Correction Space Steering for Hallucination Mitigation in Large Vision-Language Models

Songbo Yang[1,2] **Shuliang Liu**[3,4] **Sihang Jia**[3,4]

Xuming Hu[3,4]*
[1]University of Science and Technology of China
[2]Zhongguancun Academy
[3]The Hong Kong University of Science and Technology (Guangzhou)
[4]The Hong Kong University of Science and Technology
songboyang@mail.ustc.edu.cn

Abstract

Large Vision-Language Models (LVLMs) are prone to hallucinations, producing responses that are inconsistent with visual inputs. While inference-time activation steering provides a lightweight solution without retraining, existing methods are either limited by coarse global steering vectors or rely on unconstrained full-dimensional vector prediction, which can produce unreliable vectors for unseen hallucination patterns. In this paper, we propose **Correction Space Steering** (CSS), a robust activation steering framework that predicts correction coordinates in a learned low-dimensional correction space. We first conduct an empirical analysis of hallucination-related activation differences and find that they are predictive of hallucination types, form type-aware geometric structures, and exhibit low effective ranks in most layers. We further provide a theoretical analysis showing that these activation differences can be approximated by a low-dimensional subspace. Based on this insight, CSS learns a low-dimensional correction space from activation differences and trains an online predictor to infer input-specific correction coordinates from inference-time hidden states. The resulting correction vectors are then reconstructed within the correction space and applied to steering activations for hallucination mitigation. Experiments on standard hallucination benchmarks show that CSS effectively reduces hallucinations without retraining LVLMs and outperforms existing inference-time mitigation methods. Code is available at https://github.com/songboyangy/CSS.

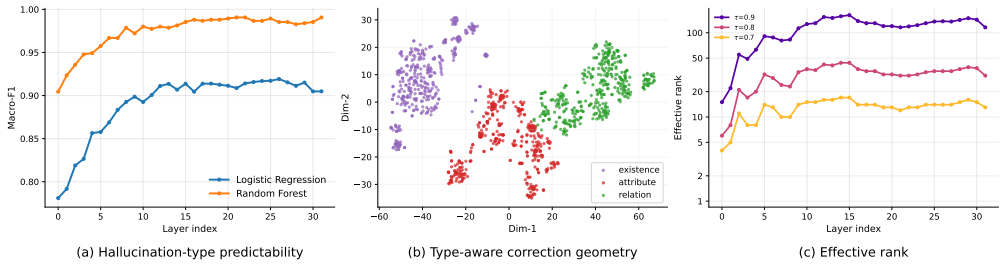
1 Introduction

Large Vision-Language Models (LVLMs) have achieved remarkable performance on a wide range of multimodal tasks, including visual question answering, image captioning, and multimodal reasoning [1–3]. Despite this progress, LVLMs remain prone to hallucinations, producing responses that are inconsistent with the input image or making incorrect claims about object attributes, quantities, and relations [4, 5]. Such errors undermine the factual reliability of LVLM outputs and limit their deployment in high-stakes scenarios such as autonomous driving [6], robotics [7], and other safety-critical domains [8, 6].

Recent hallucination mitigation efforts have shifted from costly retraining [9–11] to lightweight inference-time interventions, which can be broadly divided into decoding-based [5, 12, 13] and

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



(a) Hallucination-type predictability (b) Type-aware correction geometry (c) Effective rank

Figure 1: Preliminary activation-difference analysis of LLaVA-1.5 on the AMBER benchmark. (a) Hallucination-related activation differences are highly predictive of hallucination types across layers. (b) t-SNE visualization shows that different hallucination types form distinguishable regions in activation-difference space. (c) Effective-rank analysis of the activation-difference matrix reveals a consistently low-rank structure across layers, suggesting that hallucination-related correction signals concentrate in a low-dimensional structure.

steering-based approaches [14–16]. Decoding-based methods reduce hallucinations by adjusting token distributions during generation, but may distort the original generation distribution and degrade response quality. In contrast, steering-based methods directly modify hidden activations during inference, thereby providing a more targeted way to correct internal representations associated with hallucination. Early steering-based methods [14] typically rely on a single global steering vector shared across all inputs. Although simple and efficient, such global interventions are often too coarse to capture the context-dependent nature of hallucinations. Recent methods [16, 15] improve adaptivity by using input-aware steering. DMAS [16] retrieves category-specific prototype vectors from a database, but its intervention remains restricted to a finite set of candidate steering vectors. L2S [15] further improves adaptivity by learning an input-dependent mapping that predicts sample-specific steering vectors from input representations. While more flexible, this full-dimensional prediction is learned from the training distribution and can produce unreliable steering vectors when the test input involves unseen hallucination patterns.

To better understand how to design adaptive and robust correction methods, we conduct a preliminary analysis of hallucination-related activation differences in LLaVA-1.5. Specifically, using the AMBER benchmark [17], we construct paired truthful and hallucinated samples and extract their internal activations to derive activation differences. As shown in Figure 1, these activation differences are predictive of hallucination types, form type-aware geometric structures, and exhibit low effective ranks in most layers. Motivated by these observations, we further provide a theoretical analysis showing that hallucination-related activation differences can be approximated by a low-dimensional subspace. Based on this result, we introduce a *correction space*, which models the observed low-dimensional structure of hallucination-related activation differences while preserving the hallucination-related patterns encoded in these signals.

Based on the above empirical and theoretical findings, we propose Correction Space Steering (CSS), an inference-time activation steering framework that performs adaptive correction through coordinate prediction in a learned correction space. CSS consists of four components. First, Activation Difference Construction constructs hallucination-related activation differences from paired truthful and hallucinated samples. Second, Correction Space Learning learns a low-dimensional correction space that captures the structure of these activation differences while preserving hallucination-related patterns. Third, Coordinate Predictor Learning trains an online predictor to infer input-specific coordinates from inference-time hidden states. Finally, Inference-Time Steering reconstructs correction vectors from the predicted coordinates and applies them to hidden activations for hallucination mitigation.

Our contributions are summarized as follows:

- We reveal that hallucination-related activation differences exhibit type-aware regularity in LVLMs: they are predictive of hallucination types, form distinguishable geometric structures, and show low effective ranks in most layers. We further provide a theoretical analysis showing that such activation differences can be approximated by a low-dimensional subspace.

- We propose **Correction Space Steering** (CSS), an inference-time activation steering framework that formulates adaptive hallucination correction as coordinate prediction in a learned low-dimensional correction space, rather than unconstrained full-vector prediction in the original hidden space.
- We empirically validate CSS on standard hallucination benchmarks, demonstrating that it effectively mitigates hallucinations without retraining the LVLM and achieves strong performance compared with existing inference-time mitigation methods.

2 Related Work

2.1 Large Vision-Language Models

Large Vision-Language Models (LVLMs) [1–3] extend large language models with visual perception capabilities and have demonstrated strong performance across diverse vision-language tasks, such as image captioning, visual question answering, and visual grounding [18–20]. Most LVLMs adopt a modular architecture consisting of a visual encoder, a modality alignment module, and a language decoder, where the modality alignment module bridges visual representations and the language model to enable cross-modal understanding and generation. Based on the design of this alignment module, existing LVLMs can be broadly categorized into projection-based and query-based models. Projection-based models, such as LLaVA [1], Shikra [21], and Qwen2.5-VL [22], use linear layers, MLP projectors, or vision-language mergers to connect visual features with large language models. Query-based models, such as BLIP-2 [23], InstructBLIP [2], and MiniGPT-4 [24], employ learnable query tokens to extract and align visual information with the language model. In this work, we evaluate our approach using three representative LVLMs, including two widely used models, LLaVA-v1.5 and InstructBLIP, as well as a stronger recent backbone, Qwen2.5-VL.

2.2 Hallucination Mitigation for LVLMs

Hallucination in LVLMs refers to content inconsistent with or unsupported by visual inputs, including nonexistent objects, incorrect attributes, and inaccurate relations [4]. Existing mitigation methods include training-based and inference-time approaches. Training-based methods improve visual faithfulness through high-quality data [25–27] or specialized optimization objectives [10, 28], but require additional model optimization and computational resources. Inference-time methods keep the LVLM backbone frozen. Decoding-based approaches adjust output distributions during generation [5, 12, 13], use textual perturbations [29], or iteratively revise responses [30]. Their additional forward passes or generation rounds can increase inference cost. Activation steering offers another inference-time approach by intervening directly in hidden representations. Global-vector methods such as VTI [14] apply shared steering directions across inputs. VTI, for example, derives its intervention from visual features obtained with image perturbations. Such shared directions provide a simple correction mechanism but have limited flexibility for input-specific hallucination patterns. Adaptive methods, including DMAS [16], L2S [15], and VLI [31], introduce input-dependent interventions. In particular, L2S learns a mapping from input representations to sample-specific steering vectors. This increases adaptivity while making the structure and constraints of the predicted vectors an important design consideration. Related projection-based methods, such as Nullu [32], identify hallucination-associated directions for suppression. CSS learns a correction space from paired truthful and hallucinatory activation differences and predicts input-dependent coordinates within that space. This formulation combines adaptive correction with an explicit low-dimensional constraint on the reconstructed additive steering vectors.

3 Preliminary

In this section, we theoretically analyze the low-dimensional structure of hallucination-related activation differences. We show that low effective rank implies a compact low-dimensional approximation with bounded reconstruction error, providing a theoretical basis for coordinate-based correction. Based on this result, CSS learns a correction space $\mathcal{S}$ and performs steering within this space, rather than predicting unconstrained full-dimensional correction vectors. We present the key result here and leave the full analysis and proofs to the appendix A.

Background. Given an image-question pair $X = (I, q)$, an LVLM produces hidden representations $h_l \in \mathbb{R}^d$ across transformer layers during decoding. Steering-based methods intervene at a selected layer l^* by adding a correction vector:

$$\tilde{h}_{l^*} = h_{l^*} + \eta\delta, \tag{1}$$

where η is a steering-strength coefficient and $\delta \in \mathbb{R}^d$ is the steering vector. Given paired non-hallucinatory and hallucinatory response trajectories, we define the activation difference at layer l as

$$\Delta h_i^{(l)} = h_{i,l}^{+} - h_{i,l}^{-}, \tag{2}$$

where $h_{i,l}^{+}$ and $h_{i,l}^{-}$ are hidden states induced by the non-hallucinatory and hallucinatory responses, respectively. The set of activation differences $\Delta\mathcal{D}_l = \{\Delta h_i^{(l)}\}_{i=1}^{N}$ characterizes hallucination-related shifts in hidden space.

Effective Rank. The empirical analysis in Section 1 shows that activation differences often have low effective rank, indicating that most of their energy is concentrated in a small number of directions. Given N paired samples, we stack activation differences into $\Delta H^{(l)} = [\Delta h_1^{(l)}, \ldots, \Delta h_N^{(l)}]^\top \in \mathbb{R}^{N\times d}$. Let the singular values of $\Delta H^{(l)}$ be $\sigma_1^{(l)} \geq \cdots \geq \sigma_r^{(l)} > 0$, where $r \leq \min(N, d)$. For an energy threshold $\tau \in (0, 1)$, the τ-effective rank is defined as

$$\mathrm{erank}_\tau(\Delta H^{(l)}) = \min\left\{ K : \frac{\sum_{j=1}^{K}(\sigma_j^{(l)})^2}{\sum_{j=1}^{r}(\sigma_j^{(l)})^2} \geq \tau \right\}. \tag{3}$$

Proposition 3.1 (Effective Rank Implies Low-Dimensional Approximation)**.** *Let* $K = \mathrm{erank}_\tau(\Delta H^{(l)})$*, and let* $B_{l,K}^{\mathrm{svd}} \in \mathbb{R}^{d\times K}$ *contain the top* K *right singular vectors of* $\Delta H^{(l)}$*. Define*

$$\Delta H_K^{(l)} = \Delta H^{(l)} B_{l,K}^{\mathrm{svd}} B_{l,K}^{\mathrm{svd}\top}. \tag{4}$$

Then $\Delta H_K^{(l)}$ *is the optimal rank-*K *approximation of* $\Delta H^{(l)}$ *under the Frobenius norm, and*

$$\frac{\|\Delta H^{(l)} - \Delta H_K^{(l)}\|_F^2}{\|\Delta H^{(l)}\|_F^2} \leq 1 - \tau. \tag{5}$$

Consequently, each activation difference can be decomposed as

$$\Delta h_i^{(l)} = B_{l,K}^{\mathrm{svd}} \alpha_i^{(l)} + \epsilon_i^{(l)}, \tag{6}$$

where $\alpha_i^{(l)} = B_{l,K}^{\mathrm{svd}\top} \Delta h_i^{(l)}$ *and* $\epsilon_i^{(l)}$ *is the residual outside the SVD subspace.*

Proposition 3.1 formalizes the implication of the effective-rank observation in Section 1. A small effective rank means that activation differences admit a compact coordinate representation, supporting low-dimensional correction instead of unconstrained full-dimensional steering.

From SVD Subspace to Correction Space. The SVD subspace above is optimal for linear reconstruction, but it is not necessarily the desired correction space for steering. Section 1 shows that activation differences are not only low-rank, but also predictive of hallucination types and organized into type-aware geometric structures. Therefore, a useful correction space should preserve such hallucination-type structure, rather than merely minimize reconstruction error.

CSS thus learns a steering-oriented correction space $\mathcal{S}$ instead of directly using $B_{l,K}^{\mathrm{svd}}$. The learned space is optimized to reconstruct activation differences, reduce redundancy among basis directions, and preserve type-aware coordinate structure, which motivates the correction-space learning objective in Section 4.

4 Method

Motivated by the low-dimensional structure of hallucination-related activation differences discussed in Section 3, we propose Correction Space Steering (CSS), an inference-time steering framework

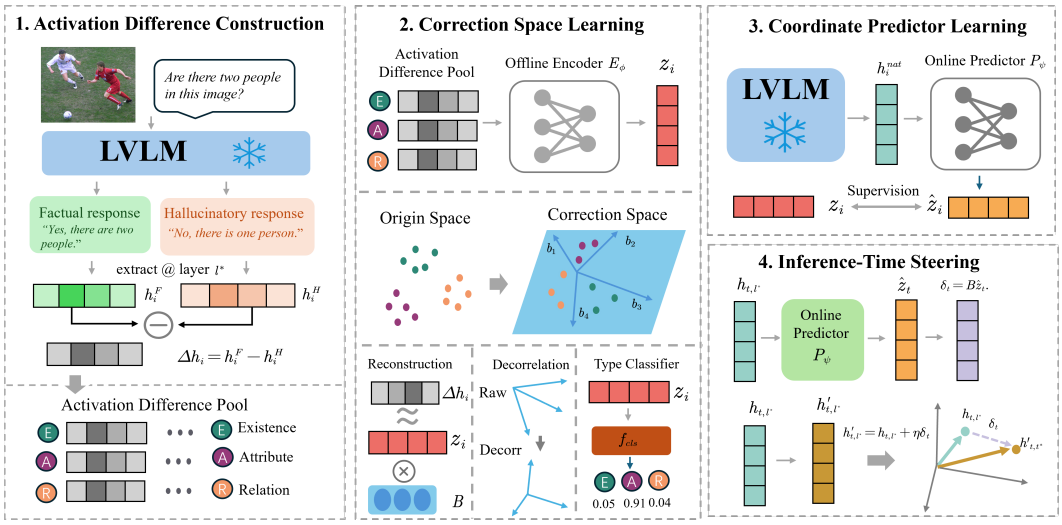


Figure 2: Overview of Correction Space Steering (CSS). CSS first constructs activation differences from paired non-hallucinatory and hallucinatory response trajectories, learns a correction space with an offline coordinate encoder, trains an online coordinate predictor, and applies the reconstructed correction vector for inference-time steering.

for LVLM hallucination mitigation. CSS learns a correction space $\mathcal{S}$ from activation differences and performs adaptive steering by predicting low-dimensional correction coordinates instead of full-dimensional steering vectors. CSS consists of four components. First, **Activation Difference Construction** constructs labeled activation differences from paired non-hallucinatory and hallucinatory response trajectories. Second, **Correction Space Learning** learns the correction space $\mathcal{S}$ and represents activation differences with low-dimensional coordinates. Third, **Coordinate Predictor Learning** trains an online predictor to estimate correction coordinates from hidden states available during generation. Finally, **Inference-Time Steering** reconstructs correction vectors from the predicted coordinates and applies them to the hidden states of the frozen LVLM.

4.1 Activation Difference Construction

To learn the correction space, we use activation differences from paired response trajectories as hallucination-related supervision signals. For each image-question input, we compare the hidden representations induced by a non-hallucinatory response and a hallucinatory response under matched image-question conditions, while keeping the image and question fixed. These differences capture faithfulness-related shifts in hidden space and serve as training signals for correction-space learning.

Following the analysis in the introduction, we use the AMBER benchmark to construct paired response trajectories. For each image-question input $X_i = (I_i, q_i)$, we form a positive trajectory and a negative trajectory by constraining only the initial truthful/hallucinatory Yes/No response token and then allowing subsequent tokens to be generated autoregressively:

$$X_i^+ = (I_i, q_i \| r_i^+), \qquad X_i^- = (I_i, q_i \| r_i^-), \tag{7}$$

where r_i^+ denotes the non-hallucinatory response and r_i^- denotes the hallucinatory response.

We feed the paired inputs into the target LVLM and extract the hidden states at the last token position from the steering layer l^*, denoted by h_{i,l^*}^+ and $h_{i,l^*}^- \in \mathbb{R}^d$. The corresponding activation difference is defined as

$$\Delta h_i = h_{i,l^*}^+ - h_{i,l^*}^-. \tag{8}$$

Since each question in AMBER is annotated with a hallucination-type label, we pair each activation difference with its corresponding type label to form the dataset $\mathcal{D} = \{(\Delta h_i, y_i)\}_{i=1}^N$, where $y_i \in \mathcal{Y}$ denotes the hallucination-type label. The resulting labeled activation differences are used for correction-space learning.

4.2 Correction Space Learning

Given the activation-difference dataset $\mathcal{D}$, we learn a K-dimensional correction space $\mathcal{S} \subset \mathbb{R}^d$ that captures the dominant structure of hallucination-related representation shifts. Let $B = [b_1, b_2, \ldots, b_K] \in \mathbb{R}^{d \times K}$ denote the basis matrix of $\mathcal{S}$, such that $\mathcal{S} = \mathrm{span}(B)$. For each activation difference Δh_i, we seek a low-dimensional coordinate $z_i \in \mathbb{R}^K$ such that

$$\Delta h_i = B z_i + \varepsilon_i, \tag{9}$$

where $Bz_i \in \mathcal{S}$ is the correction-space component and $\varepsilon_i \in \mathbb{R}^d$ denotes the residual outside the correction space.

To amortize coordinate estimation, we introduce an offline coordinate encoder E_ϕ, implemented as a two-layer MLP, which maps each activation difference to its coordinate representation $z_i = E_\phi(\Delta h_i)$.

We jointly learn the correction-space basis B and the encoder E_ϕ by reconstructing activation differences from their low-dimensional coordinates:

$$\mathcal{L}_{\mathrm{recon}} = \frac{1}{N} \sum_{i=1}^{N} \|\Delta h_i - B z_i\|_2^2 . \tag{10}$$

To reduce redundancy among basis vectors and encourage them to capture distinct correction directions, we further introduce a decorrelation regularization term:

$$\mathcal{L}_{\mathrm{decorr}} = \left\| \bar{B}^\top \bar{B} - I \right\|_{F,\mathrm{off}}^2 , \tag{11}$$

where $\bar{B}$ denotes the column-normalized basis matrix, and $\|\cdot\|_{F,\mathrm{off}}^2$ computes the squared Frobenius norm over off-diagonal entries.

The preliminary analysis shows that activation differences exhibit type-aware geometric structure. To preserve such structure in the learned coordinate representation, we introduce an auxiliary type-supervised objective. Specifically, we use the hallucination-type labels provided by the AMBER benchmark, which annotate each sample as existence, attribute, or relation hallucination, as the supervision signal for this objective. Given the coordinate z_i, a classifier f_{cls} predicts the hallucination-type distribution $\hat{p}_i = \mathrm{softmax}(f_{\mathrm{cls}}(z_i))$. The classification loss is defined as

$$\mathcal{L}_{\mathrm{cls}} = -\frac{1}{N} \sum_{i=1}^{N} \log \hat{p}_i[y_i]. \tag{12}$$

The full objective for correction-space learning is

$$\mathcal{L}_{\mathrm{space}} = \mathcal{L}_{\mathrm{recon}} + \lambda_o \mathcal{L}_{\mathrm{decorr}} + \lambda_{\mathrm{cls}} \mathcal{L}_{\mathrm{cls}}. \tag{13}$$

In practice, we optimize this objective in two stages for training stability. We first learn the correction-space basis using $\mathcal{L}_{\mathrm{recon}}$ and $\mathcal{L}_{\mathrm{decorr}}$, and then introduce the type-supervised objective $\mathcal{L}_{\mathrm{cls}}$ to encourage the coordinates to retain hallucination-type information.

4.3 Coordinate Predictor Learning

The coordinates learned above depend on activation differences from paired response trajectories, which are unavailable during inference. To close this train–test availability gap, CSS trains an online coordinate predictor that estimates correction coordinates from hidden states naturally available during generation.

For each training sample i, we run the target LVLM without intervention and extract the hidden state at the final token of the naturally generated response from the steering layer l^*, denoted by $h_i^{\mathrm{nat}} \in \mathbb{R}^d$. We use the offline coordinate $z_i = E_\phi(\Delta h_i)$ as the supervision target and train an online coordinate predictor $P_\psi : \mathbb{R}^d \to \mathbb{R}^K$ to predict the coordinate from the natural hidden state:

$$\hat{z}_i = P_\psi(h_i^{\mathrm{nat}}). \tag{14}$$

The online predictor is optimized with

$$\mathcal{L}_{\mathrm{online}} = \frac{1}{N} \sum_{i=1}^{N} \|\hat{z}_i - z_i\|_2^2 . \tag{15}$$

Table 1: Performance evaluation on CHAIR. CH_S and CH_I denote $CHAIR_S$ and $CHAIR_I$, respectively. The best results are highlighted in **bold**.

Method	LLaVA-1.5			InstructBLIP			Qwen2.5-VL		
	$CH_S \downarrow$	$CH_I \downarrow$	Recall↑	$CH_S \downarrow$	$CH_I \downarrow$	Recall↑	$CH_S \downarrow$	$CH_I \downarrow$	Recall↑
Origin	60.7	17.6	75.8	51.3	13.8	69.8	32.0	9.6	56.5
VCD	54.7	15.2	75.3	50.4	13.9	70.8	30.8	9.2	57.0
ICD	52.6	16.2	72.2	48.6	13.3	72.1	30.3	9.0	57.4
ClearSight	46.6	14.7	70.1	45.7	13.4	72.6	29.1	8.7	58.2
VTI	41.6	12.0	73.1	42.4	12.8	72.9	28.4	8.3	58.6
L2S	46.4	13.0	**76.0**	44.9	12.5	72.5	29.4	8.6	59.3
DMAS	42.4	12.3	74.9	42.2	12.2	72.9	28.3	8.3	60.2
CSS(Ours)	**38.4**	**11.5**	73.6	**40.4**	**11.8**	**73.1**	**27.0**	**7.9**	**61.0**

4.4 Inference-Time Steering

During inference, CSS operates on the frozen target LVLM without paired response trajectories or parameter updates to the backbone. At each decoding step t, given the hidden state h_{t,l^*} at the steering layer l^*, CSS first predicts the correction coordinate using the online predictor $\hat{z}_t = P_\psi(h_{t,l^*})$. The predicted coordinate is then mapped back to the original hidden space through the correction-space basis $\delta_t = B\hat{z}_t$. Finally, CSS applies the reconstructed correction vector to the hidden state:

$$h'_{t,l^*} \leftarrow h_{t,l^*} + \eta\delta_t, \tag{16}$$

where η is a steering-strength hyperparameter. By reconstructing correction vectors within $\mathcal{S}$, CSS constrains sample-specific steering to hallucination-related directions learned from activation differences.

5 Experiments

5.1 Experimental Setup

Benchmarks and Metrics. We evaluate CSS on three benchmarks covering generative hallucination, discriminative hallucination detection, and open-world question answering. More dataset details are provided in the appendix. **(a) CHAIR.** For **generative evaluation**, we use **CHAIR** [33], which measures object hallucination in free-form generation by comparing generated object mentions against ground-truth objects. Following prior work, we report $CHAIR_I$, $CHAIR_S$, and Recall. **(b) POPE.** For **discriminative evaluation**, we use **POPE** [34], a benchmark for object hallucination detection with binary Yes/No judgments. We report Accuracy, Precision, Recall, and F1-score. **(c) MMHal-Bench.** For **open-world question answering**, we use **MMHal-Bench** [35], which evaluates hallucination in open-ended VQA scenarios. Following our evaluation protocol, we use GPT-4o as an automatic judge and report both the **hallucination rate** and the **informativeness** of model responses.

Models and Implementation Details. We conduct experiments on three commonly used LVLM backbones: LLaVA-v1.5, InstructBLIP, and Qwen2.5-VL. CSS is an inference-time steering method that keeps the LVLM backbone frozen and does not update any LVLM parameters. All experiments are implemented in PyTorch. Unless otherwise specified, we follow the same decoding and evaluation protocol for all compared methods. Detailed implementation settings, including the training data used for correction-space learning and hyperparameter choices, are provided in the appendix B.2.

Baseline and Comparative Methods. We further compare CSS with representative inference-time hallucination mitigation methods spanning three paradigms: (i) decoding-based methods, including VCD [5] and ICD [12]; (ii) attention-intervention methods, including ClearSight [36]; and (iii) activation editing methods, including VTI [14], L2S [15], and DMAS [16]. For all baselines, we follow the implementation details and hyperparameter settings reported in the original papers whenever applicable.

Table 2: Performance comparison on POPE. All entries are percentages averaged across Random, Popular, and Adversarial settings. Best results are highlighted in **bold**.

Model	Method	POPE (MSCOCO)				POPE (GQA)			
		Acc. (%)↑	Prec. (%)↑	Rec. (%)↑	F1 (%)↑	Acc. (%)↑	Prec. (%)↑	Rec. (%)↑	F1 (%)↑
LLaVA-1.5	Origin	80.67	87.47	71.78	78.78	79.33	80.69	78.44	79.36
	VCD	83.56	**89.03**	76.89	82.43	82.33	**82.64**	83.11	82.70
	ClearSight	84.22	83.28	**84.83**	83.96	80.11	74.86	90.80	81.82
	VTI	83.78	84.83	83.33	83.89	80.89	76.72	91.33	83.06
	L2S	83.90	84.43	83.21	83.86	79.67	70.88	**95.33**	80.88
	DMAS	83.42	83.95	82.80	83.37	78.95	70.20	94.10	80.40
	CSS (Ours)	**84.51**	85.23	84.52	**84.87**	**84.14**	76.96	93.28	**84.33**
InstructBLIP	Origin	77.78	77.97	77.78	77.84	76.67	73.39	84.89	78.54
	VCD	78.44	78.45	78.67	78.53	78.44	74.68	86.67	80.18
	ClearSight	78.90	78.50	**79.35**	78.92	77.85	74.10	85.80	79.52
	VTI	78.76	78.81	78.69	78.75	79.32	75.41	**87.25**	80.89
	L2S	78.94	79.02	78.97	78.91	77.18	73.92	85.34	79.22
	DMAS	78.55	78.70	78.40	78.55	76.85	73.10	84.90	78.55
	CSS (Ours)	**79.35**	**79.28**	79.21	**79.24**	**80.21**	**76.14**	86.93	**81.17**

5.2 Main Results

Results on CHAIR. Table 1 shows that CSS achieves the strongest performance among the compared methods on the generative CHAIR benchmark. On LLaVA-1.5, CSS reduces CHAIR_S by 22.3 points, from 60.7 to 38.4, and CHAIR_I by 6.1 points, from 17.6 to 11.5. On InstructBLIP, CSS achieves the lowest CHAIR_S of 40.4 and the lowest CHAIR_I of 11.8, while also obtaining the highest Recall of 73.1. Notably, CSS remains effective on the stronger Qwen2.5-VL backbone, reducing CHAIR_S from 32.0 to 27.0 and CHAIR_I from 9.6 to 7.9, while improving Recall from 56.5 to 61.0. These results indicate that CSS is not limited to weaker or more hallucination-prone LVLMs, but can further improve visual faithfulness even when applied to a stronger backbone with substantially lower hallucination rates. Compared with activation-steering baselines such as VTI, L2S, and DMAS, CSS provides a more effective intervention for free-form object hallucination.

Results on POPE. Table 2 demonstrates that CSS achieves the best overall performance on the discriminative POPE benchmark across both MSCOCO and GQA. CSS consistently obtains the highest F1 score under all backbone and dataset settings. In particular, on LLaVA-1.5, the F1 score improves from 78.78 to **84.87** on MSCOCO and from 79.36 to **84.33** on GQA. Similar gains are observed on InstructBLIP, where CSS achieves F1 scores of 79.24 and **81.17** on the two datasets, respectively. For L2S, whose steering-vector predictor is trained on MSCOCO, performance remains competitive on MSCOCO but weakens on GQA. We provide a more detailed analysis of this cross-dataset behavior in Section 5.4.

Results on MMHal-Bench. Figure 3 shows that CSS achieves the best overall performance among the compared methods on MMHal-Bench across all three LVLM backbones. This open-ended setting is more difficult than binary discrimination because responses require both factual grounding and informative generation. The improvement on MMHal-Bench therefore indicates that CSS generalizes beyond object-level hallucination detection and provides effective correction for more complex free-form VQA scenarios.

5.3 Ablation Study

Table 3 examines the roles of correction-space learning, type supervision, and adaptive coordinate prediction in CSS. The **SVD-basis** variant retains the online predictor and inference settings but replaces the learned basis with a truncated-SVD basis. Its improvement over Origin suggests that the dominant directions of activation differences provide useful correction signals, while its gap to CSS supports learning a basis tailored to steering. The **basis-projection** variant instead retains the learned basis and replaces online coordinate prediction with direct projection. Its weaker performance indicates that the basis alone is insufficient to determine an effective input-specific correction, highlighting the contribution of adaptive coordinate estimation. The **w/o** $\mathcal{L}_{\text{cls}}$ variant also increases

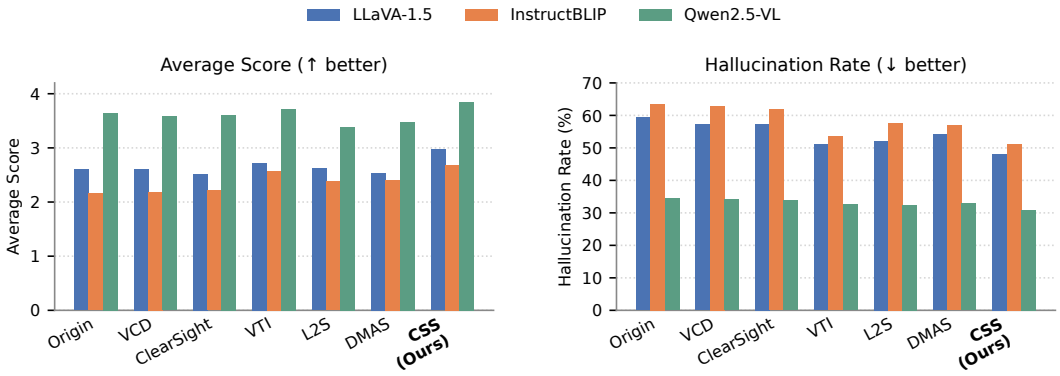


Figure 3: MMHal-Bench comparison across three LVLM backbones. The left panel shows average score (higher is better), and the right panel shows hallucination rate (lower is better). Colors distinguish the three backbones.

Table 3: Ablation study of correction-space learning and online prediction on CHAIR.

Variant	LLaVA-1.5			InstructBLIP		
	$CHAIR_S \downarrow$	$CHAIR_I \downarrow$	Recall↑	$CHAIR_S \downarrow$	$CHAIR_I \downarrow$	Recall↑
Origin	60.7	17.6	75.8	51.3	13.8	69.8
SVD basis	44.8	13.4	73.8	44.7	13.1	72.5
w/o $\mathcal{L}_{\text{cls}}$	40.3	12.4	72.5	42.8	12.6	69.8
Basis projection	45.6	13.8	74.2	44.3	12.8	70.7
CSS (Ours)	**38.4**	**11.5**	73.6	**40.4**	**11.8**	**73.1**

hallucination, suggesting that type supervision provides useful guidance beyond the reconstruction and decorrelation objectives. These trends hold across both backbones and support the complementary contributions of a learned correction basis, type-aware organization, and input-dependent prediction.

5.4 Further Analysis

Generalization of Activation-Difference Structure. To assess whether the observed activation-difference structure generalizes beyond LLaVA-1.5 on AMBER, we extend the analysis to additional backbones and the PhD dataset [37]. As shown in Table 4, activation differences remain predictive of hallucination types across these settings, although predictability and spectral concentration vary across backbones. The PhD results further indicate that these properties extend to a different dataset. This supports the broader applicability of our motivation for correction-space learning, while suggesting that the degree of low-dimensional compactness is model-dependent.

Table 4: Activation-difference structure across backbones and datasets. Macro-F1 and energy-threshold ranks are averaged over model layers. LR and RF denote logistic regression and random forest.

Backbone	Dataset	Type Macro-F1		Energy rank		
		LR	RF	70%	80%	90%
LLaVA-1.5	AMBER	0.891	0.974	12.9	31.5	113.3
InstructBLIP	AMBER	0.852	0.924	49.6	95.9	170.7
Qwen2.5-VL	AMBER	0.783	0.730	20.0	53.6	150.8
LLaVA-1.5	PhD	0.753	0.862	14.5	45.1	120.7

Generalization Analysis. We first evaluate the cross-dataset generalization ability of different steering methods, as shown in Figure 4(a). L2S-COCO and L2S-GQA denote L2S predictors trained on COCO and GQA, respectively. The results show that L2S performs well on the dataset used for training, but suffers a clear drop when transferred to the other dataset. This indicates that unconstrained full-dimensional steering-vector prediction can be sensitive to dataset-specific hallucination patterns. In contrast, CSS maintains strong performance on both COCO and GQA, suggesting that coordinate prediction within the learned correction space $\mathcal{S}$ provides a more robust

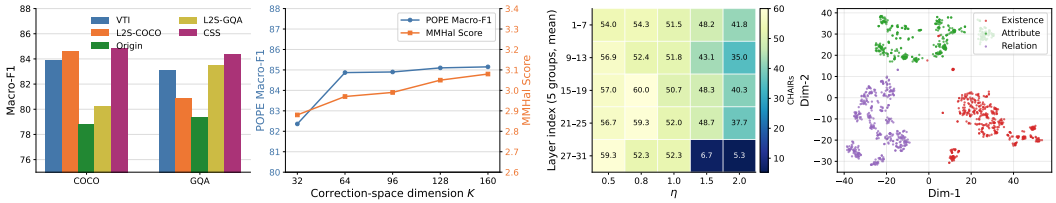


Figure 4: Further analysis of CSS on LLaVA-1.5. (a) Cross-dataset generalization. (b) Sensitivity to correction-space dimension K. (c) Layer-strength sensitivity. (d) Visualization of offline correction coordinates.

form of adaptive steering. To control for differences in training data, we additionally train L2S using the same AMBER training data as CSS. Appendix C compares the two methods when both are trained on the same AMBER data.

Analysis of Correction-Space Dimension. Figure 4(b) analyzes the effect of the correction-space dimension K. When K is small, the correction space has limited capacity and cannot fully capture hallucination-related activation shifts. As K increases, both POPE Macro-F1 and MMHal Score improve and then gradually saturate, indicating that a compact correction space is sufficient for effective intervention. This trend is consistent with our effective-rank analysis, which shows that activation differences can be approximated by a low-dimensional structure.

Analysis of Layer and Steering Strength. Figure 4(c) studies the joint effect of the steering layer and steering strength η. CSS achieves better mitigation results in middle-to-late layers with moderate steering strengths, while overly weak or overly strong interventions are less effective. This suggests that hallucination-related representations are more suitable for correction after they become sufficiently formed in intermediate layers, but excessive steering may disrupt the model's generation dynamics.

Analysis of Coordinate Structure. Finally, Figure 4(d) visualizes the offline correction coordinates $z = E_\phi(\Delta h)$. The coordinates form distinguishable regions corresponding to different hallucination types, indicating that the learned correction space preserves type-aware structure. This supports the role of type-supervised correction-space learning and further explains why CSS can perform sample-specific steering through low-dimensional coordinates rather than unconstrained full-dimensional vectors.

6 Conclusion

In this paper, we proposed Correction Space Steering (CSS), an inference-time activation steering framework for mitigating hallucinations in LVLMs. We first identified structured properties of hallucination-related activation differences and theoretically showed that their low effective rank implies a compact low-dimensional approximation, providing the foundation for coordinate-based correction. Building on this result, CSS learns a correction space $\mathcal{S}$ and performs adaptive steering by predicting low-dimensional correction coordinates rather than unconstrained full-dimensional vectors. Experiments on generative, discriminative, and open-ended hallucination benchmarks show that CSS consistently reduces hallucinations without updating LVLM parameters, demonstrating the effectiveness of theory-guided correction space steering.

Acknowledgments and Disclosure of Funding

This work was supported by the National Natural Science Foundation of China (Grant No.62506318); Guangdong Provincial Department of Education Project (Grant No.2024KQNCX028); Scientific Research Projects for the Higher-educational Institutions (Grant No.2024312096), Education Bureau of Guangzhou Municipality; Guangzhou-HKUST(GZ) Joint Funding Program (Grant No.2025A03J3957), Education Bureau of Guangzhou Municipality.

References

[1] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, volume 36, pages 34892–34916, 2023.

[2] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N. Fung, and Steven Hoi. InstructBLIP: Towards general-purpose vision-language models with instruction tuning. *Advances in Neural Information Processing Systems*, 36: 49250–49267, December 2023. URL https://papers.nips.cc/paper_files/paper/2023/hash/9a6a435e75419a836fe47ab6793623e6-Abstract-Conference.html.

[3] Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. A survey on multimodal large language models. *National Science Review*, 11(12):nwae403, November 2024. doi: 10.1093/nsr/nwae403.

[4] HuangLei, YuWeijiang, MaWeitao, ZhongWeihong, FengZhangyin, WangHaotian, ChenQianglong, PengWeihua, FengXiaocheng, QinBing, and LiuTing. A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions. *ACM Transactions on Information Systems*, January 2025. doi: 10.1145/3703155.

[5] Sicong Leng, Hang Zhang, Guanzheng Chen, Xin Li, Shijian Lu, Chunyan Miao, and Lidong Bing. 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, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Leng_Mitigating_Object_Hallucinations_in_Large_Vision-Language_Models_through_Visual_Contrastive_CVPR_2024_paper.html.

[6] Can Cui, Yunsheng Ma, Xu Cao, Wenqian Ye, Yang Zhou, Kaizhao Liang, Jintai Chen, Juanwu Lu, Zichong Yang, Kuei-Da Liao, Tianren Gao, Erlong Li, Kun Tang, Zhipeng Cao, Tong Zhou, Ao Liu, Xinrui Yan, Shuqi Mei, Jianguo Cao, Ziran Wang, and Chao Zheng. A Survey on Multimodal Large Language Models for Autonomous Driving. In *2024 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW)*, pages 958–979, Waikoloa, HI, USA, January 2024. IEEE. doi: 10.1109/WACVW60836.2024.00106.

[7] Xiaoqi Li, Mingxu Zhang, Yiran Geng, Haoran Geng, Yuxing Long, Yan Shen, Renrui Zhang, Jiaming Liu, and Hao Dong. ManipLLM: Embodied Multimodal Large Language Model for Object-Centric Robotic Manipulation, December 2023.

[8] Zhuhao Wang, Yihua Sun, Zihan Li, Xuan Yang, Fang Chen, and Hongen Liao. LLM-RG4: Flexible and factual radiology report generation across diverse input contexts. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(8):8250–8258, April 2025. doi: 10.1609/aaai.v39i8.32890.

[9] Xinpeng Ding, Kui Zhang, Jianhua Han, Lanqing Hong, Hang Xu, and Xiaomeng Li. PaMi-VDPO: Mitigating Video Hallucinations by Prompt-Aware Multi-Instance Video Preference Learning, April 2025.

[10] Chaoya Jiang, Haiyang Xu, Mengfan Dong, Jiaxing Chen, Wei Ye, Ming Yan, Qinghao Ye, Ji Zhang, Fei Huang, and Shikun Zhang. Hallucination augmented contrastive learning for multimodal large language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 27036–27046, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Jiang_Hallucination_Augmented_Contrastive_Learning_for_Multimodal_Large_Language_Model_CVPR_2024_paper.html.

[11] Yun Xing, Yiheng Li, Ivan Laptev, and Shijian Lu. Mitigating object hallucination via concentric causal attention. *Advances in Neural Information Processing Systems*, 37:92012–92035, December 2024. doi: 10.52202/079017-2920.

[12] Xintong Wang, Jingheng Pan, Liang Ding, and Chris Biemann. Mitigating hallucinations in large vision-language models with instruction contrastive decoding. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 15840–15853, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.937. Findings 2024.

[13] Zhaorun Chen, Zhuokai Zhao, Hongyin Luo, Huaxiu Yao, Bo Li, and Jiawei Zhou. HALC: Object hallucination reduction via adaptive focal-contrast decoding. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *ICML'24*, pages 7824–7846, Vienna, Austria, July 2024. JMLR.org.

[14] Sheng Liu, Haotian Ye, and James Zou. Reducing hallucinations in large vision-language models via latent space steering. In *The Thirteenth International Conference on Learning Representations*, October 2024. URL https://openreview.net/forum?id=LBl7Hez0fF.

[15] Jayneel Parekh, Pegah Khayatan, Mustafa Shukor, Arnaud Dapogny, Alasdair Newson, and Matthieu Cord. Learning to steer: Input-dependent steering for multimodal LLMs, November 2025.

[16] Jianghao Yin, Qin Chen, Kedi Chen, Jie Zhou, Xingjiao Wu, and Liang He. Dynamic Multimodal Activation Steering for Hallucination Mitigation in Large Vision-Language Models, February 2026.

[17] Junyang Wang, Yuhang Wang, Guohai Xu, Jing Zhang, Yukai Gu, Haitao Jia, Jiaqi Wang, Haiyang Xu, Ming Yan, Ji Zhang, and Jitao Sang. AMBER: An LLM-free Multi-dimensional Benchmark for MLLMs Hallucination Evaluation, February 2024.

[18] Bryan A. Plummer, Liwei Wang, Chris M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k Entities: Collecting Region-to-Phrase Correspondences for Richer Image-to-Sentence Models. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2641–2649, December 2015. doi: 10.1109/ICCV.2015.303.

[19] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollar, and C. Lawrence Zitnick. Microsoft COCO captions: Data collection and evaluation server, April 2015.

[20] Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. A-OKVQA: A benchmark for visual question answering using world knowledge. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 146–162, Cham, 2022. Springer Nature Switzerland. doi: 10.1007/978-3-031-20074-8_9.

[21] Keqin Chen, Zhao Zhang, Weili Zeng, Richong Zhang, Feng Zhu, and Rui Zhao. Shikra: Unleashing Multimodal LLM's Referential Dialogue Magic, July 2023.

[22] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL technical report, February 2025.

[23] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the 40th International Conference on Machine Learning*, pages 19730–19742. PMLR, July 2023. URL https://proceedings.mlr.press/v202/li23q.html. International Conference on Machine Learning.

[24] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. MiniGPT-4: Enhancing Vision-Language Understanding with Advanced Large Language Models. *International Conference on Learning Representations*, 2024:18378–18394, May 2024.

[25] Fuxiao Liu, Kevin Lin, Linjie Li, Jianfeng Wang, Yaser Yacoob, and Lijuan Wang. Mitigating Hallucination in Large Multi-Modal Models via Robust Instruction Tuning. *International Conference on Learning Representations*, 2024:57689–57733, May 2024.

[26] Tianyu Yu, Yuan Yao, Haoye Zhang, Taiwen He, Yifeng Han, Ganqu Cui, Jinyi Hu, Zhiyuan Liu, Hai-Tao Zheng, and Maosong Sun. RLHF-V: Towards trustworthy MLLMs via behavior alignment from fine-grained correctional human feedback. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13807–13816, June 2024. doi: 10.1109/CVPR52733.2024.01310.

[27] Yassine Ouali, Adrian Bulat, Brais Martinez, and Georgios Tzimiropoulos. CLIP-DPO: Vision-language models as a source of preference for fixing hallucinations in LVLMs. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol, editors, *Computer Vision – ECCV 2024*, pages 395–413, Cham, 2025. Springer Nature Switzerland. doi: 10.1007/978-3-031-73116-7_23.

[28] Xinyu Lyu, Beitao Chen, Lianli Gao, Jingkuan Song, and Heng T. Shen. Alleviating hallucinations in large vision-language models through hallucination-induced optimization. *Advances in Neural Information Processing Systems*, 37:122811–122832, December 2024. doi: 10.52202/079017-3902.

[29] Sihang Jia, Shuliang Liu, Songbo Yang, Yibo Yan, Xin Zou, and Xuming Hu. Decoding by perturbation: Mitigating mllm hallucinations via dynamic textual perturbation, 2026. URL https://arxiv.org/abs/2604.12424.

[30] Seongyun Lee, Sue Hyun Park, Yongrae Jo, and Minjoon Seo. Volcano: Mitigating multimodal hallucination through self-feedback guided revision, April 2024.

[31] Shuliang Liu, Songbo Yang, Dong Fang, Sihang Jia, Yuqi Tang, Lingfeng Su, Ruoshui Peng, Yibo Yan, Xin Zou, and Xuming Hu. Vision-language introspection: Mitigating overconfident hallucinations in MLLMs via interpretable bi-causal steering. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens, editors, *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 38518–38543, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1784. URL https://aclanthology.org/2026.acl-long.1784/.

[32] Le Yang, Ziwei Zheng, Boxu Chen, Zhengyu Zhao, Chenhao Lin, and Chao Shen. Nullu: Mitigating object hallucinations in large vision-language models via HalluSpace projection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14635–14645, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Yang_Nullu_Mitigating_Object_Hallucinations_in_Large_Vision-Language_Models_via_HalluSpace_CVPR_2025_paper.html.

[33] Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. Object hallucination in image captioning, March 2019.

[34] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 292–305, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.20. EMNLP 2023.

[35] Zhiqing Sun, Sheng Shen, Shengcao Cao, Haotian Liu, Chunyuan Li, Yikang Shen, Chuang Gan, Liangyan Gui, Yu-Xiong Wang, Yiming Yang, Kurt Keutzer, and Trevor Darrell. Aligning large multimodal models with factually augmented RLHF. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 13088–13110, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.775. Findings 2024.

[36] Hao Yin, Guangzong Si, and Zilei Wang. ClearSight: Visual signal enhancement for object hallucination mitigation in multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14625–14634, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Yin_ClearSight_Visual_Signal_Enhancement_for_Object_Hallucination_Mitigation_in_Multimodal_CVPR_2025_paper.html.

[37] Jiazhen Liu, Yuhan Fu, Ruobing Xie, Runquan Xie, Xingwu Sun, Fengzong Lian, Zhanhui Kang, and Xirong Li. Phd: A chatgpt-prompted visual hallucination evaluation dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19857–19866, June 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Liu_PhD_A_ChatGPT-Prompted_Visual_Hallucination_Evaluation_Dataset_CVPR_2025_paper.html.

[38] Shuliang Liu, Zheng Qi, Jesse Jiaxi Xu, Yibo Yan, Junyan Zhang, He Geng, Aiwei Liu, Peijie Jiang, Jia Liu, Yik-Cheung Tam, and Xuming Hu. VLA-mark: A cross modal watermark for large vision-language alignment models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 26409–26427. Association for Computational Linguistics, November 2025. doi: 10.18653/v1/2025.emnlp-main.1342. URL https://aclanthology.org/2025.emnlp-main.1342/.

[39] Qi Zheng, Shuliang Liu, Yu Huang, Sihang Jia, Jungang Li, Lyuhao Chen, Junhao Chen, Hanqian Li, Aiwei Liu, Yibo Yan, and Xuming Hu. A visual semantic adaptive watermark grounded by prefix-tuning for large vision-language model, 2026. URL https://arxiv.org/abs/2601.07291.

[40] Shuliang Liu, Xingyu Li, Hongyi Liu, Dong Fang, Bingchen Duan, Zheng Qi, Lingfeng Su, and Xuming Hu. Distilling the thought, watermarking the answer: A principle semantic guided watermark for reasoning large language models. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust, editors, *International Conference on Learning Representations*, volume 2026, pages 140799–140829, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/e45b0e8c296fa782d997f8a660bac512-Paper-Conference.pdf.

Appendix Contents

A Theoretical Details **16**

A.1 A Frobenius-Norm Optimality Lemma 16

A.2 Proof of Proposition 3.1 17

A.3 From Low-Dimensional Approximation to Correction-Space Steering 18

B Experimental Setup **18**

B.1 Datasets and Metrics 18

B.2 Implementation and Hyperparameters 19

B.3 Compute Resources 19

C Supplementary Experimental Results **20**

C.1 Extended Benchmark Evaluation 20

C.2 Comparison with L2S Trained on the Same AMBER Data 20

C.3 Activation-Difference Analysis and Controls 21

C.4 Extended Component Ablations 21

C.5 Sensitivity to Type Supervision 21

C.6 Evaluation of the Online Coordinate Predictor 22

C.7 Hallucination Mitigation Across Generation Stages 23

D Limitations and Broader Impacts **23**

D.1 Limitations 23

D.2 Broader Impacts 24

A Theoretical Details

In this section, we provide additional theoretical details for the low-dimensional approximation result introduced in Section 3. We first state and prove a standard optimality property of truncated SVD under the Frobenius norm, and then use it to prove Proposition 3.1. Finally, we discuss how this result motivates coordinate-based correction and the learned correction space used by CSS.

A.1 A Frobenius-Norm Optimality Lemma

Lemma A.1 (Optimality of Truncated SVD under Frobenius Norm)**.** *Let $A \in \mathbb{R}^{N\times d}$ have singular value decomposition*

$$A = U\Sigma V^\top, \tag{17}$$

where the non-zero singular values satisfy $\sigma_1 \geq \cdots \geq \sigma_r > 0$. Let

$$A_K = U_K\Sigma_K V_K^\top \tag{18}$$

be the rank-K truncated SVD approximation of A. Then, for any matrix M with $\mathrm{rank}(M) \leq K$,

$$\|A - A_K\|_F^2 \leq \|A - M\|_F^2. \tag{19}$$

Moreover,

$$\|A - A_K\|_F^2 = \sum_{j=K+1}^{r} \sigma_j^2. \tag{20}$$

Proof. Since the Frobenius norm is invariant under orthogonal transformations, for any rank-K matrix M, we have

$$\|A - M\|_F^2 = \|U^\top(A - M)V\|_F^2 = \|\Sigma - U^\top MV\|_F^2. \tag{21}$$

Let $G = U^\top MV$. Since multiplication by orthogonal matrices preserves rank, $\mathrm{rank}(G) \leq K$. Therefore, it is sufficient to approximate the diagonal matrix Σ by a rank-K matrix G.

Let $\mathcal{S}$ be the column space of G, and let $P_{\mathcal{S}}$ be the orthogonal projection onto $\mathcal{S}$. Since $\dim(\mathcal{S}) \leq K$, for each column of Σ, the projection onto $\mathcal{S}$ gives the best approximation among all vectors in $\mathcal{S}$. Hence,

$$\|\Sigma - G\|_F^2 \geq \|\Sigma - P_{\mathcal{S}}\Sigma\|_F^2. \tag{22}$$

Because Σ is diagonal, we have

$$\|\Sigma - P_{\mathcal{S}}\Sigma\|_F^2 = \sum_{j=1}^{r} \sigma_j^2 \|(I - P_{\mathcal{S}})e_j\|_2^2, \tag{23}$$

where e_j denotes the j-th standard basis vector. Since $P_{\mathcal{S}}$ is an orthogonal projection with rank at most K,

$$\sum_{j=1}^{r} \|P_{\mathcal{S}}e_j\|_2^2 = \mathrm{tr}(P_{\mathcal{S}}) \leq K. \tag{24}$$

Equivalently, the total amount of discarded projection weight is at least $r - K$. Since the singular values are sorted in descending order, the smallest possible weighted residual is obtained by assigning the discarded weight to the directions with the smallest singular values. Therefore,

$$\|\Sigma - P_{\mathcal{S}}\Sigma\|_F^2 \geq \sum_{j=K+1}^{r} \sigma_j^2. \tag{25}$$

Thus, for any rank-K matrix M,

$$\|A - M\|_F^2 \geq \sum_{j=K+1}^{r} \sigma_j^2. \tag{26}$$

On the other hand, the truncated SVD approximation satisfies

$$A - A_K = \sum_{j=K+1}^{r} \sigma_j u_j v_j^\top. \tag{27}$$

The matrices $\{u_j v_j^\top\}_{j=1}^r$ are orthonormal under the Frobenius inner product. Therefore,

$$\|A - A_K\|_F^2 = \sum_{j=K+1}^{r} \sigma_j^2. \tag{28}$$

Hence, A_K attains the minimum Frobenius reconstruction error among all rank-K matrices. This completes the proof. □

A.2 Proof of Proposition 3.1

Let the singular value decomposition of the activation-difference matrix be

$$\Delta H^{(l)} = U \Sigma V^\top, \tag{29}$$

where $\Sigma = \mathrm{diag}(\sigma_1^{(l)}, \ldots, \sigma_r^{(l)})$ contains the non-zero singular values with $\sigma_1^{(l)} \geq \cdots \geq \sigma_r^{(l)} > 0$. Let $V_K = [v_1, \ldots, v_K] \in \mathbb{R}^{d \times K}$ denote the top K right singular vectors. In the notation of Proposition 3.1, we have

$$B_{l,K}^{\mathrm{svd}} = V_K. \tag{30}$$

The rank-K truncated SVD approximation of $\Delta H^{(l)}$ is

$$\Delta H_K^{(l)} = U_K \Sigma_K V_K^\top. \tag{31}$$

Since $V_K V_K^\top$ is the orthogonal projection onto the subspace spanned by the top K right singular vectors, we also have

$$\Delta H_K^{(l)} = \Delta H^{(l)} V_K V_K^\top = \Delta H^{(l)} B_{l,K}^{\mathrm{svd}} B_{l,K}^{\mathrm{svd}\top}. \tag{32}$$

By Lemma A.1, $\Delta H_K^{(l)}$ is the optimal rank-K approximation of $\Delta H^{(l)}$ under the Frobenius norm, and its squared reconstruction error is

$$\left\| \Delta H^{(l)} - \Delta H_K^{(l)} \right\|_F^2 = \sum_{j=K+1}^{r} (\sigma_j^{(l)})^2. \tag{33}$$

The squared Frobenius norm of the original matrix is

$$\left\| \Delta H^{(l)} \right\|_F^2 = \sum_{j=1}^{r} (\sigma_j^{(l)})^2. \tag{34}$$

By the definition of $K = \mathrm{erank}_\tau(\Delta H^{(l)})$, we have

$$\frac{\sum_{j=1}^{K} (\sigma_j^{(l)})^2}{\sum_{j=1}^{r} (\sigma_j^{(l)})^2} \geq \tau. \tag{35}$$

Therefore,

$$\frac{\left\| \Delta H^{(l)} - \Delta H_K^{(l)} \right\|_F^2}{\left\| \Delta H^{(l)} \right\|_F^2} = \frac{\sum_{j=K+1}^{r} (\sigma_j^{(l)})^2}{\sum_{j=1}^{r} (\sigma_j^{(l)})^2} = 1 - \frac{\sum_{j=1}^{K} (\sigma_j^{(l)})^2}{\sum_{j=1}^{r} (\sigma_j^{(l)})^2} \leq 1 - \tau. \tag{36}$$

This proves the relative reconstruction error bound.

It remains to show the sample-level decomposition. Since $B_{l,K}^{\mathrm{svd}} B_{l,K}^{\mathrm{svd}\top}$ is an orthogonal projection onto the SVD subspace, each activation difference $\Delta h_i^{(l)}$ can be decomposed into its projection onto this subspace and the residual outside it:

$$\Delta h_i^{(l)} = B_{l,K}^{\mathrm{svd}} B_{l,K}^{\mathrm{svd}\top} \Delta h_i^{(l)} + \left(I - B_{l,K}^{\mathrm{svd}} B_{l,K}^{\mathrm{svd}\top} \right) \Delta h_i^{(l)}. \tag{37}$$

Let

$$\alpha_i^{(l)} = B_{l,K}^{\mathrm{svd}\top} \Delta h_i^{(l)} \tag{38}$$

and

$$\epsilon_i^{(l)} = \left(I - B_{l,K}^{\mathrm{svd}} B_{l,K}^{\mathrm{svd}\top} \right) \Delta h_i^{(l)}. \tag{39}$$

Then we obtain

$$\Delta h_i^{(l)} = B_{l,K}^{\mathrm{svd}} \alpha_i^{(l)} + \epsilon_i^{(l)}. \tag{40}$$

This completes the proof of Proposition 3.1.

A.3 From Low-Dimensional Approximation to Correction-Space Steering

Proposition 3.1 shows that when the activation-difference matrix has low effective rank, its rows can be approximated by coordinates in a low-dimensional subspace with bounded reconstruction error. This provides a theoretical basis for replacing unconstrained full-dimensional correction-vector prediction with low-dimensional coordinate prediction.

In particular, the proposition shows that there exists a basis $B_{l,K}^{\mathrm{svd}} \in \mathbb{R}^{d\times K}$ such that each activation difference can be approximated as

$$\Delta h_i^{(l)} \approx B_{l,K}^{\mathrm{svd}} \alpha_i^{(l)}. \tag{41}$$

Therefore, a correction vector does not need to be predicted freely in the full hidden space $\mathbb{R}^d$. Instead, it can be represented through a compact coordinate vector in $\mathbb{R}^K$ and reconstructed through a basis matrix. This motivates the CSS formulation, where the inference-time correction vector is written as

$$\delta_t = B\hat{z}_t, \tag{42}$$

with $B \in \mathbb{R}^{d\times K}$ denoting the learned correction-space basis and $\hat{z}_t \in \mathbb{R}^K$ denoting the predicted correction coordinate.

However, the SVD basis $B_{l,K}^{\mathrm{svd}}$ is only optimal for minimizing the Frobenius reconstruction error of activation differences. It does not explicitly preserve hallucination-type structure or optimize for inference-time steering. As shown in the empirical analysis in Section 1, activation differences are not merely low-rank; they are also predictive of hallucination types and organized into type-aware geometric structures. Therefore, CSS learns a correction space $\mathcal{S} = \mathrm{span}(B)$ instead of directly using the SVD subspace.

The learned correction space is optimized to reconstruct activation differences, reduce redundancy among basis directions, and preserve hallucination-type information through the type-supervised objective. This distinction also explains the role of the SVD-space ablation in our experiments. The SVD subspace provides a reconstruction-oriented low-dimensional baseline, while the learned correction space provides a steering-oriented representation. The performance gap between the SVD-space variant and the full CSS model shows that low-dimensionality is useful but not sufficient; preserving type-aware correction structure is also important for effective hallucination mitigation.

B Experimental Setup

B.1 Datasets and Metrics

We use AMBER to construct activation-difference training signals for correction-space learning, and evaluate CSS on three standard hallucination benchmarks: CHAIR, POPE, and MMHal-Bench. These benchmarks cover generative object hallucination, discriminative object hallucination detection, and open-ended multimodal hallucination evaluation.

AMBER. We use AMBER [17] to construct activation-difference training signals for correction-space learning. AMBER is an LLM-free multi-dimensional hallucination benchmark with annotations for existence, attribute, and relation hallucinations, which align with the type-aware structure analyzed in our paper. Based on these annotations, we construct paired non-hallucinatory and hallucinatory trajectories for the same image-question input: only the first Yes/No response token is constrained, and subsequent tokens are generated autoregressively. Pairs with a response-length difference greater than two tokens are filtered out before extracting activation differences at the steering layer. We use 2,250 AMBER samples, with 750 samples for each hallucination type.

CHAIR. CHAIR [33] evaluates object hallucination in free-form image captioning by comparing objects mentioned in generated responses with ground-truth objects in the image. Following prior work, we generate captions for images from the MSCOCO validation set and report CHAIR_S, CHAIR_I, and Recall. CHAIR_S measures the fraction of generated sentences that contain hallucinated objects, while CHAIR_I measures the fraction of hallucinated object instances among all mentioned objects:

$$\mathrm{CHAIR}_S = \frac{|\{\text{sentences with hallucinated objects}\}|}{|\{\text{all sentences}\}|}, \qquad \mathrm{CHAIR}_I = \frac{|\{\text{hallucinated objects}\}|}{|\{\text{all mentioned objects}\}|}. \tag{43}$$

Lower CHAIR_S and CHAIR_I indicate fewer object hallucinations, while higher Recall indicates better coverage of ground-truth objects.

POPE. POPE [34] evaluates object hallucination as a binary Yes/No question-answering task. Each question asks whether a specific object exists in the given image. We evaluate CSS on both MSCOCO and GQA. Following prior work, POPE contains three settings: Random, Popular, and Adversarial. In the Random setting, queried objects are randomly sampled; in the Popular setting, queried objects are selected from frequent objects; and in the Adversarial setting, queried objects are selected from objects that frequently co-occur with the ground-truth objects. We report Accuracy, Precision, Recall, and F1 score, and the main paper reports results averaged over the three settings.

MMHal-Bench. MMHal-Bench [35] evaluates hallucination in open-ended VQA scenarios beyond object existence. It contains questions covering diverse hallucination categories, including Attribute, Adversarial, Comparison, Counting, Relation, Environment, Holistic, and Other. We use GPT-4o as the automatic evaluator. The evaluator assigns a score from 0 to 6, where higher scores indicate more informative and less hallucinated responses. A response with a score lower than 3 is considered hallucinated. We report both the hallucination rate and the average response score.

B.2 Implementation and Hyperparameters

CSS keeps the LVLM backbone frozen throughout all experiments and only learns lightweight correction modules. To learn the correction space, we use the AMBER samples described in Section B.1. Unless otherwise specified, the correction-space dimension is set to $K = 64$. Both the offline coordinate encoder E_ϕ and the online coordinate predictor P_ψ are implemented as two-layer MLPs. The basis matrix $B \in \mathbb{R}^{d\times K}$ is learned jointly with the offline coordinate encoder during correction-space learning. For the correction-space learning objective in Equation 13, we set $\lambda_o = 0.1$ and $\lambda_{\text{cls}} = 0.1$. We apply CSS at a single steering layer l^* for each backbone. Specifically, we set $l^* = 13$ for LLaVA-v1.5, $l^* = 15$ for InstructBLIP, and $l^* = 13$ for Qwen2.5-VL. During inference-time steering, the steering strength in Equation 16 is set to $\eta = 2$ unless otherwise specified. All experiments are implemented in PyTorch. For all compared baselines, we follow the experimental settings reported in their original papers or released implementations whenever available. CSS and the original model are evaluated under the same decoding and evaluation protocol to ensure fair comparison.

Optimization, validation, and decoding. We use AdamW with learning rate 10^{-3}, weight decay 10^{-5}, batch size 64, and a cosine learning-rate schedule to learn the correction space for 200 epochs (including 50 first-stage epochs). The online coordinate predictor is a two-layer MLP with hidden size 256 and dropout 0.1. It is trained using AdamW (learning rate 10^{-3}, weight decay 10^{-5}, batch size 64) for 50 epochs. We reserve 15% of the training data for validation, use random seed 42, and decode greedily with beam size 1 and `max_new_tokens=128`. The intervention layer is selected on held-out validation data from odd-numbered layers between 1 and 31, with steering strengths searched over $\{0.5, 0.8, 1.0, 2.0, 3.0\}$. The chosen settings are held fixed for evaluation. The correction-space dimension and loss weights are predefined.

B.3 Compute Resources

All experiments are conducted on NVIDIA RTX 4090 GPUs with 24GB memory. CSS keeps the LVLM backbone frozen and only trains lightweight correction modules, including the correction-space basis, the offline coordinate encoder, and the online coordinate predictor. Therefore, the additional training cost is substantially lower than full LVLM fine-tuning. The main computational cost comes from extracting hidden states from paired response trajectories and running inference-time evaluation on hallucination benchmarks.

C Supplementary Experimental Results

C.1 Extended Benchmark Evaluation

We provide comprehensive POPE comparisons across the evaluated LVLM backbones. Table 5 reports the expanded POPE results on MSCOCO and GQA for LLaVA-1.5, InstructBLIP, and Qwen2.5-VL, including additional inference-time baselines and all four evaluation metrics.

Table 5: Expanded POPE comparisons. All numbers are percentages, averaged across Random, Popular, and Adversarial settings.

Backbone	Method	POPE (MSCOCO)				POPE (GQA)			
		Acc. (%)↑	Prec. (%)↑	Rec. (%)↑	F1 (%)↑	Acc. (%)↑	Prec. (%)↑	Rec. (%)↑	F1 (%)↑
LLaVA-1.5	Origin	80.67	87.47	71.78	78.78	79.33	80.69	78.44	79.36
	VCD	83.56	89.03	76.89	82.43	82.33	82.64	83.11	82.70
	ICD	83.50	81.88	86.55	84.08	76.90	70.68	93.93	80.49
	ClearSight	84.22	83.28	84.83	83.96	80.11	74.86	90.80	81.82
	VTI	83.78	84.83	83.33	83.89	80.89	76.72	91.33	83.06
	L2S	83.90	84.43	83.21	83.86	79.67	70.88	95.33	80.88
	DMAS	83.42	83.95	82.80	83.37	78.95	70.20	94.10	80.40
	CSS (Ours)	84.51	85.23	84.52	84.87	84.14	76.96	93.28	84.33
InstructBLIP	Origin	77.78	77.97	77.78	77.84	76.67	73.39	84.89	78.54
	VCD	78.44	78.45	78.67	78.53	78.44	74.68	86.67	80.18
	ICD	79.18	77.66	82.11	79.77	74.79	71.05	84.83	77.23
	ClearSight	78.90	78.50	79.35	78.92	77.85	74.10	85.80	79.52
	VTI	78.76	78.81	78.69	78.75	79.32	75.41	87.25	80.89
	L2S	78.94	79.02	78.97	78.91	77.18	73.92	85.34	79.22
	DMAS	78.55	78.70	78.40	78.55	76.85	73.10	84.90	78.55
	CSS (Ours)	79.35	79.28	79.21	79.24	80.21	76.14	86.93	81.17
Qwen2.5-VL	Origin	87.90	97.00	78.20	86.60	85.70	89.20	82.00	85.30
	VCD	89.00	97.10	80.40	88.00	85.80	87.80	83.80	85.70
	ICD	87.00	96.65	76.67	85.50	87.56	90.19	84.67	87.26
	ClearSight	87.60	96.50	77.90	86.21	85.40	88.90	81.60	85.09
	VTI	88.40	96.50	79.80	87.40	86.20	88.50	83.50	85.90
	L2S	86.00	98.23	73.33	83.97	85.67	90.57	80.00	84.88
	DMAS	85.50	96.80	72.80	83.10	85.20	89.80	79.50	84.30
	CSS (Ours)	89.50	97.90	80.80	88.53	88.20	91.20	85.20	88.10

Evaluation on Qwen3-VL-8B. We additionally evaluate CSS on Qwen3-VL-8B to examine transfer to a more recent backbone. Table 6 shows that CSS reduces both CHAIR hallucination metrics and improves POPE Accuracy and F1 over Origin and L2S, extending the observed benefits beyond the three primary backbones.

Table 6: Hallucination mitigation on Qwen3-VL-8B, evaluated on CHAIR and POPE.

Method	CHAIR		POPE	
	$\text{CHAIR}_S \downarrow$	$\text{CHAIR}_I \downarrow$	Acc↑	F1↑
Origin	27.3	9.2	88.6	88.4
L2S	25.6	7.4	89.1	88.8
CSS	24.3	6.5	89.4	89.3

C.2 Comparison with L2S Trained on the Same AMBER Data

The cross-dataset comparison in the main paper uses L2S predictors trained on COCO or GQA, while CSS learns its correction modules from AMBER. Consequently, the observed performance differences may reflect both the intervention method and the training data. To assess the contribution of the method under a shared data source, we retrain L2S on the same AMBER training data used by CSS and evaluate both methods with LLaVA-1.5.

Table 7 shows that CSS achieves lower CHAIR-S and CHAIR-I and higher POPE Accuracy and F1 than L2S under this setting. The advantage persists when the training data are shared, supporting a contribution from CSS's correction-space formulation beyond the choice of AMBER as its training source.

Table 7: Comparison on LLaVA-1.5 with L2S and CSS trained on the same AMBER data.

Method	CHAIR		POPE	
	$\text{CHAIR}_S \downarrow$	$\text{CHAIR}_I \downarrow$	Acc. (%)↑	F1 (%)↑
L2S	45.3	13.7	83.51	82.81
CSS (Ours)	**38.4**	**11.5**	**84.51**	**84.87**

C.3 Activation-Difference Analysis and Controls

We analyze activation differences on AMBER discriminative questions, whose existence, attribute, and relation labels allow us to assess hallucination-type predictability. We follow the paired-trajectory construction described in Section 4 and Appendix B.2, and compare both type-prediction performance and spectral concentration.

Trajectory Construction Controls. We evaluate fixed-template responses (*The answer is Yes.* and *The answer is No.*), random pairing of truthful and hallucinatory responses from the response pool, and question paraphrases that preserve the paired-response construction. Table 8 shows that random pairing substantially weakens type predictability, whereas fixed templates and question paraphrases retain predictive structure.

Table 8: Controls for the construction of truthful–hallucinatory activation-difference pairs.

Control	Type Macro-F1	Top-10 Energy	Rank@70	Rank@80	Rank@90
Fixed-template	0.80	0.990	1	1	1
Random pairing	0.21	0.570	22	45	115
Question paraphrases	0.78	0.647	12	36	89

C.4 Extended Component Ablations

Table 9 extends the component ablations in Table 3 with random-basis and linear-predictor controls on both backbones. The random-basis variant performs worse than the learned basis, supporting the importance of correction-space construction. Replacing the online predictor with a linear mapping also degrades hallucination mitigation, supporting the nonlinear predictor. Together with the SVD-basis, type-loss, and basis-projection variants, these results support the complementary contributions of basis learning, type supervision, and adaptive coordinate prediction.

C.5 Sensitivity to Type Supervision

Type supervision encourages correction coordinates to retain hallucination-type information. We examine how this benefit depends on the availability and reliability of the annotations through two experiments on LLaVA-1.5. The label-availability experiment varies the fraction of available type labels, while the label-noise experiment varies the fraction of corrupted labels. Table 10 reports their effects on downstream CHAIR performance.

With fewer available labels, both hallucination metrics deteriorate gradually, while Recall remains relatively stable. Thus, useful correction-space learning is possible with partial type supervision, although full supervision gives the lowest hallucination rates. Corrupting labels produces a larger and progressively worsening deterioration in the tested settings. This pattern is consistent with incorrect type targets providing conflicting guidance during space learning. Together, the experiments suggest that the method can tolerate incomplete annotation more readily than unreliable annotation, and that the benefit of type supervision depends on label quality.

Table 9: Controlled comparisons of correction-space construction and adaptive prediction. SVD basis retains the predictor and inference configuration while substituting a truncated-SVD basis; basis projection uses the learned basis without online coordinate prediction.

Backbone	Variant	CHAIR-S↓	CHAIR-I↓	Recall↑
LLaVA-1.5	Origin	60.7	17.6	75.8
	SVD basis	44.8	13.4	73.8
	Random basis	57.3	14.4	74.9
	Linear predictor	42.5	13.1	72.6
	w/o $\mathcal{L}_{\text{cls}}$	40.3	12.4	72.5
	Basis projection	45.6	13.8	74.2
	CSS (Ours)	38.4	11.5	73.6
InstructBLIP	Origin	51.3	13.8	69.8
	SVD basis	44.7	13.1	72.5
	Random basis	51.8	13.5	70.1
	Linear predictor	41.6	12.4	70.2
	w/o $\mathcal{L}_{\text{cls}}$	42.8	12.6	69.8
	Basis projection	44.3	12.8	70.7
	CSS (Ours)	40.4	11.8	73.1

Table 10: Sensitivity to type-label availability and noise on LLaVA-1.5. The proportion denotes available labels in the first group and corrupted labels in the second. Best results within each group are in **bold**.

Type-label setting	Proportion (%)	CHAIR		Recall↑
		$\text{CHAIR}_S \downarrow$	$\text{CHAIR}_I \downarrow$	
Available labels	20	40.8	12.1	72.8
	50	40.1	12.0	**74.0**
	80	39.0	11.7	73.5
	100	**38.4**	**11.5**	73.6
Label noise	0	**38.4**	**11.5**	**73.6**
	20	41.2	12.3	73.2
	30	45.6	13.5	72.6
	40	49.8	14.6	72.1
	50	53.5	15.4	71.8

C.6 Evaluation of the Online Coordinate Predictor

This experiment tests whether the online predictor can estimate correction coordinates that mitigate hallucinations without access to paired truthful and hallucinatory trajectories at inference time. On held-out AMBER samples, we assess the accuracy of the predicted coordinates and whether the resulting interventions reduce hallucinations.

Coordinate Prediction Accuracy. For each sample, the target coordinate is obtained by encoding the activation difference between its paired truthful and hallucinatory responses. We compare the predictor's output with this target and with a baseline that outputs the same mean coordinate for every sample. The predictor achieves MSE 43.7466 and NRMSE 0.7554, compared with 76.7596 and 1.0006 for the mean-coordinate baseline. The lower prediction errors indicate that the predictor captures variation in correction coordinates across inputs.

Effect of Predicted Corrections. We then compare four intervention settings on LLaVA-1.5 (Table 11). **No intervention** gives the original model's performance. **Paired difference** applies the activation difference extracted from the paired responses directly. **Reconstructed difference** encodes that difference into correction coordinates and reconstructs it using the learned basis. **Predicted correction** uses coordinates predicted from the question-end hidden state to construct the intervention. The paired and reconstructed differences serve as reference conditions that require information unavailable during ordinary inference.

Reconstructed differences achieve answer accuracy close to that of directly applied paired differences, showing that encoding and reconstruction retain much of the useful correction effect. Predicted corrections achieve lower accuracy than these reference conditions, but improve both answer accuracy and the correct–incorrect logit margin over no intervention. Thus, the predictor provides effective corrections from the available question-end hidden state, despite the remaining gap to corrections constructed with paired responses.

Table 11: Effect of different correction sources on held-out AMBER samples with LLaVA-1.5. Paired and reconstructed differences require paired response information.

Correction source	Accuracy (%)↑	Logit margin↑
No intervention	75.94	1.2884
Paired difference	91.48	1.9457
Reconstructed difference	90.48	1.9971
Predicted correction	86.97	1.6622

C.7 Hallucination Mitigation Across Generation Stages

CSS constructs its training targets from response-end activation differences and uses the learned predictor throughout generation. We examine whether the resulting corrections mitigate hallucinations at early, middle, and late decoding stages. On LLaVA-1.5, we divide each response according to the relative position within its actual length and compare Origin with CSS at each stage.

As shown in Table 12, the original model exhibits higher CHAIR-S and CHAIR-I in later portions of its responses. CSS lowers both metrics across all three stages, with larger reductions in the middle and late stages. This pattern indicates that its corrections remain useful as generation proceeds and the response context grows. It also provides evidence that supervision derived from response-end differences can support interventions at different decoding positions. Recall decreases alongside the hallucination metrics, especially in the middle stage, indicating that some of the reduction is accompanied by lower object coverage.

Table 12: LLaVA-1.5 decoding-stage analysis, with stages defined relative to actual response length.

Method	Stage	CHAIR-S↓	CHAIR-I↓	Recall↑
Origin	Early	11.3	4.3	49.1
	Middle	32.0	17.9	19.6
	Late	42.0	29.7	7.5
Full CSS	Early	8.0	3.2	46.2
	Middle	19.3	12.0	15.6
	Late	22.0	15.8	7.3

D Limitations and Broader Impacts

D.1 Limitations

CSS learns its correction space from paired non-hallucinatory and hallucinatory trajectories, so its effectiveness depends on their quality and coverage. The activation differences may also encode answer polarity or response semantics. Our controls and transfer experiments support useful steering effects, but further work is needed to disentangle these factors and characterize the shift from response-end supervision to intermediate decoding states.

CSS uses a fixed steering layer, correction-space dimension, and steering strength for each backbone. These choices may require adjustment across models and tasks. Automatically selecting them could improve adaptability.

Our evaluation covers several backbones and standard hallucination benchmarks, but broader testing is needed for specialized domains and complex visual reasoning. Small performance margins also warrant repeated-run analysis. CSS provides no guarantee of fully faithful generation, and high-stakes deployment requires additional verification and human oversight.

D.2 Broader Impacts

CSS can improve the visual faithfulness of LVLM responses and reduce unsupported claims. Its lightweight inference-time intervention can be applied to existing models with frozen backbone parameters, potentially making hallucination mitigation more accessible. Related watermarking research addresses content traceability and the preservation of visual or semantic fidelity [38–40].

Reduced hallucination rates may encourage excessive trust in model outputs, which can still be incorrect under challenging or out-of-distribution conditions. Safety-critical applications require additional verification and human oversight.

This work uses public benchmarks and LVLM backbones and involves no private user data, human-subject experiments, or new sensitive datasets. Its intended use is to support research on reliable multimodal generation.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes].

   Justification: The abstract and introduction state the main contributions of the paper, including the empirical activation-difference analysis, the theoretical low-dimensional approximation result, the proposed CSS method, and the experimental validation on hallucination benchmarks.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes].

   Justification: We discuss limitations in the appendix, including the dependence on paired activation differences for learning the correction space, the evaluation scope across a limited set of LVLM backbones and benchmarks, and the sensitivity to steering-layer and hyperparameter choices.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes].

   Justification: The main paper states the key theoretical result on effective rank and low-dimensional approximation, and the full assumptions and proof are provided in the appendix.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes].

   Justification: The paper describes the evaluated benchmarks, LVLM backbones, compared baselines, metrics, and evaluation protocol. Additional implementation details, including training data for correction-space learning and hyperparameter choices, are provided in the appendix.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No].

   Justification: The public project repository is provided at `https://github.com/songboyangy/CSS`. This checklist answer reflects the availability of implementation files and reproduction instructions at the time of preparation.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes].

   Justification: The main paper reports the benchmarks, metrics, backbones, baselines, and evaluation protocol. Detailed settings, including correction-space training data, hyperparameters, and optimization details, are provided in the appendix.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No].

Justification: We do not report error bars or confidence intervals because the experiments involve large LVLM inference runs across multiple benchmarks and baselines, making repeated runs computationally expensive. We instead report consistent results across several benchmarks, backbones, and analysis settings.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No].

   Justification: Appendix B.3 reports the GPU model and memory capacity. Per-experiment execution times and total computational costs are not reported.

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes].

   Justification: The work uses public datasets and models for hallucination mitigation research and does not involve human-subject experiments, private data collection, or unsafe data release. The research is intended to improve the reliability of LVLM outputs.

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes].

    Justification: The paper discusses the positive impact of reducing hallucinations in LVLMs, which can improve reliability in multimodal applications. It also acknowledges that mitigation methods may be imperfect and should not be treated as complete safeguards for safety-critical deployment.

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A].

    Justification: This paper does not release a new pretrained model, image generator, or scraped dataset with high misuse risk. The proposed method is an inference-time steering approach evaluated on existing LVLMs and public benchmarks.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [No].

    Justification: The paper cites the models, benchmarks, and baseline methods used in the experiments, but does not explicitly list the licenses and terms of use for all existing assets.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A].

    Justification: The paper does not introduce or release a new dataset, benchmark, or pretrained model. It proposes a new inference-time steering method evaluated using existing assets.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A].

Justification: The paper does not involve crowdsourcing experiments or research with human subjects.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A].

Justification: The paper does not involve human-subject experiments, crowdsourcing, or collection of participant data, so IRB approval is not applicable.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A].

Justification: GPT-4o is used as the automatic evaluator for MMHal-Bench, as described in Appendix B.1. The method itself introduces no additional LLM-based component. Any use of LLMs for writing or formatting is outside the core methodology.