often perform uncalibrated directional shifts without accounting for the underlying geometric structure of the base model's representation space. Direct additive interventions applied without global structural alignment risk displacing valid representation manifolds, leading to over-editing or degradation of non-target behaviors. To overcome these limitations, we propose **B**i-subspace **Re**s**id**ual **G**uidance for Spectral Representation **E**diting (BRIDGE) to enhance LLM truthfulness. BRIDGE first constructs Target-Alignment and De-biasing cross-covariance matrices to delineate the semantic evolutionary trajectories across the baseline, untruthful, and truthful domains. Through Singular Value Decomposition (SVD), it extracts the principal singular vectors mapping directly into the target truthful subspace. Subsequently, ultra-lightweight learnable representation editing vectors are introduced into the low-dimensional subspace spanned by these principal singular vectors. Extensive experiments on multiple models and benchmark datasets demonstrate that BRIDGE not only effectively enhances performance on various factuality and logical reasoning tasks, but also reduces the learnable parameters by 20x to 3000x compared to existing methods.

1 INTRODUCTION

Large Language Models (LLMs) have demonstrated strong generative capabilities and are increasingly applied in Natural Language Processing (NLP) tasks (Wu et al., 2023). However, LLMs may generate responses that are fluent and instruction-compliant yet untruthful or logically inconsistent - a phenomenon commonly referred to as hallucination (Shuster et al., 2021; Ji et al., 2023). Hallucinations in LLMs are generally categorized into two types: Factuality hallucinations, where the generated content contradicts verifiable real-world facts; faithfulness hallucinations, where the output deviates from the user's instructions or the given context (Huang et al., 2025b). In this work, we focus on Factuality hallucinations, particularly cases where the model encodes correct factual knowledge in its internal representations but fails to generate it in the output.

To mitigate hallucinations, a common approach is to fine-tune pretrained LLMs on task-specific datasets (Dai & Le, 2015). However, full fine-tuning is often computationally expensive and memory-intensive. To address this limitation, Parameter-Efficient Fine-Tuning (PEFT) methods have been proposed, which update only a small subset of model parameters while keeping the rest frozen (Hu et al., 2022). In contrast, a growing body of interpretability research highlights that hidden representations within LLMs encode rich semantic and factual information (Turner et al., 2024; Li et al., 2023a; Geiger et al., 2024), inspiring representation editing as a lightweight alternative (Wu et al., 2024b). Some representation editing methods perform projection-editing-reconstruction

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

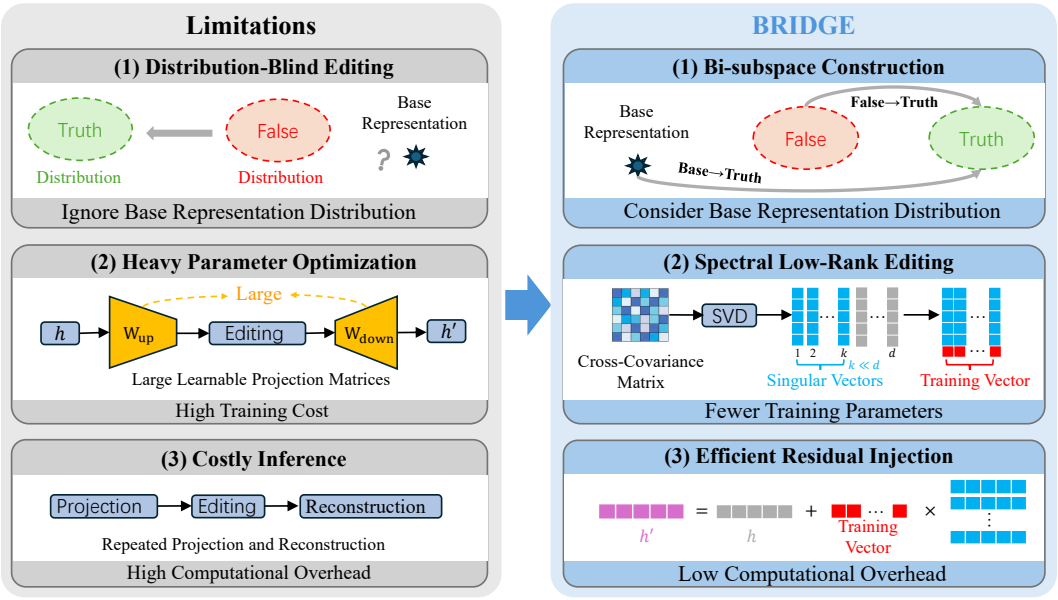


Figure 1: Motivation of BRIDGE. Existing representation editing methods rely on large projection matrices and repeated projection–reconstruction operations, resulting in high training and inference costs. Moreover, they ignore the distribution of the original representation within the latent space. BRIDGE addresses these limitations by discovering spectral Bi-subspaces, enabling efficient residual guidance with only ultra-lightweight learnable parameters.

steps on representations by learning a projection matrix containing a large number of parameters (Wu et al., 2024b; Zhang et al., 2024; Wu et al., 2024a). Others fine-tune the scaling factor of the model's attention heads to identify factually relevant attention heads, and then add a learnable bias to the representations of these highly relevant attention heads to improve the model's truthfulness (Yin et al., 2024). These methods aim to steer the model's output toward truthful responses by directly intervening in the internal representations of specific layers.

As shown in Figure 1, even though these methods are more lightweight compared to PEFT methods that change model weights, the number of learnable parameters is still enormous. Reducing the representation to a low-rank subspace still requires optimizing a large number of weight matrices, and explicit projection and reconstruction between the high and low dimensional spaces are required during inference, resulting in significant computational overhead. Furthermore, these methods focus on identifying the relative directional differences between truthful and untruthful regions. During inference, they ignore the distribution of the original representation within the latent space. Applying uncalibrated directional offset risks pushing representation into out-of-distribution latent regions

To overcome the limitations, we propose **Bi**-subspace **Resid**ual **G**uidance for Spectral Representation **E**diting (BRIDGE). Unlike prior methods, BRIDGE incorporates the base representation geometry at the distributional level during offline subspace extraction. Specifically, we construct the Target-Alignment and the De-biasing cross-covariance matrices to explicitly delineate the semantic evolutionary trajectories within the LLM's baseline, untruthful, and truthful domains. Through performing Singular Value Decomposition (SVD) on these matrices, we extract the top-k principal singular vectors that map the representations into the target subspace. We introduce learnable representation editing vectors into an ultra-low-dimensional subspace spanned by these principal singular vectors. We adopt a subspace substitution strategy to fully preserve the model's original inference information within the orthogonal complement space. Crucially, this sophisticated substitution logic can be mathematically streamlined into an exceptionally efficient direct residual injection operation. BRIDGE uses the base representation during offline subspace construction, but its current inference-time intervention is a static residual injection. Thus, the method incorporates base-position information in the learned subspaces rather than through query-dependent runtime conditioning. Finally, we conducted extensive experiments on multiple open-source models. Our method not only demonstrated strong performance but also used 20x-3000x fewer learned parameters.

2 PRELIMINARIES

2.1 LANGUAGE MODELING

We focus on autoregressive, decoder-only transformer-based (Vaswani et al., 2017) LLMs denoted as $\mathcal{F}_\theta$. An individual transformer layer consists of MHSA and FFN, which generate contextual representations of tokens' sequence. Given a sequence of t input tokens $x = (x_1, \cdots, x_t)$ and the model first embeds them into a list of representations $\mathbf{h}^0 = (\boldsymbol{h}_1^0, \cdots, \boldsymbol{h}_t^0)$. Then input $\mathbf{h}^0$ into L transformer layers, and the output of the l-th layer at time step t is hidden representation $\boldsymbol{h}_t^l \in \mathbb{R}^d$:

$$\boldsymbol{h}_t^l = \boldsymbol{h}_t^{l-1} + \boldsymbol{a}_t^l + \boldsymbol{m}_t^l, \tag{1}$$

where $\boldsymbol{a}^l$ and $\boldsymbol{m}^l$ are the representations of the MHSA and the FFN of the l-th layer, respectively. The general forms of representations of MHSA and FFN at the l-th layer and the t-th token are given by:

$$\begin{aligned} \boldsymbol{a}_t^l &= \boldsymbol{W}_{O^{\text{MHSA}}}^l \, \text{MHSA}^l \left(\gamma \left(\boldsymbol{h}_1^{l-1}, \boldsymbol{h}_2^{l-1}, \ldots, \boldsymbol{h}_t^{l-1} \right) \right), \\ \boldsymbol{m}_t^l &= \boldsymbol{W}_{value}^l \sigma \left(\boldsymbol{W}_{key}^l \gamma \left(\boldsymbol{h}_t^{l-1} + \boldsymbol{a}_t^l \right) \right), \end{aligned} \tag{2}$$

where $\boldsymbol{W}_{O^{\text{MHSA}}}^l$ denotes the output weights of the MHSA; $\boldsymbol{W}_{value}^l \in \mathbb{R}^{d \times d_F}$ and $\boldsymbol{W}_{key}^l \in \mathbb{R}^{d_F \times d}$ are the weight matrices of the FFN; σ and γ are the non-linear activation function and layer normalization, respectively. Finally, LLM projects the representation $\boldsymbol{h}_t^L$ of the final layer onto the distribution of the vocabulary $\mathcal{V}$ to predict the next token x_{t+1}:

$$\begin{aligned} \mathcal{F}_\theta \left(x_1, \ldots, x_t \right) &= \text{softmax} \left(\boldsymbol{W}_{\text{E}} \gamma \left(\mathbf{h}_t^L \right) \right) \\ &= \mathbb{P} \left(x_{t+1} \mid x_1, \ldots, x_t \right), \end{aligned} \tag{3}$$

where $W_E \in \mathbb{R}^{|\mathcal{V}| \times d}$ is the embedding matrix.

3 METHODOLOGY

3.1 REPRESENTATION EXTRACTION

Let an LLM consist of L transformer layers. We construct a triplet dataset $\mathcal{D} = \{(Q, A_{\text{pos}}, A_{\text{neg}})\}$ to induce truthful, untruthful, and baseline states within the LLM. Q is the question. A_{pos} and A_{neg} denote truthful and untruthful responses, respectively. We prompt the LLM with $Q + A_{\text{pos}}$, $Q + A_{\text{neg}}$, and Q to extract the FFN intermediate hidden representations at the last token, denoted as $\boldsymbol{H}_{\text{pos}}, \boldsymbol{H}_{\text{neg}}, \boldsymbol{H}_{\text{base}} \in \mathbb{R}^{N \times L \times d_F}$, where N signifies the total number of triplets. For layer $l \in \{0, 1, \ldots, L-1\}$, the extracted layer-wise representations are formalized as $\boldsymbol{H}_{\text{pos}}^{(l)}, \boldsymbol{H}_{\text{neg}}^{(l)}, \boldsymbol{H}_{\text{base}}^{(l)} \in \mathbb{R}^{N \times d_F}$.

3.2 BI-SUBSPACE CONSTRUCTION

Upon extracting the representation matrices $\boldsymbol{H}_{\text{base}}, \boldsymbol{H}_{\text{pos}}, \boldsymbol{H}_{\text{neg}} \in \mathbb{R}^{N \times d_F}$, respectively, we define two cross-covariance matrices to formalize the evolutionary trajectories within the latent space:

- Target-Alignment Cross-Covariance Matrix $\boldsymbol{C}_{\text{bp}} = \frac{1}{N} \boldsymbol{H}_{\text{base}}^T \boldsymbol{H}_{\text{pos}}$: Characterize the main alignment direction when mapping the base representation to the target space.
- De-biasing Cross-Covariance Matrix $\boldsymbol{C}_{\text{np}} = \frac{1}{N} \boldsymbol{H}_{\text{neg}}^T \boldsymbol{H}_{\text{pos}}$: Characterize the deflection relationship between the untruthful space and the truthful space.

To extract principal modes of variation across these subspaces, we leverage SVD to both matrices:

$$\boldsymbol{C}_{\text{bp}} = \boldsymbol{U}_{\text{bp}} \boldsymbol{\Sigma}_{\text{bp}} \boldsymbol{V}_{\text{bp}}^T, \quad \boldsymbol{C}_{\text{np}} = \boldsymbol{U}_{\text{np}} \boldsymbol{\Sigma}_{\text{np}} \boldsymbol{V}_{\text{np}}^T. \tag{4}$$

Here, $\boldsymbol{U}$ and $\boldsymbol{V}$ denote orthogonal matrices, and $\boldsymbol{\Sigma} = \text{diag}(\sigma_1, \ldots, \sigma_{d_F})$ contains the non-negative singular values sorted in descending order. Statistically, this factorization is equivalent to solving a constrained optimization problem that maximizes the projected cross-covariance between the respective latent spaces. Formally, taking the De-biasing Cross-Covariance matrix $\boldsymbol{C}_{\text{np}}$ as an exemplar, its leading pair of singular vectors $(\boldsymbol{u}_1, \boldsymbol{v}_1)$ satisfies:

$$\text{Cov}(\boldsymbol{H}_{\text{neg}} \boldsymbol{u}, \boldsymbol{H}_{\text{pos}} \boldsymbol{v}) = \max_{\|\boldsymbol{u}\|_2 = 1, \|\boldsymbol{v}\|_2 = 1} \boldsymbol{u}^T \boldsymbol{C}_{\text{np}} \boldsymbol{v}. \tag{5}$$

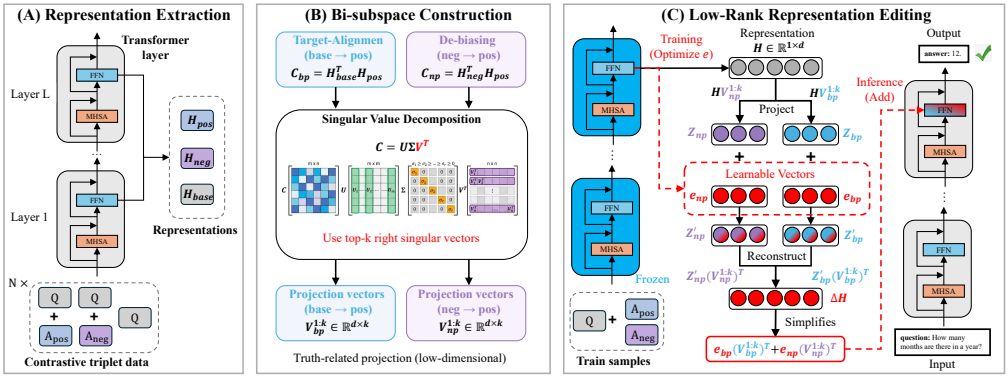


Figure 2: Overview of the BRIDGE framework. **(A) Representation Extraction:** Constructing contrastive triplets (Q, A_{pos}, A_{neg}) to extract hidden representations ($\boldsymbol{H}_{base}, \boldsymbol{H}_{pos}, \boldsymbol{H}_{neg}$) from the Transformer layers. **(B) Bi-subspace Construction:** Building Target-Alignment ($\boldsymbol{C}_{bp}$) and Debiasing ($\boldsymbol{C}_{np}$) cross-covariance matrices and applying SVD to identify the top-k right singular vectors as the principal projection bases ($\boldsymbol{V}_{:,:k}$). **(C) Low-Rank Representation Editing:** Optimizing learnable editing vectors ($\boldsymbol{e}_{bp}, \boldsymbol{e}_{np}$) during training and performing inference via efficient residual injection to steer the model toward truthful outputs.

Geometrically, $\boldsymbol{u}_i$ and $\boldsymbol{v}_i$ define a pair of coupled, optimal coordinate axes—acting as a semantic bridge—within their respective latent spaces. Projecting the representation matrices onto these singular vectors isolates the maximal linear correlation between the two distributions, where the singular value σ_i rigorously quantifies the magnitude of this shared co-varying energy. Within this decomposition framework, the left singular matrix (e.g., $\boldsymbol{U}_{\text{np}}$) uncovers the principal axes of perturbation within the source space (i.e., the untruthful domain). Conversely, the right singular matrices ($\boldsymbol{V}_{\text{bp}}$ and $\boldsymbol{V}_{\text{np}}$) precisely isolate the dominant covariant channels directed toward the target space.

Given that our ultimate objective is to steer the LLM toward truthful generations, our analysis focuses primarily on the right singular matrix $\boldsymbol{V}$, which maps directly into the truthful subspace. We extract the column vectors from $\boldsymbol{V}$ corresponding to the top-k largest singular values to serve as the principal component projection directions, denoted as $\boldsymbol{V}^{1:k} \in \mathbb{R}^{d_F \times k}$, for subspace mapping:

$$\boldsymbol{V}^{1:k} = \boldsymbol{V}_{:,:k} = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k]. \tag{6}$$

It is precisely through these dominant directional channels that latent information propagates and co-varies from the initial or erroneous states into the target truthful state.

3.3 Low-Rank Representation Editing

Having designated the $\boldsymbol{V}_{\text{bp}}^{1:k}, \boldsymbol{V}_{\text{np}}^{1:k} \in \mathbb{R}^{d_F \times k}$ as the projection bases, this section delineates the training of representation editing vectors designed to improve model truthfulness while safeguarding intrinsic capabilities. We freeze all the pre-trained weights of the LLM and add learnable parameters $\boldsymbol{e}_{\text{bp}}$ and $\boldsymbol{e}_{\text{np}}$ so that during the forward pass, the representation $\boldsymbol{h}$ will have an additional offset vector in the low-dimensional subspace spanned by $\boldsymbol{V}_{\text{bp}}^{1:k}$ and $\boldsymbol{V}_{\text{np}}^{1:k}$.

Low-dimensional space projection. During the forward pass, let $\boldsymbol{h} \in \mathbb{R}^{d_F}$ denote the intermediate hidden representation within the l-th FFN layer, where the layer index is omitted for notational brevity. Firstly, we project the original feature $\boldsymbol{h}$ onto the two subspaces $\boldsymbol{z}_{\text{bp}}, \boldsymbol{z}_{\text{np}} \in \mathbb{R}^k$:

$$\boldsymbol{z}_{\text{bp}} = \boldsymbol{h}\boldsymbol{V}_{\text{bp}}^{1:k}, \quad \boldsymbol{z}_{\text{np}} = \boldsymbol{h}\boldsymbol{V}_{\text{np}}^{1:k}. \tag{7}$$

Where $\boldsymbol{z}_{\text{bp}}$ and $\boldsymbol{z}_{\text{np}}$ are the core components of the original hidden states along the k directions of maximal variance.

Latent-Space Intervention. We introduce trainable editing vectors into the low-dimensional bi-subspaces to intervene the latent representation, yielding the modified coordinates $\boldsymbol{z}'_{\text{bp}}$ and $\boldsymbol{z}'_{\text{np}}$:

$$\boldsymbol{z}'_{\text{bp}} = \boldsymbol{z}_{\text{bp}} + \boldsymbol{e}_{\text{bp}}, \quad \boldsymbol{z}'_{\text{np}} = \boldsymbol{z}_{\text{np}} + \boldsymbol{e}_{\text{np}}. \tag{8}$$

Here, $\boldsymbol{e}_{\text{bp}} \in \mathbb{R}^k$ is incorporated to inject missing factual logic, whereas $\boldsymbol{e}_{\text{np}} \in \mathbb{R}^k$ aims to suppress erroneous biases. Although these two subspaces may partially overlap in the high-dimensional original space, jointly optimizing $\boldsymbol{e}_{\text{bp}}$ and $\boldsymbol{e}_{\text{np}}$ during training enables the model to adaptively balance the trade-offs between truth enhancement and bias elimination.

High-Dimensional Reconstruction and Subspace Substitution. To preserve the model's intrinsic language capabilities, it is imperative to fully safeguard the original information contained within the orthogonal complement of the target subspaces. Thus, the original feature $\boldsymbol{h}$ can be decomposed into the sum of its projection in the principal subspace and its residual in the orthogonal space:

$$\boldsymbol{h} = \boldsymbol{h}_{\text{ortho}} + \boldsymbol{h}\boldsymbol{V}_{\text{bp}}^{1:k}(\boldsymbol{V}_{\text{bp}}^{1:k})^T. \tag{9}$$

During feature reconstruction, the orthogonal components $\boldsymbol{h}_{\text{ortho}}$ remain unchanged, and we replace the original principal subspace projection with the features after incorporating the edit vectors. Taking the bp subspace as an exemplar, the edited hidden representation $\hat{\boldsymbol{h}}_{\text{bp}}$ is formulated as:

$$\hat{\boldsymbol{h}}_{\text{bp}} = \boldsymbol{h}_{\text{ortho}} + \boldsymbol{z}'_{\text{bp}}(\boldsymbol{V}_{\text{bp}}^{1:k})^T. \tag{10}$$

Substituting $\boldsymbol{h}_{\text{ortho}} = \boldsymbol{h} - \boldsymbol{h}\boldsymbol{V}_{\text{bp}}^{1:k}(\boldsymbol{V}_{\text{bp}}^{1:k})^T$ and $\boldsymbol{z}'_{\text{bp}} = \boldsymbol{h}\boldsymbol{V}_{\text{bp}}^{1:k} + \boldsymbol{e}_{\text{bp}}$ into the equation yields the following expansion:

$$\hat{\boldsymbol{h}}_{\text{bp}} = \left(\boldsymbol{h} - \boldsymbol{h}\boldsymbol{V}_{\text{bp}}^{1:k}(\boldsymbol{V}_{\text{bp}}^{1:k})^T\right) + \left(\boldsymbol{h}\boldsymbol{V}_{\text{bp}}^{1:k} + \boldsymbol{e}_{\text{bp}}\right)(\boldsymbol{V}_{\text{bp}}^{1:k})^T$$

In this expanded formulation, the projection terms $\boldsymbol{h}\boldsymbol{V}_{\text{bp}}^{1:k}(\boldsymbol{V}_{\text{bp}}^{1:k})^T$ naturally cancel each other out. Extending and linearly superimposing this decomposition-substitution logic across both the bp and np subspaces simplifies the entire representation intervention process into an exceptionally efficient residual injection operation:

$$\hat{\boldsymbol{h}} = \boldsymbol{h} + e_{\text{bp}}(\boldsymbol{V}_{\text{bp}}^{1:k})^T + e_{\text{np}}(\boldsymbol{V}_{\text{np}}^{1:k})^T. \tag{11}$$

This simplification demonstrates that when the editing vectors $\boldsymbol{e}$ are parameterized as learnable weights independent of $\boldsymbol{h}$, the forward pass bypasses the explicit computation of the latent projection $\boldsymbol{z}$, thereby drastically mitigating computational overhead during model inference.

Training and Inference. During training, we learn $\boldsymbol{e}_{\text{bp}}$ and $\boldsymbol{e}_{\text{np}}$ on the same training data used to construct $\boldsymbol{V}_{\text{bp}}$ and $\boldsymbol{V}_{\text{np}}$. To further promote sparse modifications and improve the interpretability of the learned editing directions, we incorporate an L_1 penalty into loss:

$$\mathcal{L} = \mathcal{L}_{\text{task}} + \lambda \sum (\|\boldsymbol{e}_{np}\|_1 + \|\boldsymbol{e}_{bp}\|_1), \tag{12}$$

where λ determines the contribution of the regularization term and controls the degree of sparsity imposed on the editing vectors. During inference, $\boldsymbol{e}_{\text{bp}}$ and $\boldsymbol{e}_{\text{np}}$ are added to the intermediate hidden representation of the FFN in the same manner as the forward pass during training (Eq. 11). We clarify that BRIDGE is an offline alignment procedure rather than an online dynamic computation. During the offline training phase, cross-covariance SVD identifies the maximum covariance channels across representation distribution. Once trained, editing vector is cached directly into the hidden state space. Consequently, at inference time, BRIDGE operates as a zero-overhead residual addition, eliminating the need for real-time input conditioning while retaining the global geometric alignment discovered during calibration.

4 Experiments

4.1 Experimental Setup

Datasets. TruthfulQA (Lin et al., 2021) and FACTOR (expert) (Muhlgay et al., 2023) datasets are used for question answering tasks. CLUTRR (Sinha et al., 2019) and MQuAKE (Zhong et al., 2023) are used for multi-hop and counterfactual reasoning tasks.

- **TruthfulQA** is used to identify and measure LLM authenticity and contains 817 questions in 38 different categories. Our experiments are based on a multiple-choice discrimination task. We split the data into 326/82/407 train/dev/test examples, respectively, and apply two-fold cross-validation. Performance is measured using MC1, MC2 and MC3 accuracy.

Table 1: Results on Llama2-7B, Llama3-8B-Instruct and Mistral-7B-Instruct (v0.2 and v0.3) across TruthfulQA and FACTOR (expert) multiple-choice tasks. Results are averaged over 5 random seeds and the best results are bolded.

Methods	TruthfulQA			FACTOR	TruthfulQA			FACTOR
	MC1	MC2	MC3	ACC	MC1	MC2	MC3	ACC
Llama	**Llama2-7B**				**Llama3-8B-Instruct**			
Base	28.52	43.40	20.72	63.14	31.09	49.69	23.73	63.14
ITI	33.40	49.50	26.30	51.69	43.39	61.53	33.94	52.76
DoLa	24.94	45.37	21.92	67.30	42.96	65.76	35.71	64.06
SEA	29.87	46.55	22.53	84.75	41.13	59.87	32.49	82.63
TruthX	50.67	70.94	45.98	65.25	–	–	–	–
LOFIT	58.10	75.80	47.80	84.32	65.85	81.59	54.67	83.47
ACT	30.64	46.73	23.96	65.72	34.32	51.88	26.77	66.14
ODESTEER	32.93	49.34	24.45	66.45	37.33	54.97	27.38	68.56
PIXEL	27.31	42.75	22.28	62.50	46.30	60.28	35.49	64.71
PLI	40.88	70.17	41.60	71.11	63.35	79.22	**59.47**	65.87
BRIDGE	**62.43**$_{\pm 1.82}$	**79.51**$_{\pm 1.15}$	**52.64**$_{\pm 0.65}$	**87.29**$_{\pm 1.91}$	**67.69**$_{\pm 1.23}$	**82.63**$_{\pm 1.78}$	56.74$_{\pm 1.05}$	**85.17**$_{\pm 2.52}$
Mistral	**Mistral-7B-Instruct-v0.2**				**Mistral-7B-Instruct-v0.3**			
Base	55.20	70.94	42.54	63.98	48.72	66.29	37.52	67.37
ITI	55.12	72.30	44.87	59.82	46.27	63.16	35.29	60.79
DoLa	52.19	77.85	48.21	66.70	39.53	68.43	36.05	66.95
SEA	53.24	71.00	43.48	83.86	53.12	70.05	41.43	85.59
TruthX	56.67	74.90	45.99	64.89	–	–	–	–
LOFIT	69.16	82.89	59.90	83.90	70.01	84.10	59.80	82.63
ODESTEER	54.47	70.85	41.39	67.89	55.70	74.51	43.23	70.23
PIXEL	43.98	56.35	32.81	64.71	33.80	50.65	28.37	68.52
PLI	65.56	80.49	58.97	68.52	–	–	–	–
BRIDGE	**74.02**$_{\pm 0.94}$	**85.12**$_{\pm 1.28}$	**63.01**$_{\pm 1.73}$	**84.32**$_{\pm 1.86}$	**71.48**$_{\pm 1.12}$	**86.55**$_{\pm 1.61}$	**63.26**$_{\pm 1.34}$	**85.59**$_{\pm 2.49}$

- **FACTOR (Expert)** focused on text completion, challenges the model to identify the factually correct completion among several non-factual statements given a prefix. We split the dataset into 94/24/137 train/dev/test examples and report accuracy on the test set.
- **CLUTRR** is a deductive reasoning dataset that requires multi-hop reasoning over family relationship chains. It is commonly used to assess the logical consistency and reasoning depth of models. We use the subset containing 2-hop relational reasoning questions and randomly split it into 300/450/450 train/dev/test QA pairs. We report EM on the test set.
- **MQuAKE** evaluates knowledge editing and counterfactual reasoning by testing whether models can apply edited knowledge to infer indirectly related facts. We focus on a subset of 2-hop knowledge propagation instances. We split it into 134/95/864 train/dev/test examples and report EM accuracy on the test set.

Baselines Methods & LLMs. We first compare BRIDGE with methods that enhance the factuality of LLMs, including **ITI** (Li et al., 2023a), **DoLa** (Chuang et al., 2024), **SEA** (Qiu et al., 2024), **TruthX** (Zhang et al., 2024), **LOFIT** (Yin et al., 2024), **ACT** (Wang et al., 2025), **ODESTEER** (Zhao et al., 2026), **PIXEL** (Yu et al., 2026) and **PLI** (Chen et al., 2025). We also compared BRIDGE with existing PEFT methods: **RED** (Wu et al., 2024a), **LoRA** (Hu et al., 2022), **ReFT** (Wu et al., 2024b). Our experiments are conducted on a variety of open-source models, including LLaMA2-7B (Touvron et al., 2023), LLaMA3-8B-Instruct (AI@Meta, 2024), Mistral-7B-Instruct (v0.2 and v0.3) (Jiang et al., 2023).

4.2 Results on Question Answering Task

We first input the positive and negative samples from the training set (we did not use the test dataset to prevent data leakage) into the baseline model and edit the FFN intermediate representation of the 9-15 layers. Then, we construct the Target-Alignment Cross-Covariance Matrix C_{bp} and the De-biasing Cross-Covariance Matrix C_{np} based on these representations. Next, we perform SVD on these two matrices and extract the first K principal components to construct the basis of the Bi-subspace. Finally, we freeze the pre-trained parameters of the LLMs and train two extremely low-dimensional editing vectors e_{bp} and e_{np} according to Eq.(11), and inject them into the representations. For Llama2-7B, Llama3-8B-Instruct, Mistral-7B-Instruct-v0.2, and Mistral-7B-Instruct-v0.3, we select K as 500, 600, 90, and 700 respectively. More training parameters are available in

Table 2: Results of BRIDGE and PEFT methods on Llama2-7B across TruthfulQA, MQUAKE and CLUTRR datasets. Results are averaged over 5 random seeds and the best results are bolded.

Methods	TruthfulQA				MQUAKE		CLUTRR	
	Params	MC1	MC2	MC3	Params	EM	Params	EM
Base	-	28.52	43.40	20.72	-	25.12	-	64.70
RED	229K / 0.003%	56.32	76.81	49.26	229K / 0.003%	75.91	229K / 0.003%	89.12
LoRA	3.21M / 0.04%	52.64	71.48	46.34	3.21M / 0.04%	74.96	3.21M / 0.04%	90.08
ReFT	2.10M / 0.03%	48.58	68.73	43.77	2.10M / 0.03%	**77.17**	2.10M / 0.03%	87.64
TruthX	21M / 0.3%	50.67	70.94	45.98	-	-	-	-
LOFIT	143K / 0.002%	58.10	75.80	47.80	143K / 0.002%	73.68	143K / 0.002%	**91.96**
BRIDGE	**7K / 0.0001%**	**62.43±1.82**	**79.51±1.15**	**52.64±0.65**	**17.6K / 0.0002%**	76.16±0.88	**5K / 0.00007%**	91.78±0.52

the appendix. Table 1 presents the comparison results of BRIDGE with methods for enhancing the truthfulness of LLMs on the TruthfulQA and FACTOR (Expert) datasets.

This experiment evaluated the performance of the proposed method BRIDGE on the TruthfulQA and FACTOR (expert) datasets using Llama2-7B, Llama3-8B-Instruct, and Mistral-7B-Instruct (v0.2 and v0.3). Experimental results show that BRIDGE achieves best performance across most test models and evaluation metrics. Specifically, on Llama2-7B, BRIDGE significantly improved MC1, MC2, MC3, and FACTOR ACC from 28.52%, 43.40%, 20.72%, and 63.14% of the base model to 62.43%, 79.51%, 52.64%, and 87.29%, respectively. On Llama3-8B-Instruct, although PLI's MC3 (59.47%) was slightly higher than BRIDGE's (56.74%), BRIDGE still maintained an absolute lead in MC1 (67.69%), MC2 (82.63%), and FACTOR ACC (85.17%). On Mistral-7B (v0.3 and v0.2), BRIDGE also achieved the most significant performance leap across all metrics. While ITI improved the accuracy of TruthfulQA, it exhibited significant performance degradation on the FACTOR task (e.g., from 63.14% to 51.69% on Llama2-7B). DoLa and PIXEL also showed limited or even lower realism improvements on some models compared to the Base model. This comparison demonstrates the superiority of BRIDGE by constructing a Target-Alignment and De-biasing Bi-subspace and performing low-rank residual injection.

4.3 Comparison with PEFT Methods

We also compared BRIDGE with existing PEFT methods **RED**, **LoRA**, **ReFT**, **TruthX**, and **LOFIT**. We conducted experiments on question answering task, multi-hop reasoning and counterfactual reasoning tasks to validate the effectiveness of our method. Our experimental setup is consistent with LOFIT. Details of the experimental parameter settings are provided in the appendix.

Table 2 compares the parameter count and multi-task fine-tuning performance of BRIDGE and PEFT methods on the Llama2-7B model. More experimental results are available in the appendix. The experimental results show that BRIDGE achieves the best overall performance while exhibiting extreme parameter efficiency. In terms of parameter consumption, BRIDGE only needs to optimize 7K parameters (0.0001% of the total parameters) on TruthfulQA, 17.6K (0.0002%) on MQuAKE, and 5K (0.00007%) on CLUTRR. Compared to TruthX (21M) and LOFIT (143K), BRIDGE successfully reduces the number of trainable parameters by approximately 1200x–4200x and 8x–28x respectively, demonstrating a lightweight advantage. In terms of performance, BRIDGE achieved the best results in TruthfulQA on MC1 (62.43%), MC2 (79.51%), and MC3 (52.64%). On the MQuAKE and CLUTRR datasets, BRIDGE achieved accuracy (EM) of 76.16% and 91.78%, respectively, only slightly lower than ReFT (77.17%) and LOFIT (91.96%). This demonstrates that BRIDGE's low-rank bi-subspace design not only improves the truthfulness of LLMs in an extremely sparse parameter space but also perfectly balances the generalization ability and robustness of large models in logical reasoning tasks.

4.4 Analyses

We conducted extensive analytical experiments to investigate the advantages and improvements of BRIDGE. Unless otherwise stated, all our analytical experiments were performed using Llama2-7B on the TruthfulQA dataset. More experiments are available in the appendix.

Ablation Study. Table 3 presents the ablation experiment results of BRIDGE editing in bi-subspace, aiming to evaluate the independent and synergistic contributions of e_{bp} and e_{np} to improving the

Table 3: Ablation study of different residual injection components. Here, $+e_{bp}$ and $+e_{np}$ represent the editing vectors introduced into the Target-Alignment and De-biasing subspaces, respectively, where k denotes the top-k latent selection dimension. The best results are bolded.

Dataset	Metric	Base	$k = 500$		$k = 1000$		$k = 500$
			$+e_{bp}$	$+e_{np}$	$+e_{bp}$	$+e_{np}$	$+e_{bp} + e_{np}$
TruthfulQA	MC1	28.52	46.51	51.53	62.18	60.71	**62.43**
	MC2	43.40	65.75	69.96	79.40	78.70	**79.51**
	MC3	20.72	36.55	41.53	52.44	52.28	**52.64**

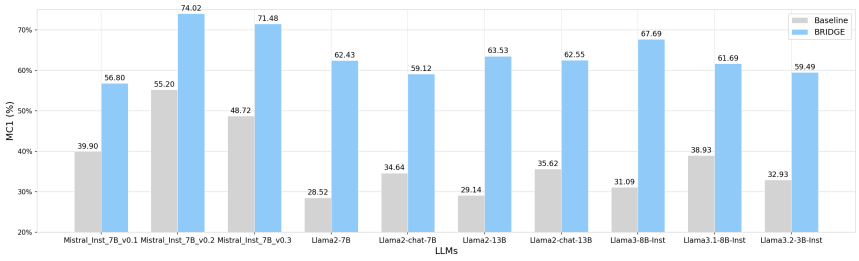
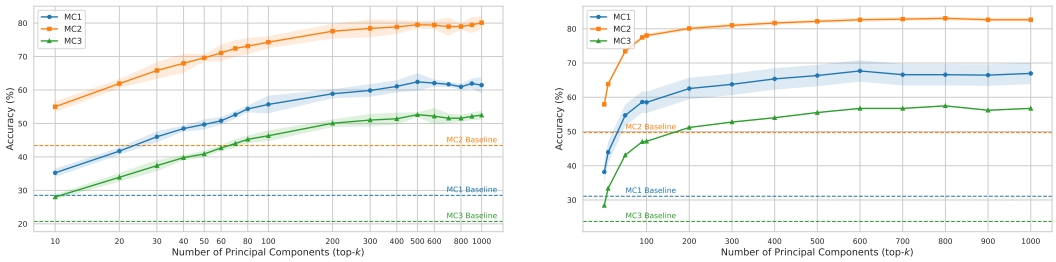


Figure 3: Improvements of BRIDGE on TruthfulQA (MC1 %) across 10 different base models.

factuality of the model. Compared to the base model, injecting edit vectors into either subspace individually yields a significant improvement in factual accuracy. When both components are injected simultaneously (i.e., the BRIDGE framework), the model achieves a leap in optimal performance, with MC1, MC2, and MC3 surging to 62.43%, 79.51%, and 52.64%, respectively. This ablation experiment strongly demonstrates the necessity and complementarity of the bi-subspace design in the BRIDGE framework. Injecting correct fact guidance (e_{bp}) and eliminating erroneous semantic biases (e_{np}) are not mutually exclusive; their combination produces a powerful synergistic effect, achieving maximum fact alignment within ultra-low-dimensional subspaces.

Testing on More Base Models. Figure 3 shows the MC1 accuracy comparison between BRIDGE and the uninterrupted baseline model on 10 open-source LLMs with different architectures, parameter sizes, and alignment states. Extensive experimental evidence suggests that BRIDGE does not rely on any particular model structure or the benefits of fine-tuning distributions. The intervention effect of BRIDGE does not weaken with changes in the number of model parameters, demonstrating excellent scaling performance. By capturing general semantic trajectories in bi-subspaces and injecting low-rank residuals into the underlying logic, it successfully addresses the fundamental bottleneck of storing and transferring factual knowledge in the Transformer architecture, making it a highly general, robust, and effective representation editing solution.

Parameter Sensitivity of Top-k. Figure 4 illustrates the parameter sensitivity analysis of the number of principal components (k) in the range [10, 1000], compared with the baseline. The results show that even in the extremely low-dimensional configuration of $k = 10$, BRIDGE exhibits low-rank effectiveness, significantly outperforming the baseline on all three metrics. This rapid performance improvement validates that the intrinsic paths controlling fact activation in LLMs are highly concentrated in a low-dimensional subspace spanned by the principal singular vectors. As k expands from 10 to 500 (and up to 600), accuracy across MC1, MC2, and MC3 increases monotonically and robustly, reaching a global performance peak around $k = 500 \sim 600$. However, after expanding the dimension from this optimal threshold, the gains diminish, even showing a slight decrease. This behavior aligns with spectral representation theory: singular vectors associated with smaller singular values primarily capture generic background variance or task-irrelevant redundant features. Excessively increasing the value of k not only fails to provide additional factual gain but also injects this high-frequency noise into the model as residuals, unnecessarily perturbing the original general semantic space. The shaded band is the standard deviations of multiple random seeds, remains narrow throughout the entire k range. This demonstrates that BRIDGE's SVD-based spectral discovery mechanism reliably separates deterministic, truth-related directions and exhibits strong robustness to data sampling volatility and training randomness.



(a) Llama2-7B (b) Llama3-8B-Instruct

Figure 4: Sensitivity analysis of the number of principal components (k) on Llama2-7B and Llama3-8B-Instruct. The solid lines indicate the mean accuracy of MC1, MC2, and MC3 scores, while the shaded bands represent the variance across multiple random seeds.

5 Related Work

Inference-Time Hallucination Mitigation. A growing body of research indicates that intervention during LLM inference can effectively alleviate hallucinations (Li et al., 2026; Hu et al., 2024; Wang et al., 2025; Cheng et al., 2025; Zhou et al., 2026). CD (Li et al., 2023b) corrects output probabilities by comparing strong and weak models. ICD (Zhang et al., 2025) improves LLM accuracy by using contrastive decoding with weak models that incorporate illusions. DoLa (Chuang et al., 2024) improves realism by comparing the differences in logits obtained by projecting later layers onto the lexical space with those of earlier layers. PLI (Chen et al., 2025) mitigates hallucinations by inserting premature layers formed by mathematical interpolation of adjacent layers. ITI (Li et al., 2023a) improves output factuality by probing and adjusting truth-related attention head representations. TruthX (Zhang et al., 2024) employs an autoencoder to map LLM representations to semantic and real latent spaces, respectively, and applies contrastive learning to identify true edit directions in the real space. PIXEL (Yu et al., 2026) learns attribute-aligned subspaces from dual views and selects intervention strengths through constrained geometric objectives with closed solutions. ODESTEER (Zhao et al., 2026) achieves multi-step adaptive steering by numerically solving an ordinary differential equation (ODE).

Representation editing. Representation editing methods for LLMs enable precise and efficient knowledge correction or behavior modulation by directly intervening in internal representations, without modifying the original parameters (Zou et al., 2023; Liu et al., 2023; Huang et al., 2025a). SEA (Qiu et al., 2024) is edited to maintain a high correlation with positive behaviors and decorrelate with negative behaviors. RED (Wu et al., 2024a) fine-tunes the scaling factor and bias of the LLM hidden representation to influence the model's behavior. ReFT (Wu et al., 2024b) trains the projection matrix and bias, and then maps the representation to a low-rank linear subspace for fine-tuning. LOFIT (Yin et al., 2024) uses scaling factors to locate relevant attention heads and inserts biases into the representations of highly relevant attention heads. However, these methods still require optimization of a large number of parameters and suffer from the drawback of "distribution-blind", resulting in significant computational overhead and performance degradation.

6 Conclusion

In this paper, we introduced BRIDGE, a highly efficient residual steering framework that uses base-conditioned subspace construction to learn contribution-informed editing directions. By constructing Target-Alignment and De-biasing Cross-Covariance matrices, BRIDGE leverages SVD to isolate principal truthful subspaces, mathematically streamlining the intervention into an fast residual injection that replacing the costly projection-reconstruction operation. Extensive experiments across various models and benchmarks demonstrate that BRIDGE enhances factual accuracy while reducing learnable parameters by 20x to 3000x. BRIDGE balances robust capability steering with the preservation of the model's intrinsic language capacity, offering a scalable and lightweight solution for representation editing.

REFERENCES

AI@Meta. Llama 3 model card. 2024. URL https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md.

Dingwei Chen, Ziqiang Liu, Feiteng Fang, Chak Tou Leong, Shiwen Ni, Ahmadreza Argha, Hamid Alinejad-Rokny, Min Yang, and Chengming Li. Expanding before inferring: Enhancing factuality in large language models through premature layers interpolation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 12781–12796, 2025.

Yi Cheng, Xiao Liang, Yeyun Gong, Wen Xiao, Song Wang, Yuji Zhang, Wenjun Hou, Kaishuai Xu, Wenge Liu, Wenjie Li, et al. Integrative decoding: Improving factuality via implicit self-consistency. In *International Conference on Learning Representations*, volume 2025, pp. 69573–69618, 2025.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James R Glass, and Pengcheng He. Dola: Decoding by contrasting layers improves factuality in large language models. In *International Conference on Learning Representations*, volume 2024, pp. 54158–54183, 2024.

Andrew M Dai and Quoc V Le. Semi-supervised sequence learning. *Advances in Neural Information Processing Systems*, 28, 2015.

Atticus Geiger, Zhengxuan Wu, Christopher Potts, Thomas Icard, and Noah Goodman. Finding alignments between interpretable causal variables and distributed neural representations. In *Causal Learning and Reasoning*, pp. 160–187. PMLR, 2024.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Minda Hu, Bowei He, Yufei Wang, Liangyou Li, Chen Ma, and Irwin King. Mitigating large language model hallucination with faithful finetuning. *arXiv preprint arXiv:2406.11267*, 2024.

Baixiang Huang, Canyu Chen, Xiongxiao Xu, Ali Payani, and Kai Shu. Can knowledge editing really correct hallucinations? In *International Conference on Learning Representations*, volume 2025, pp. 88116–88149, 2025a.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55, 2025b.

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

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.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, 36:41451–41530, 2023a.

Shike Li, Xiaokai Wang, Xiaofeng Liu, Xin Tong, and Hu Zhang. Cofact: Dynamic coordination of attention heads for improving factual consistency in llms. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 31708–31716, 2026.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori B Hashimoto, Luke Zettlemoyer, and Mike Lewis. Contrastive decoding: Open-ended text generation as optimization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (volume 1: Long papers)*, pp. 12286–12312, 2023b.

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

Sheng Liu, Haotian Ye, Lei Xing, and James Zou. In-context vectors: Making in context learning more effective and controllable through latent space steering. *arXiv preprint arXiv:2311.06668*, 2023.

Dor Muhlgay, Ori Ram, Inbal Magar, Yoav Levine, Nir Ratner, Yonatan Belinkov, Omri Abend, Kevin Leyton-Brown, Amnon Shashua, and Yoav Shoham. Generating benchmarks for factuality evaluation of language models. *arXiv preprint arXiv:2307.06908*, 2023.

Yifu Qiu, Zheng Zhao, Yftah Ziser, Anna Korhonen, Edoardo.M. Ponti, and Shay.B. Cohen. Spectral editing of activations for large language model alignment. *Advances in Neural Information Processing Systems*, 37:56958–56987, 2024.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. Retrieval augmentation reduces hallucination in conversation. *arXiv preprint arXiv:2104.07567*, 2021.

Koustuv Sinha, Shagun Sodhani, Jin Dong, Joelle Pineau, and William L Hamilton. Clutrr: A diagnostic benchmark for inductive reasoning from text. *arXiv preprint arXiv:1908.06177*, 2019.

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.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering, 2024. URL https://arxiv.org/abs/2308.10248.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

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.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pp. 38–45, 2020.

Muling Wu, Wenhao Liu, Xiaohua Wang, Tianlong Li, Changze Lv, Zixuan Ling, Zhu JianHao, Cenyuan Zhang, Xiaoqing Zheng, and Xuan-Jing Huang. Advancing parameter efficiency in fine-tuning via representation editing. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13445–13464, 2024a.

Tianyu Wu, Shizhu He, Jingping Liu, Siqi Sun, Kang Liu, Qing-Long Han, and Yang Tang. A brief overview of chatgpt: The history, status quo and potential future development. *IEEE/CAA Journal of Automatica Sinica*, 10(5):1122–1136, 2023.

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.

Fangcong Yin, Xi Ye, and Greg Durrett. Lofit: Localized fine-tuning on llm representations. *Advances in Neural Information Processing Systems*, 37:9474–9506, 2024.

Manjiang Yu, Hongji Li, Priyanka Singh, Xue Li, Di Wang, and Lijie Hu. Pixel: Adaptive steering via position-wise injection with exact estimated levels under a subspace calibration. In *Proceedings of the ACM Web Conference 2026*, pp. 1574–1585, 2026.

Shaolei Zhang, Tian Yu, and Yang Feng. TruthX: Alleviating hallucinations by editing large language models in truthful space. 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. 8908–8949, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.483. URL https://aclanthology.org/2024.acl-long.483/.

Yue Zhang, Leyang Cui, Wei Bi, and Shuming Shi. Alleviating hallucinations of large language models through induced hallucinations. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 8233–8247, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-195-7. doi: 10.18653/v1/2025.findings-naacl.459. URL https://aclanthology.org/2025.findings-naacl.459/.

Hongjue Zhao, Haosen Sun, Jiangtao Kong, Xiaochang Li, Qineng Wang, Liwei Jiang, Qi Zhu, Tarek Abdelzaher, Yejin Choi, Manling Li, et al. Odesteer: A unified ode-based steering framework for llm alignment. In *International Conference on Learning Representations*, 2026.

Zexuan Zhong, Zhengxuan Wu, Christopher D Manning, Christopher Potts, and Danqi Chen. Mquake: Assessing knowledge editing in language models via multi-hop questions. *arXiv preprint arXiv:2305.14795*, 2023.

Xiaoling Zhou, Mingjie Zhang, Zhemg Lee, Yuncheng Hua, Wei Ye, Flora Salim, Shikun Zhang, et al. Boosting resilience of large language models through causality-driven robust optimization. *Advances in Neural Information Processing Systems*, 38:55821–55845, 2026.

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.

7 APPENDIX

7.1 TRUTHFULQA EVALUATION

Metrics. TruthfulQA (Lin et al., 2021) adopts a multiple-choice evaluation protocol to investigate the extent to which language models can distinguish factual answers from misleading alternatives. In this benchmark, the distractor options are intentionally designed based on widespread misconceptions and erroneous beliefs commonly held by humans, making the task challenging beyond simple factual retrieval. Therefore, the evaluation reflects not only a model's capability to access stored knowledge but also its tendency to avoid producing answers that appear credible while lacking factual correctness. Following prior work, we employ the MC1, MC2, and MC3 metrics as quantitative measures of model truthfulness. The definition of each score is as follows:

- **MC1:** Among the set of true and false reference answers, we need to choose the best correct answer. MC1 is computed by whether the language model assigns the highest likelihood to the best correct answer over the false answers given the question.
- **MC2:** MC2 is the total normalized probability of the true reference answers. The score is the probability mass for correct answers.
- **MC3:** MC3 is computed by whether the language model assigns a higher likelihood to the correct answers over the false answers given the question.

Prompt Templates. We follow the prompt strategy in ITI (Li et al., 2023a): at the fine-tuning steps, we simply concatenate the question with the gold response or the incorrect response in the preference pair of the training and validation data as: *Q:* {*Question*} *A:* {*Response*}.

Details on Spectral Bases Construction. To address concerns regarding sample sizes and potential rank deficiency when constructing the cross-covariance matrices, we present the detailed formulation of the contrastive triplet dataset $\mathcal{D}$ extracted from TruthfulQA. TruthfulQA consists of 326 training questions. However, each question Q_i ($i \in \{1, \ldots, 326\}$) naturally contains multiple reference truthful answers $\{A^{(m)}_{pos,i}\}_{m=1}^{M_i}$ and untruthful answers $\{A^{(n)}_{neg,i}\}_{n=1}^{N_i}$. Instead of restricting each question to a single pair, we systematically pair every positive answer $A^{(m)}_{pos,i}$ with every negative answer $A^{(n)}_{neg,i}$ under question Q_i to form candidate contrastive units $(Q_i, A^{(m)}_{pos,i}, A^{(n)}_{neg,i})$.

From this expanded pairwise candidate pool, we uniformly sample $N = 1,000$ distinct contrastive triplets to build $\mathcal{D} = \{(Q^{(j)}, A^{(j)}_{pos}, A^{(j)}_{neg})\}_{j=1}^{N}$. The baseline state is induced by querying the LLM with prompt $Q^{(j)}$, while the truthful and untruthful states are induced by prompting with $Q^{(j)} + A^{(j)}_{pos}$ and $Q^{(j)} + A^{(j)}_{neg}$, respectively. This process yields the layer-wise intermediate hidden representation matrices $H_{base}, H_{pos}, H_{neg} \in \mathbb{R}^{N \times d_F}$. Given $N = 1,000$ sampled triplets, the resulting cross-covariance matrices $\boldsymbol{C}_{bp} = \frac{1}{N}\boldsymbol{H}^T_{base}\boldsymbol{H}_{pos}$ and $\boldsymbol{C}_{np} = \frac{1}{N}\boldsymbol{H}^T_{neg}\boldsymbol{H}_{pos}$ have dimensions $\boldsymbol{d}_F \times \boldsymbol{d}_F$ (where $\boldsymbol{d}_F$ is the FFN intermediate dimension, e.g., $\boldsymbol{d}_F = 11,008$ for Llama2-7B). Algebraically, the maximum possible numerical rank of $\boldsymbol{C}_{bp}$ and $\boldsymbol{C}_{np}$ is bounded by $\min(N, \boldsymbol{d}_F) = 1,000$. Because the rank upper bound is $1,000$, all singular directions up to $k \leq 1,000$ in Figure 4 are mathematically well-defined and strictly identified from data.

Training Setup. We fine-tune BRIDGE and baselines using a single NVIDIA-RTX A800 GPU with 80G memory. We implement all optimization procedures using the Hugging Face Transformers library (Wolf et al., 2020) with PyTorch. For fine-tuning on the TruthfulQA benchmark, we adopt the TRL library and employ Direct Preference Optimization (DPO) (Rafailov et al., 2023) to align model outputs with human-preferred responses. In our setup, factual statements are treated as positive samples and non-factual statements as negative samples. We obtain the editing vectors through a parameter-efficient optimization process over a small collection of task-relevant examples. Specifically, the original parameters of the pre-trained language model remain fixed throughout training, while only the editing vectors e_{np} and e_{bp} are updated by minimizing DPO. The initial values of these learnable vectors are sampled from a distribution $N(0, 0.2 \times lr)$, where lr is learning rate. Parameter settings are shown in Table 4. The results for various base models are shown in Table 5.

Compare with PEFT Methods. Table 6 compares the parameter count and performance of BRIDGE and PEFT methods on the Llama2-13B model.

Table 4: Hyperparameters settings for different models for BRIDGE on TruthfulQA.

Models	λ	β	Learning Rate	Batch Size	Epoch	Top-K
Llama2-7B	1.00E-04	0.5	1.00E-02	8	10	500
Llama2-7B-Chat	1.00E-04	0.5	1.00E-02	8	10	500
Llama2-13B	1.00E-04	0.5	1.00E-02	8	10	500
Llama2-13B-Chat	1.00E-04	0.5	1.00E-02	8	10	500
Llama3-8B-Inst	1.00E-04	0.5	5.00E-03	8	10	600
Llama3.1-8B-Inst	1.00E-04	0.5	1.00E-02	8	10	500
Llama3.2-3B-Inst	1.00E-04	0.5	1.00E-02	8	10	500
Mistral-7B-v0.1-Inst	1.00E-04	0.5	5.00E-02	8	10	90
Mistral-7B-v0.2-Inst	1.00E-04	0.5	5.00E-02	8	10	90
Mistral-7B-v0.3-Inst	1.00E-04	0.5	1.00E-02	8	10	700

Table 5: The results for vary models on TruthfulQA.

Model	TruthfulQA		
	MC1	MC2	MC3
Llama2-7B	28.52%	43.40%	20.72%
BRIDGE	**62.43%**	**79.51%**	**52.64%**
Llama2-Chat-7B	34.64%	51.31%	25.10%
BRIDGE	**59.12%**	**77.65%**	**49.45%**
Llama2-13B	29.14%	44.32%	25.41%
BRIDGE	**63.53%**	**82.06%**	**55.69%**
Llama2-Chat-13B	35.62%	53.41%	26.83%
BRIDGE	**62.55%**	**80.09%**	**51.66%**
Llama3-8B-Inst	31.09%	49.69%	23.73%
BRIDGE	**67.69%**	**82.63%**	**56.74%**
Llama3.1-8B-Inst	38.93%	58.44%	30.63%
BRIDGE	**61.69%**	**79.85%**	**54.95%**
Llama3.2-3B-Inst	32.93%	53.57%	26.74%
BRIDGE	**59.49%**	**77.83%**	**51.23%**
Mistral-7B-v0.1-Inst	39.90%	56.45%	29.06%
BRIDGE	**56.80%**	**75.93%**	**48.23%**
Mistral-7B-v0.2-Inst	55.20%	70.94%	42.54%
BRIDGE	**74.02%**	**85.12%**	**63.01%**
Mistral-7B-v0.3-Inst	48.72%	66.29%	37.52%
BRIDGE	**71.48%**	**86.55%**	**63.26%**

Table 6: Results of BRIDGE and PEFT methods on Llama2-13B on TruthfulQA.

	Methods	Learned Params		TruthfulQA		
		# Params	% Params	MC1	MC2	MC3
Llama2-13B	Base	-	-	29.12	44.34	21.67
	RED	262K	0.004%	53.70	74.10	51.49
	LoRA	4.19M	0.06%	56.30	75.40	49.83
	ReFT	2.10M	0.03%	53.20	72.90	46.26
	LOFIT	175K	0.003%	57.00	76.80	50.36
	BRIDGE	9K	0.0001%	**63.53**	**82.06**	**55.69**

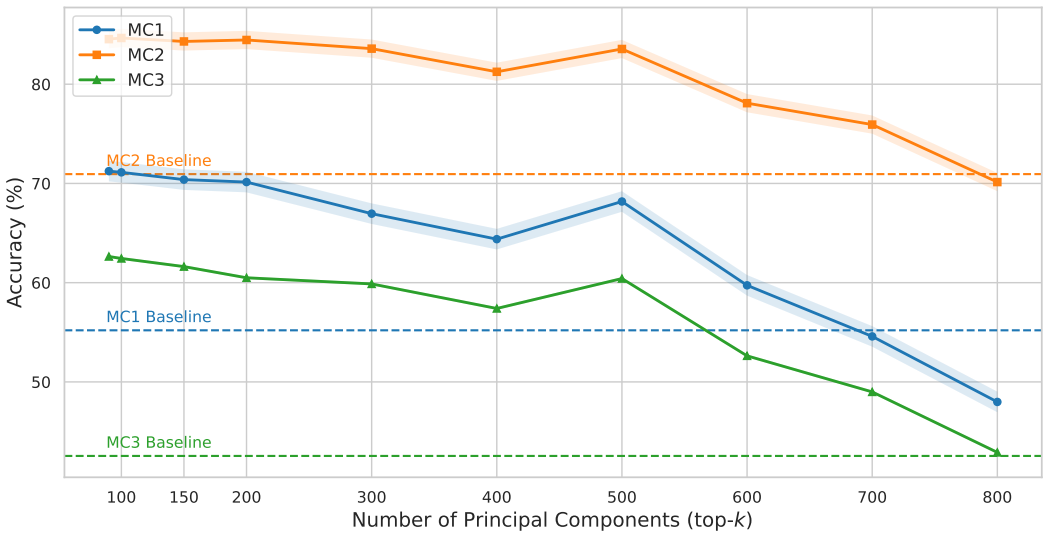


Figure 5: Top-K results of Mistral-Inst-7B-v0.2 on TruthfulQA.

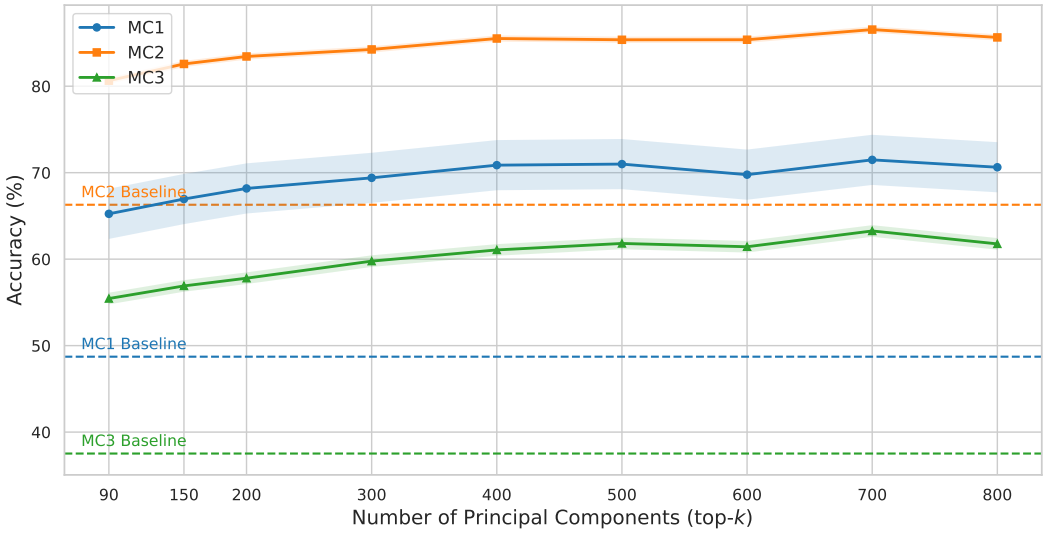


Figure 6: Top-K results of Mistral-Inst-7B-v0.3 on TruthfulQA.

Results of Vary Top-K. We conducted Top-K parameter analysis experiments on models such as Llama3-8B-Inst and Mistral-Inst-7B (v0.2, v0.3). The results are shown in the Figure 5, 6.

Ablation on Vary Models. Table 7 records the results of using different editing type on different base models.

7.2 FACTOR (EXPERT) EVALUATION

Metrics. FACTOR (Expert) dataset focuses on evaluating the factual accuracy of language models through a text completion task. Given a factual prefix (context), the model is challenged to identify the factually correct completion among several non-factual or hallucinated distractor statements. Unlike standard multi-choice QA where options are distinct answers, FACTOR assesses the model's intrinsic ability to discriminate subtle factual nuances within continuous text generation. Following the evaluation protocol of FACTOR, we adopt Accuracy (ACC) as the quantitative metric, which

Table 7: Ablation study of different residual injection components for vary models on TruthfulQA.

Editing Type	TruthfulQA		
	MC1	MC2	MC3
Llama2-7B	28.52%	43.40%	20.72%
e_{bp}	46.51%	65.75%	36.55%
e_{np}	51.53%	69.96%	41.53%
e_{bp}+e_{np}	**62.43%**	**79.51%**	**52.64%**
Llama3-8B-Inst	31.09%	49.69%	23.73%
e_{bp}	54.22%	72.14%	42.60%
e_{np}	57.78%	75.64%	44.89%
e_{bp}+e_{np}	**67.69%**	**82.63%**	**56.74%**
Mistral-7B-Instruct-v0.3	48.72%	66.29%	37.52%
e_{bp}	65.24%	81.02%	53.62%
e_{np}	67.08%	82.04%	55.81%
e_{bp}+e_{np}	**71.48%**	**86.55%**	**63.26%**
Mistral-7B-Instruct-v0.2	55.20%	70.94%	42.54%
e_{bp}	64.38%	79.36%	53.65%
e_{np}	67.08%	81.16%	56.83%
e_{bp}+e_{np}	**74.02%**	**85.12%**	**63.01%**

Table 8: Hyperparameters settings for different models for BRIDGE on FACTOR (Expert).

Models	λ	β	Learning Rate	Batch Size	Epoch	Top-K
Llama2-7B	0	0.5	5.00E-03	4	10	90
Llama3-8B-Inst	0	0.5	5.00E-03	4	10	90
Mistral-7B-v0.2-Inst	0	0.5	5.00E-03	4	10	90
Mistral-7B-v0.3-Inst	0	0.5	5.00E-03	4	10	90

Table 9: The results for different models for BRIDGE on MQuAKE.

Model	Llama2-7B	Llama3-8B-Inst	Mistral-7B-Instruct-v0.3	Mistral-7B-Instruct-v0.2
Base	25.12%	30.56%	12.73%	12.96%
BRIDGE	**76.16%**	**67.71%**	**76.50%**	**75.69%**

Table 10: Hyperparameters settings for different models for BRIDGE on MQuAKE.

Models	λ	β	Learning Rate	Batch Size	Epoch	Top-K
Llama2-7B	1.00E-04	0.5	5.00E-03	8	10	800
Llama3-8B-Inst	1.00E-04	0.5	1.00E-02	8	10	800
Mistral-7B-v0.2-Inst	1.00E-04	0.5	1.00E-02	8	10	600
Mistral-7B-v0.3-Inst	1.00E-04	0.5	5.00E-03	8	10	700

Table 11: The results for different models for BRIDGE on CLUTRR.

Model	Llama2-7B	Llama3-8B-Inst	Mistral-7B-Instruct-v0.3	Mistral-7B-Instruct-v0.2
Base	64.70%	76.00%	71.33%	67.33%
BRIDGE	**91.78%**	**89.78%**	**88.67%**	**89.78%**

Table 12: Hyperparameters settings for different models for BRIDGE on CLUTRR.

Models	λ	β	Learning Rate	Batch Size	Epoch	Top-K
Llama2-7B	1.00E-04	0.5	1.00E-01	8	10	500
Llama3-8B-Inst	1.00E-04	0.5	1.00E-01	8	10	400
Mistral-7B-v0.2-Inst	1.00E-04	0.5	5.00E-01	8	10	700
Mistral-7B-v0.3-Inst	1.00E-04	0.5	5.00E-01	8	10	500

measures whether the model assigns a higher probability or likelihood score to the true factual completion compared to the non-factual alternatives.

Prompt Templates. Our setup is the same as TruthfulQA: *Q:* {*Question*} *A:* {*Response*}.

Training Setup. We also used DPO for training and parameter settings are shown in the Table 8.

7.3 MQuAKE Evaluation

Prompt Templates. Each example in MQuAKE comes with a piece of edited knowledge and a multi-hop reasoning question that uses the edited knowledge. We used the following prompt for MQuAKE: *Q: Imagine that* {*edited knowledge*}. {*Question*} *A:* {*Answer*}. To extract the model's internal representation from the MQuAKE dataset, we constructed cue words for each data sample, representing both positive and negative samples, and extracted the hidden state.

- **Positive Prompt**: The correct answer (new answer) is appended to the end of the above Prompt, representing the complete context that aligns with the new knowledge.
- **Negative Prompt**: The original answer (answer) is appended to the end of the above Prompt, representing the context that is influenced by prior knowledge.

Training Setup. We use Supervised Fine-Tuning (SFT) and cross-entropy loss with gold responses for fine-tuning. We selected the intermediate representations of FFN from layers [2, 3, 7, 10, 11, 13, 14, 15, 26, 30, 31] for editing. Other settings are the same as TruthfulQA. Parameter settings are shown in the Table 10. The results are show in the Table 9.

7.4 CLUTRR Evaluation

Prompt Templates. Each example in CLUTRR comes with a few-sentence story that describes relationships between fictional characters, along with a target character pair whose relationship needs to be inferred from the story and answered by the model. We use the following prompt format for CLUTRR. In preliminary experiments, we found that large language models sometimes refuse to answer, stating that the relationships in the provided story are incorrect based on real-world knowledge of names and relationships. Therefore, in addition to the basic prompt format, we include further clarification instructions within the prompt: *Read the following story about a family.* {*Story*} *Assume the relations described in the story are all true. Based on relations between the fictional characters in the story (assume the relations are all true) and your commonsense knowledge about family relationship, how is* {*Character2*} *related to* {*Character1*} *? Answer:*{*Character2*} *is* {*Character1*} *'s* {*gold rel*}

- **Positive Prompt**: Uses a template to fill in the text story, maintaining the order that matches the target relationship (e.g., asking about the relationship between Per_2 and Per_1), and appends the correct relationship label $gold_rel$ to the end of the Prompt.
- **Negative Prompt**: Maintains the same target relationship label $gold_rel$, but reverses the entity query direction in the template question (e.g., changes the question to ask about the relationship between Per_1 and Per_2), thus constructing a negative sample with an inverted entity order and logical errors.

Training Setup. We use Supervised Fine-Tuning (SFT) and cross-entropy loss with gold responses for fine-tuning. We selected the intermediate representations of FFN from layers [2, 3, 6, 22, 31] for editing. Other settings are the same as TruthfulQA. Parameter settings are shown in the Table 12.