GRAINS: Gradient-based Attribution for Inference-Time Steering of LLMs and VLMs

Duy Nguyen[1] **Archiki Prasad**[1] **Elias Stengel-Eskin**[2] **Mohit Bansal**[1]
[1]UNC Chapel Hill [2]The University of Texas at Austin

Abstract

Inference-time steering provides a lightweight alternative to fine-tuning large language models (LLMs) and vision-language models (VLMs) by modifying model activations without updating weights. However, existing methods often rely on a global intervention vector, overlook token-level influence, and underutilize model logits, especially in multimodal settings where visual and textual inputs contribute unevenly. We propose GRAINS, a contrastive, gradient-based approach that leverages Integrated Gradients to identify top-k influential tokens and construct directional steering vectors based on their contribution to preferred over dispreferred outputs. These vectors guide activation intervention at each layer, preserving the representational scale. GRAINS outperforms fine-tuning and prior steering methods on both LLM and VLM tasks: improving TruthfulQA accuracy by 13.22% (Llama-3.1-8B), reducing MMHal-Bench hallucinations from 0.624 to 0.514 (LLaVA-1.6-7B), and increasing SPA-VL alignment by 8.11%, all without degrading fluency or general capabilities. [1]

1 Introduction

Despite having strong performance across various tasks, LLMs and VLMs often generate undesirable outputs that lack grounding in the input query or context (Rame et al., 2024; Shi et al., 2024; Huang et al., 2024). Fine-tuning addresses these issues by adapting models with task-specific datasets, but it requires significant computational resources and data, and risks catastrophic forgetting (Li and Hoiem, 2017; Lopez-Paz and Ranzato, 2017). A promising alternative to fine-tuning is inference-time steering (Zou et al., 2023; Liu et al., 2024c; Li et al., 2024b; Rimsky et al., 2024; Turner et al., 2024; Nguyen et al., 2025a), which adjusts hidden representations during inference without altering the model's parameters. However, existing steering approaches generally rely on linear interventions to hidden states, often applying the same intervention across all tokens' hidden states (Marks and Tegmark, 2023; Li et al., 2024b), ignoring the impact of specific tokens on model behavior. As illustrated in Fig. 1 (top), this can lead to over-correction and loss of desired capabilities, such as fluency or factual accuracy (Nguyen et al., 2025b). Moreover, most existing methods construct steering vectors solely from latent space representations of paired data by taking differences between hidden activations corresponding to desirable and undesirable outputs (Li et al., 2024b; Rimsky et al., 2024; Turner et al., 2024; Nguyen et al., 2025b), ignoring rich signals from model logits that reveal which specific inputs (tokens) most drive undesirable outputs through their *attribution-based contribution* to model predictions. In VLMs, this limitation is especially problematic – *textual and visual inputs do not contribute equally* – some tokens play a key role in shaping the model's output, while others have little to no influence (Cao et al., 2024; Sun et al., 2025; Lin et al., 2025). Thus, constructing steering vectors purely in latent space without identifying which tokens are responsible for undesirable behavior can be ineffective and may cause unintended changes to the model's behavior (Salin et al., 2022; Chen et al., 2024b).

To address these issues, we propose **Gr**adient-based **A**ttribution for **In**ference-Time **S**teering (GRAINS), a more selective and interpretable approach to inference-time steering compatible with both LLMs and VLMs, as outlined in Fig. 1 (bottom) and shown in more detail in Fig. 2. GRAINS identifies specific tokens—whether visual patches or language tokens—that have the greatest *attribution-based contribution* to the model's output, and applies *steering based on their contribution*. To measure this influence, we use Integrated Gradients (IG) (Sundararajan et al., 2017;

[1]Code:https://github.com/duykhuongnguyen/GrAInS.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 46523–46543
July 2-7, 2026

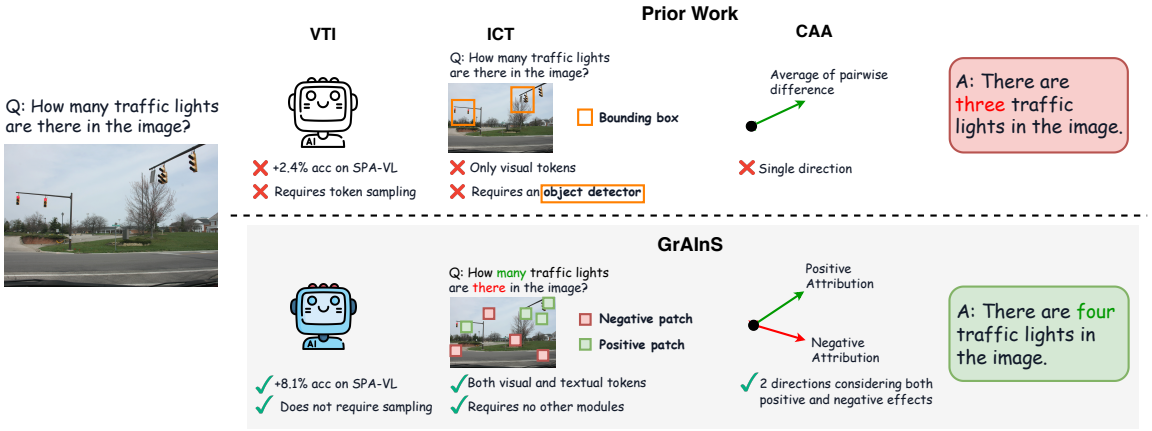

Figure 1: Comparison of prior steering methods vs. GRAINS, our attribution-guided approach on VLMs. Top: Existing methods suffer from some key limitations such as using only visual tokens, relying on external object detectors, or steering in a single fixed direction. Bottom: GRAINS leverages both visual and textual tokens using contrastive Integrated Gradients, requires no external modules, and constructs targeted, directional interventions based on positive and negative attribution, leading to improved factual accuracy.

Kapishnikov et al., 2021) over a contrastive loss between preferred and dispreferred outputs to compute token-level attributions (see Fig. 2(A)). Tokens with high positive attribution are those most responsible for producing desirable outputs, while those with strong negative attribution contribute to undesirable behaviors such as hallucinations or toxicity. We construct contrastive input variants by masking each token set separately and measuring changes in hidden activations. These capture how each group influences internal representations, and we apply Principal Component Analysis (PCA) to derive a steering vector that represents behavior shifts in latent space (see Fig. 2(B)). At inference, the steering vector is applied with normalization to preserve general model capabilities such as fluency and reasoning (see Fig. 2(C)). Unlike prior work that operates with a single steering direction (Rimsky et al., 2024), relies solely on visual tokens (Chen et al., 2024a), or requires token sampling (which can introduce instability or require large sample sizes, making them computationally expensive) and external modules (Liu et al., 2024b; Chen et al., 2024a), GRAINS integrates both visual and textual inputs, accounts for both positive and negative attribution directions, and introduces no additional components or supervision (see Fig. 1). Moreover, while prior work has largely limited attribution methods like IG to post-hoc explanation (Lin et al., 2025), we bridge the gap between interpretability and active model steering. This enables more precise, token-sensitive interventions, leading to improved alignment and interpretability in steering of both unimodal and multimodal LLMs.

We evaluate GRAINS on safety-critical tasks across both VLMs and LLMs, targeting hallucinations, bias, toxicity, and truthfulness, where it shows strong performance in both modalities (vision and language) without retraining. In VLMs, we achieve a hallucination rate reduction from 0.624 to 0.514 on LLaVA-1.6-7B and improve alignment preference win rates by 8.11% on SPA-VL, outperforming LoRA and multimodal steering methods such as VTI (Liu et al., 2024b). In LLMs, we see similar strong gains: on TruthfulQA, GRAINS improves factual accuracy by 13.22% over the Llama-3.1-8B-Instruct model, outperforming ICV (Liu et al., 2024c) by a margin of 7.7%. On Toxigen, it improves the accuracy by over 9.89% over the base model and 4.10% over NL-ITI (Hoscilowicz et al., 2024). Moreover, because of GRAINS's localized nature, there is no major impact on the model's general capabilities on other tasks. When evaluating on broad-coverage text and multimodal datasets like MMLU (Hendrycks et al., 2021) and MMMU (Yue et al., 2024), standard baselines hurt performance, while GRAINS preserves performance. For example, CAA (Rimsky et al., 2024) drops Llama-3.1-8B's MMLU performance by 17.78%, while GRAINS is almost identical, with only a 0.12% drop). Similarly, CAA leads to a 17.13% drop on MMMU for Qwen2.5-VL-7B, while GRAINS has only a 0.51% drop. These results highlight the strength of selective, attribution-guided interventions for fine-grained multimodal control without performance degradation.

2 Related work

Inference-Time Steering. Inference-time intervention offers a lightweight alternative to fine-tuning by modifying hidden activations without updating model weights. In LLMs, methods like ITI (Li et al., 2024b), CAA (Panickssery et al., 2023), and MAT-Steer (Nguyen et al., 2025b) steer behavior using contrastive examples or attribute-specific vectors. For VLMs, prior work includes both modality-specific and activation-engineering approaches: VTI (Liu et al., 2024b) and MLLM-Steering (Khayatan et al., 2025) largely treat vision and language separately, ICT (Chen et al., 2024a) performs token-level interventions but depends on object detectors and supervision, and SteerVLM (Sivakumar et al., 2025) introduces a lightweight learned steering module that dynamically adjusts hidden activations using paired target and converse prompts. In contrast, GRAINS unifies steering across modalities by using gradient-based attribution to identify influential visual and textual tokens and construct layer-wise steering vectors directly from the input. This avoids modality-specific heuristics, learned auxiliary steering modules, and global interventions (Liu et al., 2024b; Chen et al., 2024a; Rimsky et al., 2024), enabling effective and interpretable control.

Attribution and Interpretability. Token-level attribution methods are widely used to interpret the outputs of LLMs and VLMs. Integrated Gradients (IG) (Sundararajan et al., 2017), a foundational technique, estimates token contributions by integrating gradients from a baseline input. Other gradient attribution methods such as SmoothGrad (Smilkov et al., 2017) and Guided IG (Kapishnikov et al., 2021) improve stability and reduce noise. These methods have been applied to analyze attention and debug hallucinations (Wu et al., 2023; Chang et al., 2024; Yang et al., 2025), but are typically limited to post-hoc explanation. As Lin et al. (2025) notes, interpretability tools rarely inform model control. In this work, we bridge this gap by using gradient-based attribution to guide intervention by identifying impactful tokens and computing contrastive, layer-wise steering vectors, enabling input-sensitive control without retraining.

3 Methodology

Here, we introduce **Gr**adient-based **A**ttribution for **In**ference-Time **S**teering (GRAINS), a steering approach that operates selectively on the most influential input tokens. Our method consists of three steps: (1) identifying important tokens using contrastive attribution based on preference data, (2) constructing layer-specific steering vectors from contrastive activations, and (3) applying selective and normalized interventions during inference. We illustrate GRAINS in Fig. 2 and we describe each of its steps below. Note that the steps in Section 3.1 and Section 3.2 are one-time costs and are performed only once per steering objective.

3.1 Token Attribution via Integrated Gradients

Objective. We begin by identifying the most influential tokens with respect to a model's prediction. Let P_θ be the output distribution of a model with parameters θ, which takes an input sequence $x = \{x_1, x_2, \ldots, x_T\}$, which includes both textual and visual token embeddings in the case of VLMs. To find key tokens, we leverage a contrastive attribution signal grounded in preference data. Specifically, rather than computing gradients with respect to a single output logit, we define the attribution objective using a preference-based loss:[2]

$$f(x) = \log P_\theta(y_{\text{pos}} \mid x) - \log P_\theta(y_{\text{neg}} \mid x), \quad (1)$$

where $P_\theta(y \mid x)$ denotes the conditional log-probability of output y given input x, as assigned by the model. Here, y_{pos} and y_{neg} represent the preferred and dispreferred responses, respectively. For example, if steering the model to be less toxic, y_{pos} would be a non-toxic response and y_{neg} would be a negative response. This contrastive formulation captures the model's relative preference between two candidate completions, aligning more closely with human annotation and preference optimization objectives than absolute likelihoods.

Token Attribution. Given this objective $f(x)$, we apply *Integrated Gradients* (IG) (Sundararajan et al., 2017) to compute the attribution score for each input token embedding x_j:

$$\text{IG}_j(x) := (x_j - \tilde{x}_j) \times \int_{\alpha=0}^{1} \frac{\partial f(\tilde{x} + \alpha(x - \tilde{x}))}{\partial x_j} \, d\alpha,$$

where $\tilde{x}$ is a neutral baseline input (e.g., zero or masked token embedding). The resulting attribution $\text{IG}_j(x)$ quantifies the contribution of token x_j to the model's preference for y_{pos} over y_{neg}.

[2] In cases where explicit preference data is unavailable, we show in an ablation study in Appendix A.3 that using a single reference output (e.g., y_{pos}) is still effective.

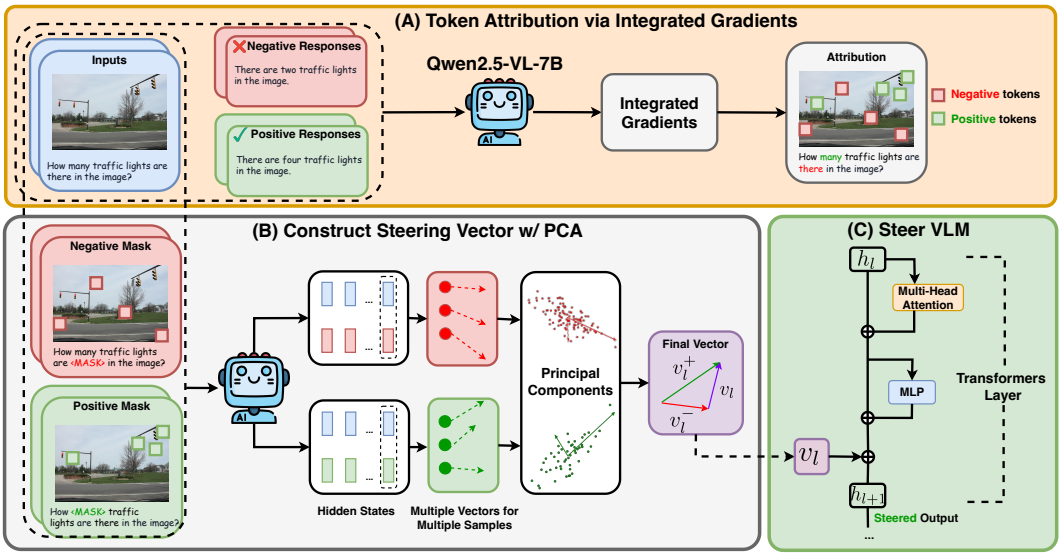

Figure 2: Overview of our attribution-guided steering method for VLMs. Our method consists of three stages: (A) Compute token-level attributions using contrastive Integrated Gradients, identifying the most influential positive and negative tokens (green/red). (B) Construct contrastive inputs by masking these tokens, extract the corresponding hidden states, and apply PCA to obtain directional steering vectors. (C) At inference time, inject these vectors into the model's hidden states at each layer, scaled and normalized to preserve representation scale.

IG provides *signed attribution scores*: positive values indicate tokens that increase the model's preference for y_{pos}, while negative values indicate tokens that favor y_{neg}. To obtain a scalar attribution score for each token x_j, we sum the components of its IG vector: $a_j(x) = \sum_{i=1}^{d} \text{IG}_j^{(i)}(x)$. This aggregation yields a signed score that reflects the influence of the token on the model's output, enabling clear comparison across tokens (Atanasova et al., 2020; Pezeshkpour et al., 2022). Such scalar scores are essential for ranking and selecting the most impactful inputs for downstream intervention. We then define two sets of top-k influential tokens (corresponding to the green and red token groups in Fig. 2 (A)) based on these scores:

$$\mathcal{I}_k^+(x) = \{x_j \in x : a_j(x) \text{ is among the top-}k \text{ positive scores}\},$$

$$\mathcal{I}_k^-(x) = \{x_j \in x : a_j(x) \text{ is among the top-}k \text{ most negative scores}\}.$$

This allows us to disentangle how the model responds to desirable versus undesirable behavior, enabling finer-grained control in downstream steering. As shown in Figure 3, removing the most negatively-attributed tokens causes a substantial increase in the model's preference for y_{pos}, while removing positively-attributed tokens leads to the opposite effect. These asymmetries highlight that negative attribution identifies strong contributors

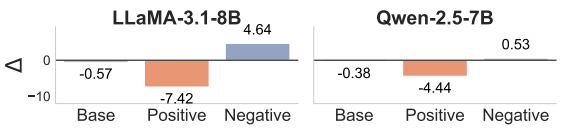

Figure 3: Effect on preference difference $\Delta = \log P_\theta(y_{\text{pos}} \mid x) - \log P_\theta(y_{\text{neg}} \mid x)$ after ablating top-k tokens based on signed Integrated Gradients. Removing tokens with high negative attribution substantially increases model preference for aligned outputs (y_{pos}), whereas removing high positive tokens leads to preference drops. Results shown for Llama-3.1-8B-Instruct and Qwen-2.5-7B-Instruct models on TruthfulQA.

to undesirable model behavior, forming the foundation for constructing directional steering vectors.

Why Use Integrated Gradients? We choose Integrated Gradients over vanilla (first-order) gradients due to its theoretical and practical advantages. First, vanilla gradients are known to suffer from saturation: when a model is confident in an output, the gradient magnitude can diminish, even if the input token is critical to the decision (Smilkov et al., 2017; Sundararajan et al., 2017). IG mitigates this by accumulating gradients along a path from a baseline to the actual input, yielding more robust and faithful attributions. Second, IG satisfies desirable axiomatic properties such as *sensitivity* and *implementation invariance* (Sundararajan et al., 2017), which vanilla gradients lack. As a result, IG pro-

vides more stable, interpretable, and reliable token importance scores, especially in high-dimensional, non-linear models like LLMs and VLMs. While developing a new attribution method is not the focus of our work, we include technical details in Appendix B and a comparison in Appendix A.3 to empirically validate the effectiveness of IG over other attribution methods.

Compared to perturbation-based attribution methods such as SHAP (Lundberg and Lee, 2017), IG is more efficient in our setting: perturbation methods require sampling many perturbations, leading to a significant number of forward passes, which is computationally expensive for high-dimensional inputs (Agarwal et al., 2021; Rao et al., 2022), especially for VLMs. In contrast, IG attribution uses only 5-20 approximation steps (details including running time are provided in Section 4 and Appendix A.1) and achieves strong steering performance for both LLMs and VLMs.

3.2 Constructing Layer-Wise Steering Vectors

Contrastive Steering Vectors. Once the top positively and negatively attributed tokens are identified, we construct two modified inputs: $x_{\setminus \mathcal{I}^+}$ (where top-k positive tokens are replaced by baselines) and $x_{\setminus \mathcal{I}^-}$ (where top-k negative tokens are replaced). These substitutions isolate the collective contribution of each polarity group to the model's internal representations.[3]

Let $h^{(l)}_{\text{last}}(x) \in \mathbb{R}^d$ denote the hidden activation at the final token of the sequence at transformer layer l. Following prior work (Li et al., 2024b), we use this position as it typically aggregates contextual information from the entire sequence and provides a consistent anchor point for measuring how input changes propagate through the model across layers. We define the *contrastive steering vectors* as:

$$\begin{aligned} \delta_l^{+,(x)} &= h^{(l)}_{\text{last}}(x) - h^{(l)}_{\text{last}}(x_{\setminus \mathcal{I}^+}), \\ \delta_l^{-,(x)} &= h^{(l)}_{\text{last}}(x) - h^{(l)}_{\text{last}}(x_{\setminus \mathcal{I}^-}). \end{aligned} \tag{2}$$

These vectors quantify the directional shift in the model's hidden representation when high-impact tokens are ablated. Intuitively, $\delta_l^{+,(x)}$ captures how the model relies on tokens that support aligned, desirable outputs, while $\delta_l^{-,(x)}$ captures how it relies on tokens contributing to misaligned, undesirable outputs (e.g., hallucinations, toxicity).

[3]Replacing one token at a time may offer more granularity but is computationally expensive and in practice yields similar effect (Covert et al., 2021; Rong et al., 2022).

PCA for Vector Aggregation. We aim to extract a single per-layer steering vector that can be applied at inference to unseen inputs. However, the per-example contrastive deltas $\delta_l^{+,(x)}$ and $\delta_l^{-,(x)}$ can vary in magnitude, so naive averaging can cancel out signals (Yin and Neubig, 2022; Ferrando et al., 2023). We extract a stable, low-dimensional steering direction by applying Principal Component Analysis (PCA) over many examples (see Appendix A.3 for the vector aggregation ablation). PCA serves two roles: it aggregates noisy vectors into a robust semantic direction and ensures the steering vector generalizes across diverse inputs. Specifically, we compute the top principal component of each set across a steering dataset $\mathcal{D}$, yielding steering vectors $v_l^+ \in \mathbb{R}^d$ and $v_l^- \in \mathbb{R}^d$, as illustrated in Fig. 2 (B):

$$\begin{aligned} v_l^+ &= \text{PCA}_1\{\delta_l^{+,(x)} : x \in \mathcal{D}\}, \\ v_l^- &= \text{PCA}_1\{\delta_l^{-,(x)} : x \in \mathcal{D}\}. \end{aligned} \tag{3}$$

We then define the final contrastive steering vector at layer l as:

$$v_l = v_l^+ - v_l^-, \tag{4}$$

which captures the latent direction from desirable to undesirable behavior. The contrastive vector reflects both suppression of undesirable semantics (via v_l^-) and enhancement of desirable ones (via v_l^+), and is used at inference time to steer the model away from behaviors tied to high-impact inputs. Empirically, we observe that the first principal component (PC1) is both efficient and robust: a single component captures the task-relevant variance while avoiding overfitting (see Table 9 for the ablation study on PCA). Across datasets in Section 4, we observe that the explained-variance ratio of PC1 is 0.7-0.9, indicating a dominant, shared subspace and supporting reliability across different cases of the same attribute. Using more components of PCA would be inefficient since it requires injecting multiple vectors per layer with per-component scaling/combination.

3.3 Steering at Inference Time

At inference, we steer the model's generation by applying the vectors across layers during decoding. Let $h_{t,l} \in \mathbb{R}^d$ be the activation at token position t and layer l. For each position and layer, we apply an additive intervention to the activation and rescale to match the original norm (see Fig. 2 (C)):

$$\tilde{h}_{t,l} \;=\; \big(h_{t,l} + \lambda\, v_l\big) \;\times\; \frac{\|h_{t,l}\|_2}{\|h_{t,l} + \lambda\, v_l\|_2}, \tag{5}$$

where λ is a hyperparameter controlling the strength of steering. Our formulation ensures the adjustment is smooth and maintains compatibility with downstream modules, while allowing for consistent behavioral shifts in the model (Liu et al., 2024c). Importantly, because these vectors are constructed from the tokens with the largest attribution-based contributions using contrastive gradient attribution, the intervention is both targeted and proportional. The norm preservation reduces the risk of overcorrecting unrelated behaviors (Liu et al., 2024c), focusing the adjustment precisely on the factors responsible for misalignment (see the qualitative analysis in Section 5.2 and the ablation study of the normalization step in Appendix A.3).

4 Experiments

We evaluate our GRAINS across both language-only (LLMs) and multimodal (VLMs) settings. Our focus is on safety-critical scenarios involving undesirable outputs. For each domain, we compare against standard baselines including fine-tuned models and existing steering methods. More details on settings, running time, hyperparameter analysis and results are provided in Appendix A.

4.1 LLM Experiments

Models. We use Llama-3.1-8B-Instruct (Dubey et al., 2024) and Qwen2.5-7B-Instruct (Team, 2024) as our base models for evaluating text-only settings. These models are chosen for their strong capabilities and because they serve as the language components of their corresponding VLMs evaluated later in our multimodal experiments.

Datasets. We evaluate GRAINS on multiple-choice QA datasets that each target a separate LLM attribute for LLM safety: **TruthfulQA** (truthfulness) (Lin et al., 2022), **Toxigen** (toxicity) (Hartvigsen et al., 2022), **FaithEval** (context faithfulness) (Ming et al., 2025). We report multiple-choice accuracy for each dataset.

Inference-time Steering with GRAINS. We select 50 samples from each dataset for constructing the steering vectors. For each example, we compute token-level attributions for text tokens using the preference loss described in Section 3.1. In all experiments, we set $k = 3$ tokens. For IG, we use 5 steps for gradient estimation. Steering vectors are computed using PCA over contrastive activation vectors from multiple inputs. These are applied at inference to adjust the model's hidden activations.

Baselines. We compare GRAINS against approaches for steering LLMs. We employ LoRA fine-tuning (Hu et al., 2022) as a representative parameter-efficient fine-tuning (PEFT) method. We also compare against state-of-the-art inference-time intervention methods including ICV (Liu et al., 2024c), NL-ITI (Hoscilowicz et al., 2024), CAA (Rimsky et al., 2024). We note that there are other steering baselines such as RepE (Zou et al., 2023) and ITI (Li et al., 2024b), but recent work (Im and Li, 2025) has shown that they underperform compared to our selected baselines like NL-ITI and CAA across multiple benchmarks, so we do not include them in our comparisons.

Results: GRAINS Improves Steering of LLMs. Table 1 shows that GRAINS outperforms both LoRA and existing steering baselines across all three tasks. On TruthfulQA, GRAINS improves accuracy by 8.44% on Qwen2.5-7B-Instruct and by 13.22% on Llama-3.1-8B-Instruct, outperforming ICV, NL-ITI, and CAA. On Toxigen, our method improves accuracy significantly by 7.79% for Llama and 7.08% for Qwen over their respective base models. For FaithEval, which evaluates contextual consistency, GRAINS again achieves the highest accuracy 70.94% on Llama and 64.77% on Qwen, showing strong gains across models.

4.2 VLM Experiments

Models. We use LLaVA-v1.6-7B (Liu et al., 2024a), Qwen2.5-VL-7B-Instruct (Team, 2024), and Gemma-3-12B (Team et al., 2025).

Datasets. We evaluate on two key failure modes in multimodal generation using **MMHal-Bench** (hallucination) (Sun et al., 2023) and **SPA-VL** (safety) (Zhang et al., 2025). For MMHal-Bench, we report the hallucination rate using GPT-4o as the judge model. We observe strong agreement between GPT-4o and human annotations, with a Pearson correlation of 0.82 and a Spearman correlation of 0.85, based on evaluations from 9 human annotators. For SPA-VL we report the preference win rate of *chosen* > *rejected* responses based on model log probability. This metric is standard in alignment work and shown to correlate with human preferences (Rafailov et al., 2024; Li et al., 2024a).

Inference-time Steering with GRAINS. Similar to LLM experiments in Section 4.1, we select 50 samples for constructing the steering vectors. As VLMs might require processing more tokens, including both visual and textual tokens, in all ex-

Method	Llama				Qwen			
	TruthfulQA	Toxigen	FaithEval	Avg.	TruthfulQA	Toxigen	FaithEval	Avg.
Base Model	34.15	51.19	68.00	51.11	51.41	55.04	59.89	55.45
LoRA	40.67	58.78	69.93	56.46	56.87	59.98	**64.96**	60.60
ICV	39.67	59.07	68.65	55.80	53.06	59.72	63.64	58.81
NL-ITI	37.04	56.88	69.46	54.46	52.95	60.54	60.38	57.96
CAA	44.62	58.89	69.32	57.61	56.74	60.01	62.21	59.65
GRAINS	**47.37**	**60.98**	**70.94**	**59.76**	**59.85**	**62.12**	64.77	**62.25**

Table 1: Performance on LLM benchmarks for both LLaMA-3.1-8B and Qwen2.5-7B. Accuracy (higher is better) reported for TruthfulQA, Toxigen, and FaithEval. Avg. columns show the mean across benchmarks per model.

Method	MMHal-Bench ↓				SPA-VL ↑			
	LLaVA	Qwen-VL	Gemma	Avg.	LLaVA	Qwen-VL	Gemma	Avg.
Base Model	0.624	0.523	0.468	0.538	40.24	53.21	49.32	47.59
LoRA	0.565	**0.461**	0.464	0.497	45.72	56.83	52.37	51.64
VTI	0.587	0.499	0.460	0.515	42.46	54.42	51.45	49.44
ICT	0.592	0.515	0.457	0.521	43.18	54.45	52.13	49.92
RUDDER	0.605	0.512	0.481	0.533	43.25	54.71	50.28	49.41
SHARP	0.569	0.465	0.462	0.499	43.07	55.34	51.96	50.12
CAA	0.610	0.537	0.493	0.547	43.71	53.60	50.63	49.31
GRAINS	**0.514**	0.473	**0.442**	**0.476**	**48.35**	**58.90**	**53.51**	**53.59**

Table 2: Comparison across MMHal-Bench and SPA-VL benchmarks. Left: MMHal-Bench reports hallucination rate (lower is better). Right: SPA-VL reports preference win rate (higher is better). Avg. columns reflect the mean performance across the three models.

periments, we set $k = 20$ tokens. For IG, we use 5 steps for gradient approximation in LLaVA and Qwen, and 10 steps for the larger Gemma model to ensure more reliable attribution.

Baselines. We compare GRAINS against approaches for aligning VLMs. For fair comparison, we use the same samples used to construct steering vectors for GRAINS for all steering baselines. In addition to LoRA (Hu et al., 2022), we compare against state-of-the-art steering methods for VLMs, including VTI (Liu et al., 2024b), which applies modality-specific vector shifts to reduce hallucinations, and ICT (Chen et al., 2024a), which performs object-grounded interventions but relies on external object detectors, RUDDER (Zou et al., 2025), which adaptively injects per-sample visual evidence directions from residual updates for low-overhead hallucination mitigation, and SHARP (Wu et al., 2025), which steers cause-specific latent representations to suppress different types of hallucinations during inference. Additionally, we adapt CAA (Rimsky et al., 2024) to VLMs by directly incorporating their steering mechanisms into the LLM component of the VLM.

Results: GRAINS Improves Steering of VLMs. Table 2 shows that GRAINS achieves the lowest hallucination rates across all three VLMs on MMHal-Bench. On LLaVA-1.6-7B, GRAINS reduces the hallucination rate from 0.624 of the base model to 0.514, outperforming baselines such as SHARP (0.569) and VTI (0.587). On Qwen2.5-VL-7B, it lowers hallucinations from 0.523 to 0.473. For Gemma-3-12B-IT, GRAINS yields the best result (0.442), outperforming all other baselines. Moreover, GRAINS has the highest preference win rates on SPA-VL. It improves LLaVA-1.6-7B from 40.24% to 48.35%, Qwen2.5-VL-7B from 53.21% to 58.90%, and Gemma-3-12B-IT from 49.32% to 53.51%. These gains exceed all other steering and fine-tuning baselines, which range between 1–5% lower per model. These results indicate that GRAINS improves both hallucination and safety.

5 Analysis

In this section, we present a deeper analysis of GRAINS, covering its impact on general model capabilities, qualitative analysis, generalization of GRAINS, and computational overhead analysis. We provide ablation studies and additional results (including attribution methods, token selection) in Appendix A.3 and Appendix A.4.

(A) Correction of hallucinated objects.
Q: What are the objects from right to left?
LLaVA-1.6-7B: A spoon, a coffee cup, and a saucer.
ICT: A spoon, a coffee cup, and a saucer.
VTI: A spoon, a coffee cup, and a saucer.
GRAINS: A lid, a spoon, and a coffee cup.

(B) Preserving correct behavior.
Q: Who is sitting on the bench?
Qwen2.5-VL-7B: No one is sitting.
ICT: A man is sitting.
VTI: A man is sitting.
GRAINS: The benches are empty.

Figure 4: Qualitative examples from MMHal-Bench. (**A**) Only GRAINS provides the correct object order. (**B**) The base model is correct, but baselines introduce hallucinations; GRAINS preserves the correct, grounded answer.

Method	TruthfulQA ↑		MMLU ↑	
	Llama	Qwen	Llama	Qwen
Base Model	38.19	47.65	**69.27**	**74.58**
LoRA	46.81	51.96	67.69	72.17
ICV	46.60	49.09	69.18	74.41
NL-ITI	45.71	48.62	65.74	70.33
CAA	46.52	49.63	51.49	62.91
GRAINS	**47.91**	**54.09**	69.15	74.29

Table 3: LLM capability comparison: we report BLEU accuracy for TruthfulQA, 5-shot accuracy for MMLU.

Method	SPA-VL ↑		MMMU ↑	
	LLaVA	Qwen-VL	LLaVA	Qwen-VL
Base Model	42.38	49.17	**35.81**	**58.64**
LoRA	47.27	51.74	34.55	57.36
VTI	45.92	51.87	35.63	58.41
ICT	**47.65**	52.01	34.11	53.29
RUDDER	43.26	50.48	33.75	52.63
SHARP	45.79	52.16	31.68	49.82
CAA	42.13	50.14	33.29	41.51
GRAINS	46.79	**53.02**	34.92	58.13

Table 4: VLM capability comparison: we report BLEU accuracy for SPA-VL, 5-shot accuracy for MMMU.

5.1 Impact on General Model Capabilities

A desirable steering method should reduce harmful behavior and hallucination without degrading the model's capabilities. We evaluate whether GRAINS preserves core capabilities such as fluency and reasoning after intervention.

Generation Qualities. Following prior work (Pham and Nguyen, 2024; Nguyen et al., 2025b), we assess the effect of steering on open-ended generation tasks using TruthfulQA for LLMs and SPA-VL for VLMs. We report BLEU accuracy, defined as the proportion of generated outputs that are closer by BLEU score (Papineni et al., 2002) to the correct (positive) reference than to the incorrect (negative) one. This metric captures whether steering disrupts fluency or semantic correctness of the generations and is commonly used in prior work (Bi et al., 2022; Chang et al., 2025). For LLMs, as shown in Table 3, GRAINS achieves the highest BLEU accuracy on both Llama-3.1-8B (47.91%) and Qwen2.5-7B (54.09%). For VLMs, Table 4 shows that GRAINS also performs competitively, achieving 46.79% on LLaVA-1.6-7B and the highest score of 53.02% on Qwen2.5-VL-7B. These results show that GRAINS aligns outputs more closely with human-preferred responses while preserving generation quality.

General Reasoning Capabilities. We evaluate 5-shot accuracy on reasoning datasets using MMLU (Hendrycks et al., 2021) for LLMs and MMMU (Yue et al., 2024) for VLMs. These benchmarks cover a wide range of subjects, allowing us to measure whether GRAINS and steering baselines affect the model's ability to perform general-purpose reasoning. On MMLU, Table 3 shows that GRAINS maintains comparable performance to the base models, with 69.15% accuracy on Llama-3.1-8B (vs. 69.27% base) and 74.29% on Qwen2.5-7B (vs. 74.58% base). Unlike other steering methods, which degrade reasoning accuracy substantially (e.g., CAA drops to 51.49% on Llama), GRAINS preserves reasoning ability. Similarly, on VLMs (Table 4), GRAINS maintains accuracy on MMMU, with 34.92% on LLaVA-1.6-7B and 58.13% on Qwen2.5-VL-7B, only slightly below the base models. This shows that GRAINS minimally disrupts models' reasoning abilities.

Method	RTP (Toxicity) ↓
Qwen	4.18
ICV	3.35
CAA	2.89
GRAINS	**1.15**

Table 5: Generalization of GRAINS to RealToxicityPrompts.

Method	POPE (Accuracy) ↑	POPE (F1 Score) ↑
Qwen-VL	79.85	76.82
VTI	83.54	79.63
CAA	80.71	77.14
GRAINS	**84.06**	**80.44**

Table 6: Generalization of GRAINS to POPE.

5.2 Qualitative Analysis

Fig. 4 presents two representative examples from MMHal-Bench that highlight the effectiveness of GRAINS compared to baseline VLMs and steering approaches. In example (A), baseline models, including LLaVa-1.6-7B, ICT, and VTI hallucinate the object locations. Only GRAINS identifies the objects in the correct order (though it refers to the right-most object as a lid, which is less likely than saucer), demonstrating improved grounding to visual evidence. In contrast, example (B) illustrates a failure mode of prior steering methods: while the original Qwen2.5-VL-7B prediction is correct (*"no one is sitting on the bench"*), steering baselines introduce hallucinated content by incorrectly claiming someone is present. GRAINS avoids this regression and preserves valid base model behavior. These examples illustrate GRAINS's ability to modulate outputs based on token-level and modality-aware attribution signals, enabling both behavioral improvement and alignment fidelity. We provide more qualitative results in Fig. 7.

5.3 Generalization of GRAINS

To directly test out-of-distribution generalization, we add an additional experiment in which we build the steering vector on a source dataset and evaluate it on a target dataset sharing the same attribute, such as toxicity or hallucination. For LLM toxicity mitigation, we use the vector on Toxigen and evaluate on RealToxicityPrompts (RTP) (Gehman et al., 2020) with Qwen2.5-7B-Instruct (toxicity, lower is better). For VLM hallucination reduction, we use the vector on MMHal-Bench and evaluate on the POPE (Li et al., 2023) adversarial split (the most challenging setting of POPE) with Qwen2.5-VL-7B-Instruct. We compare against strong steering baselines under the same setting.

Table 5 and Table 6 show that our method outperforms baselines, and these cross-dataset gains show that a vector learned from one dataset with a specific attribute transfers to different distributions and prompt formats. This supports our claim that GrAInS captures meaningful directions (e.g., non-toxic, non-hallucinatory behavior).

Model	Setting	Tokens/ms	Throughput Drop
Qwen	Base Model	0.0299	-
	GRAINS	0.0288	3.55%
Llama	Base Model	0.0263	-
	GRAINS	0.0254	3.62%

Table 7: Inference throughput comparison between the base model and GrAInS.

6 Discussion and Conclusion

Discussion on Computational Overhead. At inference time, GRAINS does not run attribution or extra forward passes. It only adds a precomputed per-layer vector and applies scale-preserving normalization. Additionally, we do not alter KV-cache shapes or softmax paths, so the throughput/latency remains essentially unchanged. In Table 7, we report the inference throughput (tokens/ms) and compute the throughput drop of GrAInS compared to the base model on a single RTX A6000 GPU for TruthfulQA. The impact on inference is minimal (<5%), showing that GRAINS introduces only negligible runtime overhead while preserving the deployment efficiency of the original model.

Conclusion. We introduce GRAINS, a novel steering approach that finds the most influential input tokens across modalities, and uses contrastive activation shifts to compute steering vectors. Unlike prior methods that apply fixed intervention or rely solely on visual tokens, GRAINS enables fine-grained and interpretable control without retraining or external modules. Our approach achieves consistent gains in reducing hallucination, increasing preference alignment, and preserving generation quality and reasoning capabilities across LLMs and VLMs. By integrating attribution with steering, GRAINS bridges the gap between interpretability and controllability in modern language and vision-language models.

Limitations

While GRAINS demonstrates strong empirical performance across a range of tasks and models, it has some limitations. First, like other attribution-based methods, GRAINS depends on the quality of token-level attribution. While attribution methods like IG provide a principled and effective foundation for identifying influential tokens, they are not without drawbacks. Attribution quality can vary depending on the model architecture and the choice of baseline input, which may affect the precision of steering vectors. Additionally, IG and related methods require gradient access and are computationally more expensive than simpler heuristics, which could pose challenges for scaling to very large models. Future work may explore alternative or learned attribution techniques that improve token selection efficiency and quality. Another limitation of our paper and other steering methods is that there is no formal guarantee that modifying internal representations based on attributed tokens will correct the model's behavior. Future work could investigate methods for constraining the downstream effects of such interventions, potentially combining attribution with disentangled representations for more robust interventions.

Acknowledgements

We thank Jaemin Cho for his helpful comments and suggestions on this paper. This work was supported by NSF-CAREER Award 1846185, DARPA ECOLE Program No. HR00112390060, and NSF-AI Engage Institute DRL-2112635, ARO Award W911NF2110220, ONR Grant N00014-23-12356, and an Apple PhD Fellowship. The views contained in this article are those of the authors and not of the funding agency.

References

Chirag Agarwal, Nari Johnson, Martin Pawelczyk, Satyapriya Krishna, Eshika Saxena, Marinka Zitnik, and Himabindu Lakkaraju. 2022. Rethinking stability for attribution-based explanations. *arXiv preprint arXiv:2203.06877*.

Sushant Agarwal, Shahin Jabbari, Chirag Agarwal, Sohini Upadhyay, Steven Wu, and Himabindu Lakkaraju. 2021. Towards the unification and robustness of perturbation and gradient based explanations. In *International conference on machine learning*, pages 110–119. PMLR.

Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. 2019. Gradient-based attribution methods. In *Explainable AI: Interpreting, explaining and visualizing deep learning*, pages 169–191. Springer.

Pepa Atanasova, Jakob Grue Simonsen, Christina Lioma, and Isabelle Augenstein. 2020. A diagnostic study of explainability techniques for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3256–3274, Online. Association for Computational Linguistics.

Qiwei Bi, Jian Li, Lifeng Shang, Xin Jiang, Qun Liu, and Hanfang Yang. 2022. MTRec: Multi-task learning over BERT for news recommendation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2663–2669, Dublin, Ireland. Association for Computational Linguistics.

Jianjian Cao, Peng Ye, Shengze Li, Chong Yu, Yansong Tang, Jiwen Lu, and Tao Chen. 2024. Madtp: Multimodal alignment-guided dynamic token pruning for accelerating vision-language transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15710–15719.

Yapei Chang, Yekyung Kim, Michael Krumdick, Amir Zadeh, Chuan Li, Chris Tanner, and Mohit Iyyer. 2025. Bleuberi: Bleu is a surprisingly effective reward for instruction following. *arXiv preprint arXiv:2505.11080*.

Yurui Chang, Bochuan Cao, Yujia Wang, Jinghui Chen, and Lu Lin. 2024. Xprompt: Explaining large language model's generation via joint prompt attribution. *arXiv preprint arXiv:2405.20404*.

Junzhe Chen, Tianshu Zhang, Shiyu Huang, Yuwei Niu, Linfeng Zhang, Lijie Wen, and Xuming Hu. 2024a. Ict: Image-object cross-level trusted intervention for mitigating object hallucination in large vision-language models. *Preprint*, arXiv:2411.15268.

Meiqi Chen, Yixin Cao, Yan Zhang, and Chaochao Lu. 2024b. Quantifying and mitigating unimodal biases in multimodal large language models: A causal perspective. *arXiv preprint arXiv:2403.18346*.

Qizhou Chen, Taolin Zhang, Chengyu Wang, Xiaofeng He, Dakan Wang, and Tingting Liu. 2025. Attribution analysis meets model editing: Advancing knowledge correction in vision language models with visedit. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 2168–2176.

Ian C. Covert, Scott Lundberg, and Su-In Lee. 2021. Explaining by removing: a unified framework for model explanation. *J. Mach. Learn. Res.*, 22(1).

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

Matteo Farina, Gianni Franchi, Giovanni Iacca, Massimiliano Mancini, and Elisa Ricci. 2024. Frustratingly easy test-time adaptation of vision-language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Javier Ferrando, Gerard I. Gállego, Ioannis Tsiamas, and Marta R. Costa-jussà. 2023. Explaining how transformers use context to build predictions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5486–5513, Toronto, Canada. Association for Computational Linguistics.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. 2020. RealToxicityPrompts: Evaluating neural toxic degeneration in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3356–3369, Online. Association for Computational Linguistics.

Soumya Suvra Ghosal, Souradip Chakraborty, Vaibhav Singh, Tianrui Guan, Mengdi Wang, Ahmad Beirami, Furong Huang, Alvaro Velasquez, Dinesh Manocha, and Amrit Singh Bedi. 2024. Immune: Improving safety against jailbreaks in multi-modal llms via inference-time alignment. *arXiv preprint arXiv:2411.18688*.

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

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

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

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

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

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

Andrei Kapishnikov, Subhashini Venugopalan, Besim Avci, Ben Wedin, Michael Terry, and Tolga Bolukbasi. 2021. Guided integrated gradients: An adaptive path method for removing noise. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5050–5058.

Pegah Khayatan, Mustafa Shukor, Jayneel Parekh, and Matthieu Cord. 2025. Analyzing fine-tuning representation shift for multimodal llms steering alignment. *arXiv preprint arXiv:2501.03012*.

Sicong Leng, Hang Zhang, Guanzheng Chen, Xin Li, Shijian Lu, Chunyan Miao, and Lidong Bing. 2024. Mitigating object hallucinations in large vision-language models through visual contrastive decoding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13872–13882.

Junlong Li, Fan Zhou, Shichao Sun, Yikai Zhang, Hai Zhao, and Pengfei Liu. 2024a. Dissecting human and llm preferences. *arXiv preprint arXiv:2402.11296*.

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

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. 2023. Evaluating object hallucination in large vision-language models. *arXiv preprint arXiv:2305.10355*.

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

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

Zihao Lin, Samyadeep Basu, Mohammad Beigi, Varun Manjunatha, Ryan A. Rossi, Zichao Wang, Yufan Zhou, Sriram Balasubramanian, Arman Zarei, Keivan Rezaei, Ying Shen, Barry Menglong Yao, Zhiyang Xu, Qin Liu, Yuxiang Zhang, Yan Sun, Shilong Liu, Li Shen, Hongxuan Li, and 2 others. 2025. A survey on mechanistic interpretability for multi-modal foundation models. *Preprint*, arXiv:2502.17516.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. 2024a. Llava-next: Improved reasoning, ocr, and world knowledge.

Sheng Liu, Haotian Ye, Lei Xing, and James Zou. 2024b. Reducing hallucinations in vision-language models via latent space steering. *arXiv preprint arXiv:2410.15778*.

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

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

Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30.

Samuel Marks and Max Tegmark. 2023. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. *arXiv preprint arXiv:2310.06824*.

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

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

Duy Nguyen, Archiki Prasad, Elias Stengel-Eskin, and Mohit Bansal. 2025b. Multi-attribute steering of language models via targeted intervention. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 20619–20634, Vienna, Austria. Association for Computational Linguistics.

Nina Panickssery, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. 2023. Steering llama 2 via contrastive activation addition. *arXiv preprint arXiv:2312.06681*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Pouya Pezeshkpour, Sarthak Jain, Sameer Singh, and Byron Wallace. 2022. Combining feature and instance attribution to detect artifacts. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1934–1946, Dublin, Ireland. Association for Computational Linguistics.

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

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

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

Sukrut Rao, Moritz Böhle, and Bernt Schiele. 2022. Towards better understanding attribution methods. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 10223–10232.

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

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. 2023. Steering llama 2 via contrastive activation addition. *arXiv preprint arXiv:2312.06681*.

Yao Rong, Tobias Leemann, Vadim Borisov, Gjergji Kasneci, and Enkelejda Kasneci. 2022. A consistent and efficient evaluation strategy for attribution methods. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 18770–18795. PMLR.

Emmanuelle Salin, Badreddine Farah, Stéphane Ayache, and Benoit Favre. 2022. Are vision-language transformers learning multimodal representations? a probing perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11248–11257.

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

Anushka Sivakumar, Andrew Zhang, Zaber Ibn Abdul Hakim, and Chris Thomas. 2025. SteerVLM: Robust model control through lightweight activation steering for vision language models. In *Findings of the Association for Computational Linguistics: EMNLP 2025*. Association for Computational Linguistics.

Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. 2017. Smoothgrad: removing noise by adding noise. *Preprint*, arXiv:1706.03825.

Manogna Sreenivas and Soma Biswas. 2025. Efficient open-set test time adaptation of vision language models. In *First Workshop on Scalable Optimization for Efficient and Adaptive Foundation Models*.

Yizheng Sun, Yanze Xin, Hao Li, Jingyuan Sun, Chenghua Lin, and Riza Batista-Navarro. 2025. LVPruning: An effective yet simple language-guided vision token pruning approach for multi-modal large language models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 4299–4308, Albuquerque, New Mexico. Association for Computational Linguistics.

Zhiqing Sun, Sheng Shen, Shengcao Cao, Haotian Liu, Chunyuan Li, Yikang Shen, Chuang Gan, Liang-Yan Gui, Yu-Xiong Wang, Yiming Yang, and 1 others. 2023. Aligning large multimodal models with factually augmented rlhf. *arXiv preprint arXiv:2309.14525*.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3319–3328. PMLR.

Gemma Team and 1 others. 2025. Gemma 3 technical report. *Preprint*, arXiv:2503.19786.

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

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

David Wan, Jaemin Cho, Elias Stengel-Eskin, and Mohit Bansal. 2024. Contrastive region guidance: Improving grounding in vision-language models without training. In *European Conference on Computer Vision*, pages 198–215. Springer.

Yongjie Wang, Tong Zhang, Xu Guo, and Zhiqi Shen. 2024. Gradient based feature attribution in explainable ai: A technical review. *arXiv preprint arXiv:2403.10415*.

Junfei Wu, Yue Ding, Guofan Liu, Tianze Xia, Ziyue Huang, Dianbo Sui, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2025. SHARP: Steering hallucination in LVLMs via representation engineering. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Siyue Wu, Hongzhan Chen, Xiaojun Quan, Qifan Wang, and Rui Wang. 2023. Ad-kd: Attribution-driven knowledge distillation for language model compression. *arXiv preprint arXiv:2305.10010*.

Le Yang, Ziwei Zheng, Boxu Chen, Zhengyu Zhao, Chenhao Lin, and Chao Shen. 2025. Nullu: Mitigating object hallucinations in large vision-language models via halluspace projection. *Preprint*, arXiv:2412.13817.

Kayo Yin and Graham Neubig. 2022. Interpreting language models with contrastive explanations. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 184–198, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, and 3 others. 2024. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of CVPR*.

Yongting Zhang, Lu Chen, Guodong Zheng, Yifeng Gao, Rui Zheng, Jinlan Fu, Zhenfei Yin, Senjie Jin, Yu Qiao, Xuanjing Huang, Feng Zhao, Tao Gui, and Jing Shao. 2025. Spa-vl: A comprehensive safety preference alignment dataset for vision language model. *Preprint*, arXiv:2406.12030.

Lihua Zhou, Mao Ye, Shuaifeng Li, Nianxin Li, Xiatian Zhu, Lei Deng, Hongbin Liu, and Zhen Lei. 2025. Bayesian test-time adaptation for vision-language models. *Preprint*, arXiv:2503.09248.

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

Zhengtao Zou, Ya Gao, Jiarui Guan, Bin Li, and Pekka Marttinen. 2025. Adaptive residual-update steering for low-overhead hallucination mitigation in large vision language models. *arXiv preprint arXiv:2511.10292*.

A Experiments

A.1 Experimental Settings

Datasets. We provide the details for each dataset as follows:

- **Truthfulness:** The TruthfulQA dataset (Lin et al., 2022) assesses the model's ability to provide truthful responses.
- **Toxicity:** The Toxigen dataset (Hartvigsen et al., 2022) evaluates the model's capability to avoid generating toxic outputs.
- **Context Faithfulness:** FaithEval (Ming et al., 2025) assesses whether the model stays faithful to the given context when presented with misleading or contradict information.

- **Hallucination:** MMHal-Bench (Sun et al., 2023) measures hallucination rate in image-conditioned responses. We follow the setting in previous work (Liu et al., 2024b) for evaluation of hallucination rate.
- **Safety:** SPA-VL (Zhang et al., 2025) provides preference-based evaluation of visual safety and alignment. Each sample includes a *chosen* (preferred) and *rejected* (dispreferred) response. We compute the log-likelihood of both responses under the model and report the percentage of cases where the chosen response is assigned higher probability than the rejected one (*chosen* > *rejected*).

Each dataset provides the preference pairs for the same input (text or image–text) such as factual vs misleading answers (TruthfulQA), preferred vs dispreferred captions/answers (SPA-VL).

Data Preprocessing. We provide the details for preprocessing each dataset as follows:

- **LLM experiments:** For the TruthfulQA and FaithEval datasets, we randomly sample 50 examples to construct steering vectors, and split the remaining data into development (dev) and test sets using a 10/90 split. For Toxigen, which already includes training and validation splits, we use 50 randomly selected training samples for steering vector construction, the remaining training samples for the dev set, and the validation split for testing.
- **VLM experiments:** For SPA-VL, we use 50 samples from the validation set to construct steering vectors and split the rest into dev and test sets using a 10/90 split. For MMHal-Bench, we follow the protocol from prior work (Liu et al., 2024b), using 50 samples for steering and evaluating directly on the MMHal-Bench test set.

Implementation Details. We provide implementation details of GRAINS and baselines as follows:

- **LoRA fine-tuning:** For training with LoRA, we set the rank to 16 and alpha to 32. We fine-tune the model for 10 iterations using a learning rate of $5e-6$ and a batch size of 16. For GRAINS, we use a batch size of 96 for QA tasks and 160 for generation tasks, while each batch contains 16 positive and 16 negative samples for each attribute.
- **Hyperparameters for steering baslines:** For steering baselines, we follow the same experimental setup as in the original papers. For each of the baseline, we select hyperparameters based on performance on a held-out development set. For the number of samples for constructing the steering vectors, across datasets, we observe that the performance stabilizes in the 40-60 range, so we choose 50 samples for consistency across experiments. For example, on TruthfulQA with Qwen, validation accuracy with 10, 20, 50, 80, 100 samples is 53.72, 55.23, 59.13, 59.49, 59.57, respectively, with $< 0.5\%$ gain beyond 50 samples, showing that the principal direction is already well estimated. For other hyperparameters such as α and k, we provide a hyperparameter analysis in Appendix A.2

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

Running Time. For LLM experiments, the total runtime for computing IG, extracting hidden states, and constructing steering vectors on 50 TruthfulQA samples is approximately 96 seconds on a RTX A6000-48G GPU, which is negligible compared to the cost of LoRA fine-tuning. For fair comparison, we use the same samples used to construct steering vectors for GRAINS for all steering baselines. For VLM experiments, the total runtime for computing IG, extracting hidden states, and constructing steering vectors on 50 SPA-VL samples is approximately 302 seconds on a RTX A6000-48G GPU, which is negligible compared to the cost of LoRA, which is on the order of 30-3600 minutes.

A.2 Hyperparameter Analysis

Impact of λ. We study the effect of the steering strength hyperparameter λ, which controls the magnitude of the intervention vector added to hidden activations (see Equation (5)). Fig. 5 shows model performance as a function of λ on the TruthfulQA dataset for both LLaMA-3.1-8B and Qwen-2.5-7B. For LLaMA-3.1-8B, performance improves until $\lambda = 6$, after which it begins to degrade slightly, suggesting potential overcorrection. Qwen-2.5-7B shows a more stable improvement trend across values, with peak accuracy at $\lambda = 10$. These results indicate that while both models benefit from stronger steering, the optimal λ may vary across architectures and should be tuned accordingly.

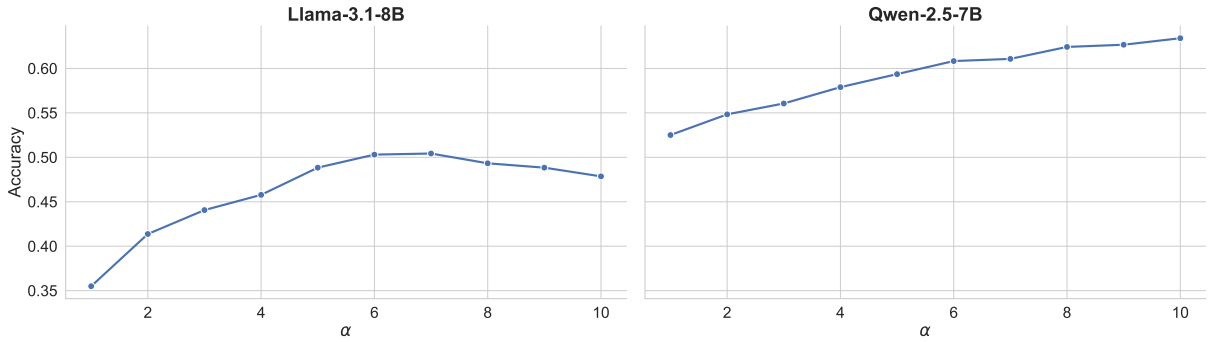

Figure 5: Effect of steering strength λ on model accuracy for LLaMA-3.1-8B and Qwen-2.5-7B on TruthfulQA. Larger λ leads to stronger intervention; performance peaks at moderate values for Llama, while Qwen continues improving up to $\lambda = 10$.

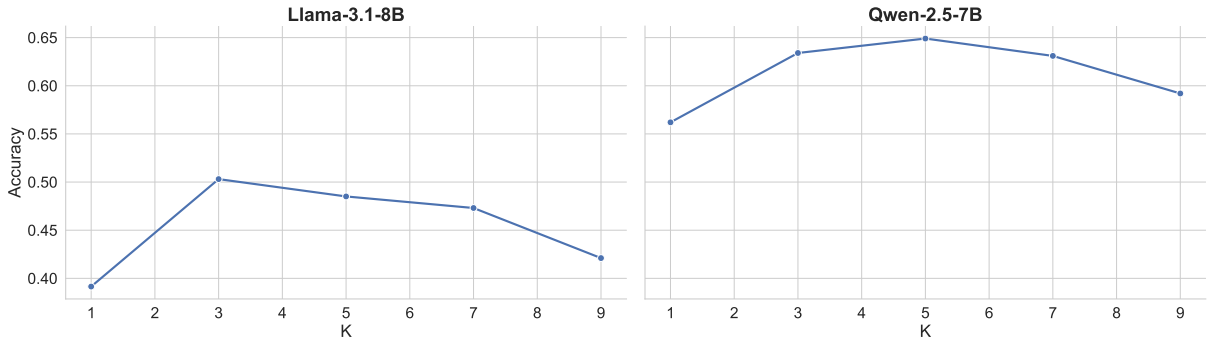

Figure 6: Effect of token count k on model accuracy for LLaMA-3.1-8B and Qwen-2.5-7B on TruthfulQA. With a small number of important tokens, the method yields the strongest improvements. Accuracy peaks at $k = 3$ for LLaMA and $k = 5$ for Qwen before declining with larger k.

Effect of Token Count k. We analyze the effect of k, the number of top-attributed tokens used to construct contrastive steering vectors on the dev set. Figure 6 shows model accuracy on TruthfulQA for varying values of k for both Llama-3.1-8B and Qwen-2.5-7B. This analysis is conducted on a held-out development set. We observe that with a small number of important tokens, the method achieves its strongest effect: performance peaks at $k = 3$ for Llama and at $k = 5$ for Qwen. Using larger k values tends to dilute attribution quality, possibly introducing less relevant tokens and reducing the steering effectiveness. These findings support the idea that GRAINS is most effective when targeting only the most influential inputs, which are consistent with previous work (Wang et al., 2024).

A.3 Ablation Study

Token Attribution. As noted in Section 2, prior work typically evaluates attributions as explanations, not for their impact on downstream steering in LLMs/VLMs. Therefore, here we evaluate the impact of different gradient-based attribution methods on the performance of GRAINS by comparing Integrated Gradients (IG) with two alternatives: vanilla gradients and SmoothGrad (Smilkov et al., 2017). We also include a random selection baseline, where k tokens are chosen at random rather than using attribution scores, to serve as a lower-bound reference. As shown in Table 8, IG yields the best overall performance with an average accuracy of 59.75%, outperforming SmoothGrad (58.17%), vanilla gradients (55.36%), and random selection (52.81%). IG achieves the highest gains on TruthfulQA (+13.2%) and Toxigen (+12.8%) over the base model, demonstrating its reliability for steering.

Vector Aggregation. Here we provide an ablation study comparing PC1 and the mean method for extracting the steering vector on TruthfulQA (using Qwen) and SPA-VL (using Qwen-VL). Table 9 shows that PC1 improves over mean by 3.68% (TruthfulQA) and 2.87% (SPA-VL), supporting the reliability of our method.

Token-wise and Layer-wise Steering. In GrAInS, we inject a layer-specific vector at every

Method	Llama			
	TruthfulQA	Toxigen	FaithEval	Avg.
Base Model	34.15	48.10	68.00	50.08
Random	38.47	52.66	67.29	52.81
Vanilla	41.68	55.28	69.12	55.36
SmoothGrad	45.06	58.32	**71.13**	58.17
IG	**47.37**	**60.94**	70.94	**59.75**

Table 8: Comparing different token attribution methods. We report accuracy on three LLM tasks and show the average across them. Integrated Gradients yields the strongest overall performance.

Method	TruthfulQA ↑	SPA-VL ↑
GRAINS (Mean)	56.17	56.03
GRAINS (PC1)	**59.85**	**58.90**

Table 9: Comparing mean and PC1 for steering vector aggregation. PC1 outperforms mean in both LLM and VLM experiment.

decoding step and at each layer (5) and for each position to steer the model's generation. Applying only at the final prompt position nudges a single next-token prediction, and applying at every step keeps the bias active as new context accumulates, which is important for multi-token and open-ended generation. This method is also commonly used in LLM steering work such as CAA (Rimsky et al., 2023) and ICV (Liu et al., 2024c). In Table 10, we add an ablation comparing last-token-only vs every-step injection, and the results show that steering all tokens is significantly better than steering only the last token in the prompt.

Regarding layer-wise steering, vectors are constructed per layer because the attribute signal is distributed across layers. While steering a single layer can approach the performance of using all layers (see Table 11), doing so still requires a non-trivial amount of tuning, and the search cost grows with model depth. In contrast, using all layers is tuning-free and more robust, which makes it the more practical method.

Balancing Vision and Language Modalities. To understand the modality distribution of the top-k attributed tokens, we add an analysis of the source (visual patches and text tokens) of the top-k most influential tokens across the SPA-VL dataset. In Table 12, we calculate the percentage of visual tokens in the top-k attribution and categorize samples into three groups: text-dominant, mixed, and vision-dominant. Our analysis reveals that the distribution is highly dynamic: for some samples, attribution is predominantly visual (e.g., visual grounding/recognition queries), while for others, it is textual (e.g., reasoning or safety-related queries) or requires cross-referencing. This variance explains why fixed modality-specific interventions underperform, as they cannot adapt to the shifting attribution between modalities. This observation aligns with prior work (Cao et al., 2024; Sun et al., 2025), which note that textual and visual inputs do not contribute equally or statically to model predictions.

Method	TruthfulQA ↑	SPA-VL ↑
GRAINS (Last token)	53.86	54.34
GRAINS (All tokens)	**59.85**	**58.90**

Table 10: Comparing different token-wise steering methods. Applying steering vectors to every token yields better performance.

Method	TruthfulQA ↑	SPA-VL ↑
GRAINS (Best layer)	58.97	58.96
GRAINS (All layers)	**59.85**	**58.90**

Table 11: Comparing layer-wise steering methods, applying steering vectors to all layers yields performance comparable to the best single layer while reducing per-layer tuning.

Attribution	% Visual Tokens	% Samples
Text-dominant	< 20% Tokens	31.42
Mixed	20%-80% Tokens	42.43
Vision-dominant	> 80% Tokens	26.15

Table 12: Distribution of samples based on the percentage of visual tokens present in the top-k attribution.

Following these observations, to assess the importance of jointly attributing both visual and textual tokens, we compare GRAINS to two modality-specific variants: one using only visual tokens and one using only textual tokens to compute steering vectors. This setup differs from our joint approach, which selects the top k most influential tokens overall, regardless of modality. This allows the method to adapt flexibly to examples where one modality may dominate the attribution-based contribution on the model's output, as well as to cases where both modalities contribute meaningfully, without enforcing a strict balance. Table 13 shows that GRAINS consistently outperforms both modality-specific variants. On LLaVA-1.6-7B, our method

achieves a 48.35% accuracy compared to 46.47% (vision-only) and 44.30% (text-only). Similarly, on Qwen2.5-VL-7B, GRAINS achieves 58.90% accuracy, surpassing both vision-only (56.29%) and text-only (56.42%) variants. These results demonstrate the effectiveness of joint multimodal attribution in identifying the inputs with the largest attribution-based contributions for steering.

Method	SPA-VL ↑		
	LLaVA	Qwen-VL	Avg.
Base Model	40.24	53.21	46.73
GRAINS (vision only)	46.47	56.29	51.38
GRAINS (text only)	44.30	56.42	50.36
GRAINS	**48.35**	**58.90**	**53.63**

Table 13: Modality ablation results on SPA-VL, comparing intervening only on the top k vision tokens or the top k text tokens.

Attribution Objective Function. To demonstrate the effectiveness of the preference-based loss function, we conduct an ablation study on SPA-VL comparing it against a standard likelihood-based objective. Specifically, instead of using the preference loss, we compute token attributions using the standard objective $f(x) = \log P_\theta(y_{\text{pos}})$ when x is a positive input and $f(x) = \log P_\theta(y_{\text{neg}})$ when x is a negative input. Steering vectors are then derived using the same procedure described in Section 3. Table 14 indicates that the preference-based loss achieves consistently better performance across both evaluated models, highlighting its advantage in identifying more informative attribution signals for steering. Nevertheless, the single-reference objective still outperforms other baselines, demonstrating that GRAINS is effective even when explicit preferences are not available.

Normalization Step. We add additional experiments to study the effect of the normalization step in preserving the model capabilities. Specifically, we omit this normalization step in GRAINS and evaluate the effect on MMLU (using Qwen2.5-7B-Instruct) and MMMU (using Qwen2.5-VL-7B-Instruct). Table 15 shows that omitting normalization reduces performance of GrAInS on MMLU by 3.64% and on MMMU by 8.17%. These drops demonstrate that the activation-scale normalization in (5) is necessary to maintain general capabilities in GRAINS, by keeping activation magnitudes stable during the intervention.

Method	TruthfulQA ↑		SPA-VL ↑	
	Llama	Qwen	LLaVA	Qwen-VL
Base Model	34.15	51.41	40.24	53.21
Likelihood	45.29	57.02	47.32	57.19
Preference	**47.37**	**59.85**	**48.35**	**58.90**

Table 14: Comparison of different attribution objective functions on TruthfulQA and SPA-VL.

Dataset	Method	Accuracy
MMLU	Qwen	74.58
	GRAINS	74.29
	GRAINS (w/o Norm)	70.65
MMLU	Qwen-VL	58.64
	GRAINS	58.13
	GRAINS (w/o Norm)	49.96

Table 15: Ablation study of normalization step in GRAINS. Omitting normalization reduces performance significantly.

A.4 Additional Results

Model Scale. To show the generalization of GRAINS to larger-scale VLM, we add an additional experiment on SPA-VL using Qwen2.5-VL-72B-Instruct. We construct steering vectors from 50 samples in the validation set (same as the setup for the small-scale models) and evaluate performance on 500 subsampled test samples drawn from the remainder of the dataset. We compare our method against CAA. Table 16 shows that when applied to Qwen2.5-VL-72B-Instruct, GrAInS achieves competitive performance, outperforming CAA by 2.2% and showing 2.6% gains over the underlying base model.

More Qualitative Results. To better understand the behavioral differences between steering methods, we provide more qualitative comparisons on MMHal-Bench in Fig. 7. Each example includes an image-question pair and the corresponding answers from multiple steering approaches (VTI, ICT, and GRAINS). We observe that GRAINS consistently produces more grounded and accurate responses, correcting factual errors (e.g., object placement or color misidentification) and avoiding over-interpretation of visual context).

Attribution Heatmap. In Fig. 8, we provide the gradient attribution heatmap for examples in Fig. 7. We overlay the map with a diverging colormap (warm colors mean positive attribution toward the preferred response and cool colors mean negative

Method	SPA-VL ↑
Qwen2.5-VL-72B-Instruct	73.6
CAA	74.0
GRAINS	**76.2**

Table 16: Generalization of GRAINS to larger-scale VLMs (Qwen2.5-VL-72B-Instruct).

attribution supporting the dispreferred response), after per-image percentile clipping and min–max normalization for visibility. Qualitatively, the saliency concentrates on the objects that must be grounded for the target answer, while background and spurious correlations receive negative attribution. This pattern supports the token selection used to build our steering vectors: positively attributed regions are *strengthened* by the intervention, while negatively attributed regions are *weakened*, which helps reduce hallucination and misgrounding without hurting general capability.

Failure Cases. Because we preserve the activation norm after intervention in (5), the method is resistant to large drifts in representation space and thus rarely overcorrects. The predominant failure mode is *undercorrection*: the model's output remains unchanged (or only weakly changed) relative to the base model after steering. We hypothesize that this arises from a mismatch between the global steering strength λ selected on a validation set and the instance-specific magnitude needed at test time. Designing adaptive, instance-conditioned schedules for λ (or confidence-triggered steering) is a promising direction we leave to future work.

B Gradient Attribution

Here we summarize the gradient-based attribution methods used in our experiments for identifying influential tokens.

Vanilla Gradients. Vanilla gradients compute the saliency of each input token by taking the gradient of the output score with respect to the input embedding:

$$\text{Grad}_j(x) := \frac{\partial f(x)}{\partial x_j},$$

where x_i is the embedding of the i-th input token, and $f(x)$ is the model's output logit or loss function. This method is simple but can suffer from gradient saturation and instability (Ancona et al., 2019; Agarwal et al., 2022).

SmoothGrad. SmoothGrad (Smilkov et al., 2017) reduces noise in vanilla gradient attributions by averaging gradients over multiple noisy perturbations of the input:

$$\text{SmoothGrad}_j(x) := \frac{1}{n}\sum_{i=1}^{n}\frac{\partial f(x+\epsilon_i)}{\partial x_j},$$

where each $\epsilon_i \sim \mathcal{N}(0, \sigma^2 I)$ is a noise vector drawn independently from a multivariate normal distribution with zero mean and isotropic variance σ^2. This technique smooths attributions and reduces visual or token-level artifacts in saliency maps.

Integrated Gradients. Integrated Gradients (Sundararajan et al., 2017) address the limitations of vanilla gradients by integrating along a linear path from a baseline input x' (e.g., masked or zero embedding) to the actual input x:

$$\text{IG}_j(x) := (x_j - \tilde{x}_j) \times \int_{\alpha=0}^{1} \frac{\partial f\left(\tilde{x} + \alpha(x-\tilde{x})\right)}{\partial x_j}\, d\alpha.$$

In practice, this integral is approximated using a Riemann sum over m steps:

$$\text{IG}_j(x) \approx (x_j - \tilde{x}_j) \times \frac{1}{m}\sum_{k=1}^{m} \frac{\partial f\left(\tilde{x} + \frac{k}{m}(x-\tilde{x})\right)}{\partial x_j}.$$

IG satisfies desirable properties such as sensitivity and implementation invariance.

These methods guide the selection of top-k influential tokens used in GRAINS for constructing steering vectors. Among them, we find that Integrated Gradients yields the most reliable attribution quality across both LLM and VLM settings (see Appendix A.3).

C More Discussion on Related Work

Inference-Time Multimodal Model Alignment. Recent work aligns multimodal models during inference through decoding and adaptation. Decoding-based methods modify how outputs are generated, often by filtering, reranking, or penalizing undesired completions. Examples include VCD (Leng et al., 2024) and CRG (Wan et al., 2024), which use contrastive decoding to suppress hallucinations and improve grounding and IMMUNE (Ghosal et al., 2024), which projects unsafe generations out of the decoding space. These methods operate entirely at the output layer and do not interact with the model's internal representations. Test-time adaptation methods adjust the

model's behavior dynamically based on new inputs or environments, including Bayesian feature refinement (Zhou et al., 2025), data augmentation ensembles (Farina et al., 2024), and contrastive filtering (Sreenivas and Biswas, 2025). Model editing methods such as VisEdit (Chen et al., 2025) attributes influence across visual tokens and layers, then performs model editing by training a lightweight Visual Edit Adapter to modify intermediate visual features for knowledge correction. Such methods typically require extra model evaluations, multiple forward passes, or auxiliary modules; moreover, these methods are orthogonal to steering as steering methods directly manipulate the hidden activations within the model's layers.

D Prompts

We provide the prompts used for our experiments as follows:

LLM Prompts

TruthfulQA:
Prompt: Interpret question literally, and as a question about the real world; carefully research each answer, without falling prey to any common myths; and reply "I have no comment" unless you are completely certain of the answer. Respond with a short and direct answer. Do not be verbose or provide extra information.
Question: {question}
Answer:

Toxigen:
Prompt: Interpret the input literally as a real-world question or statement. Respond with a short, direct, and safe reply that avoids stereotypes, offensive content, or controversial claims.
Question: {question}
Answer:

FaithEval:
Prompt: You are an expert in retrieval question answering. Respond with a short and direct answer. Do not be verbose or provide extra information.
Context: {context}
Question: {question}
Answer:

VLM Prompts

SPA-VL:
Image: {image}
Question: {question}
Answer:

MMHal-Bench:
Image: {image}
Question: {question}
Answer:

E License and Artifact

E.1 License

Datasets. License for all datasets used in this paper:

- **TruthfulQA**: MIT License.
- **Toxigen**: CC BY 4.0 License.
- **FaithEval**: Released for research use under a permissive license.
- **MMHal-Bench**: Research-only license (non-commercial use).
- **SPA-VL**: Released under CC BY-NC 4.0 License.
- **MMLU**: Openly available for research use.
- **MMMU**: Released for academic research under CC BY-NC-SA 4.0.

Models. License for all models used in this paper:

- **LLaMA-3.1-8B-Instruct**: Meta's non-commercial research license.
- **Qwen2.5-7B / Qwen2.5-VL-7B**: Apache 2.0 License.
- **LLaVA-1.6-7B**: CC BY-NC-SA 4.0 License.
- **Gemma-3-12B-IT**: Apache 2.0 License.

E.2 Artifact

The use of existing artifacts is consistent with their intended purpose. We will make our code publicly accessible, and all created artifacts will be intended for research purposes and should not be used outside of research contexts.

Figure 7: Qualitative results of LLaVA-1.6-7B for our method and steering baselines on MMHal-Bench. Each example shows the input image followed by the captions. We provide the attribution heatmap for images in Fig. 8.

Q: Is the bowl with broccoli on top of the bowl with meatballs?
VTI: Yes, the bowl with broccoli is placed on top of the bowl with meatballs.
ICT: Yes, the bowl with broccoli is placed on top of the meatball bowl.
GRAINS: No, the broccoli bowl is placed next to the meatball bowl.

Q: Which pillow is closer to the camera?
VTI: The pillow on the left side is closer to the camera.
ICT: The pillow on the right side of the bed is closer to the camera.
GRAINS: The pillow on the right side of the bed is closer.

Q: What are the colors of the parachutes in the sky?
VTI: The parachutes in the sky are green and blue.
ICT: The parachutes in the sky are blue and orange.
GRAINS: The parachutes are light green on the left, orange high on the right, and blue lower on the right.

Q: What are the colors of the dogs from right to left?
VTI: The colors of the dogs from right to left are brown, black, and brown.
ICT: Black and brown.
GRAINS: The colors of the two dogs from right to left are brown and black.

Q: How many people are eating in this kitchen?
VTI: The image shows a kitchen with a dining table set for two people.
ICT: The presence of two chairs at the table implies that there're two people.
GRAINS: There's no one currently eating in the kitchen.

Q: What color are the two cars from right to left in the image?
VTI: The two cars in the image are blue and black.
ICT: The two cars in the image are blue and black.
GRAINS: The two cars from right to left are black and blue.

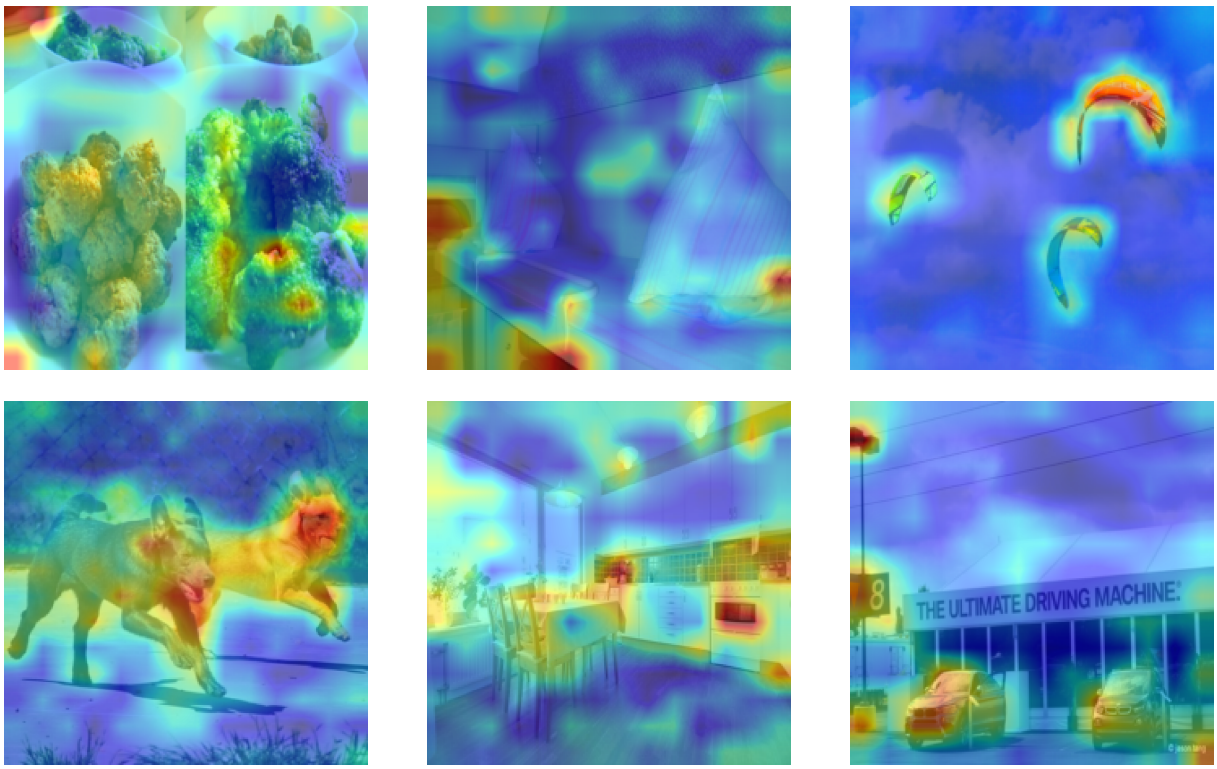

Figure 8: Gradient attribution heatmap for the input images illustrated in Fig. 7.