Supplementary materials for Nullu: Mitigating Object Hallucinations in Large Vision-Language Models via HalluSpace Projection

Le Yang*, Ziwei Zheng*, Boxu Chen, Zhengyu Zhao, Chenhao Lin, Chao Shen†
Xi'an Jiaotong University
Xi'an, 710049, China
{yangle15@, ziwei.zheng@stu., chenboxu@stu., zhengyu.zhao@}xjtu.edu.cn
{linchenhao@, chaoshen@mail.}xjtu.edu.cn

1. The derivation of null space

Here, we give the details about obtaining the null space of the $\boldsymbol{v}$. We want to proof that: any vector $\boldsymbol{z} \in \mathbb{R}^D$ in the null space of the $\boldsymbol{v} \in \mathbb{R}^D$, $(\boldsymbol{I} - \boldsymbol{v}\boldsymbol{v}^\top)$, is orthogonal to the vector $\boldsymbol{v}$. Namely, we have $\boldsymbol{v}^\top \boldsymbol{z} = 0$, where $\boldsymbol{v}$ is the vector with norm 1. $\boldsymbol{I}$ is the identity matrix with the size of $\mathbb{R}^{D\times D}$. We can write the $\boldsymbol{z}$ as

$$\boldsymbol{z} = (\boldsymbol{I} - \boldsymbol{v}\boldsymbol{v}^\top)\boldsymbol{m}, \quad \forall \boldsymbol{m} \in \mathbb{R}^D. \tag{1}$$

Then we have

$$\boldsymbol{v}^\top \boldsymbol{z} = \boldsymbol{v}^\top (\boldsymbol{I} - \boldsymbol{v}\boldsymbol{v}^\top)\boldsymbol{m} = (\boldsymbol{v}^\top - (\boldsymbol{v}^\top \boldsymbol{v})\boldsymbol{v}^\top)\boldsymbol{m}, \tag{2}$$

$$= (\boldsymbol{v}^\top - \boldsymbol{v}^\top)\boldsymbol{m} = 0, \ \forall \boldsymbol{m} \in \mathbb{R}^D. \tag{3}$$

Therefore, $(\boldsymbol{I} - \boldsymbol{v}\boldsymbol{v}^\top)$ is the null space of $\boldsymbol{v}$.

2. Decoding information in HalluSpace

As we state in the main paper, to explore the information behind the learned vectors $\boldsymbol{v}_i \in \mathbb{R}^D$, a common approach is to decode the embeddings with $\boldsymbol{O}\boldsymbol{v} \in \mathbb{R}^{|\mathcal{V}|}$, where $\boldsymbol{O} = [\boldsymbol{o}_1, \dots, \boldsymbol{o}_{|\mathcal{V}|}]^\top \in \mathbb{R}^{|\mathcal{V}|\times D}$ and $\mathcal{V}$ denotes the vocabulary. We then sort $\boldsymbol{O}\boldsymbol{v}$ in ascending order, find the top-m indices, and use the corresponding words to interpret $\boldsymbol{v}$.

Using Nullu, we can extract the $\boldsymbol{V}$ at different layers and then decode it via $\boldsymbol{O}\boldsymbol{v}$ to explore the internal information behind $\boldsymbol{V}$. We provide the decoding results in Table. 1. Moreover, we select the words with the most frequency in the output of LVLM with distorted images. For a more straightforward interpretation, we directly selected the words in Table. 1 to see the frequency of words before and after Nullu to see if the LLM biases are mitigated.

∗ Equal contribution. † Corresponding author.

layer	Top Tokens							
16	dynamic	either	further	above	background	floor	tables	...
17	another	notable	left	later	others	most	tables	...
18	nearby	notable	either	tables	group	optional	others	...
19	notable	middle	either	diverse	background	overall	concentr	...
20	notable	left	nearby	either	background	center	middle	...
21	middle	another	left	bottom	top	left	right	...
22	left	position	middle	another	right	background	top	...
23	notable	left	position	nearby	left	another	bottom	...
24	position	notable	various	various	middle	background	above	...
25	position	towards	left	nearby	right	another	bottom	...
26	in	position	towards	positions	left	nearby	engaged	...
27	in	position	towards	closer	nearby	right	background	...
28	in	the	position	closer	background	nearby	right	...
29	in	position	closer	towards	nearby	background	a	...
30	in	closer	nearby	right	left	another	top	...
31	closer	close	position	bottom	another	left	top	...

Table 1. LLaVA-1.5-7B, top-rank-4, each singular vector of the matrix is interpreted by identifying the top 10 tokens it represents. We use the output embedding vector e_j to find top-scoring tokens $j \in \mathcal{V}$ for maximizing $\langle v_i, e_j \rangle$. Tokens have been censored for readability.

3. Theoretical Analysis: How Nullu works?

3.1. Factor component analysis

The analysis is performed for each layer ℓ, and to avoid the notational burden, we will drop ℓ and focus on each layer separately. We use the same notations with these in the main paper. Based on the heuristic in [?], an embedding vector in any transformer layer can be decomposed into interpretable components. We suppose that the generated features $\boldsymbol{f}_i$ can be separated into three different elements:

$$\boldsymbol{f}_i \Rightarrow \underbrace{\hat{\boldsymbol{f}}_i \hat{\boldsymbol{B}}}_{\text{truthful contexts}} + \underbrace{\widetilde{\boldsymbol{f}}_i \widetilde{\boldsymbol{B}}}_{\text{hallucinated biases}} + \underbrace{\boldsymbol{u}_i}_{\text{noise}}. \tag{4}$$

Therefore, give positive and negative samples as input, we have

$$\boldsymbol{f}_i^+ = \underbrace{\hat{\boldsymbol{f}}_i \hat{\boldsymbol{B}}}_{\text{truthful contexts}} + \underbrace{\widetilde{\boldsymbol{f}}_i \widetilde{\boldsymbol{B}}}_{\text{hallucinated biases}} + \underbrace{\boldsymbol{u}_i^+}_{\text{noise}}. \tag{5}$$

$$\boldsymbol{f}_i^- = \underbrace{\hat{\boldsymbol{f}}_i \hat{\boldsymbol{B}}}_{\text{truthful contexts}} + \underbrace{\boldsymbol{u}_i^-}_{\text{noise}}, \tag{6}$$

based on which we have

$$\boldsymbol{E} = \boldsymbol{f}_i^+ - \boldsymbol{f}_i^- = \widetilde{\boldsymbol{f}}_i \widetilde{\boldsymbol{B}} + (\boldsymbol{u}_i^+ - \boldsymbol{u}_i^-). \tag{7}$$

The noise can be approximated to 0 on average of the whole data. The top-k singular vectors span exactly the same subspace of $\widetilde{\boldsymbol{B}}$, which can be the HalluSpace in our paper. Moreover, SVD is also efficient since SVD gives the best low-rank approximation of $\boldsymbol{E}$. Thus, our approach can be viewed as an approximate recovery of the latent subspace for hallucination semantics.

3.2. Connections to DPO

In this subsection, we try to establish the conceptual connection between DPO [?] and the proposed Nullu. Our study is mainly based on the theoretical analysis in [?], where a simple logistic model for the output token given the (continuing) prompt is used. In the following parts, we will drop ℓ and focus on each layer separately to avoid notational burden.

Although the proposed Nullu is designed for LVLMs, we mainly study its LLM parts, since our weight editing is mainly conducted on this part. Therefore, in this section, we use the term input to denote the extracted features $\boldsymbol{x}$, containing both the visual features processed by the previous visual encoder, and the text prompts projected into the embedding space. Given $\boldsymbol{x}$ with hallucinated response y^+ and truthful response y^-, where the corresponding embedding features denoted as $\boldsymbol{x}, \boldsymbol{y}^+, \boldsymbol{y}^-$ respectively, DPO optimizes the loss

$$\begin{aligned}\mathcal{L}_{\text{DPO}}(\pi_{\boldsymbol{\theta}}; \pi_{\text{ref}}) = &- \mathbb{E}_{(x,y^+,y^-)\sim\mathcal{D}}[\log\sigma(\beta\log\frac{\pi_\theta(\boldsymbol{y}^+|\boldsymbol{x})}{\pi_{\text{ref}}(\boldsymbol{y}^+|\boldsymbol{x})} \\ &- \beta\log\frac{\pi_\theta(\boldsymbol{y}^-|\boldsymbol{x})}{\pi_{\text{ref}}(\boldsymbol{y}^-|\boldsymbol{x})})],\end{aligned} \tag{8}$$

where, π_{ref} corresponds to the reference (or base) probability model generating output y given x, $\pi_{\boldsymbol{\theta}}$ is the new probability model (parametrized by $\boldsymbol{\theta}$), σ is the logistic function with $\sigma(z) = (1+\exp(-z))^{-1}$, and $\beta > 0$ is a hyperparameter. The gradient of the loss $\mathcal{L}_{\text{DPO}}$ with respect to θ at initialization $\pi_\theta = \pi_{\text{ref}}$ equals

$$\begin{aligned}&\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\text{DPO}}(\pi_{\boldsymbol{\theta}}; \pi_{\text{ref}}) \mid_{\pi_{\boldsymbol{\theta}}=\pi_{\text{ref}}} \\ &= -\beta\mathbb{E}_{(x,y^+,y^-)\sim\mathcal{D}}[\nabla_{\boldsymbol{\theta}}\log\pi(\boldsymbol{y}^+|\boldsymbol{x}) - \nabla_{\boldsymbol{\theta}}\log\pi(\boldsymbol{y}^-|\boldsymbol{x})] \mid_{\pi_\theta=\pi_{\text{ref}}}.\end{aligned} \tag{9}$$

Let $\mathcal{V}$ denote the vocabulary. We start with an input x (including both textual and visual features) and produce M next-token predictions $y_1, \cdots, y_M \in \mathcal{V}$ sequentially. Suppose the model sequentially predicts token y_m given $x_m := (x, y_1, \cdots, y_{m-1})$ for each $1 \leq m \leq M$, and let $\boldsymbol{x}_m$ denote the encoding of input x_m. We assume a logistic model generating each continuation y_m given x_m,

$$\pi_{\boldsymbol{\theta}}(y_m|x_m) \equiv \pi_{\boldsymbol{W}}(y_m|x_m) = Z_{m,\boldsymbol{W}}^{-1}\exp\left(\boldsymbol{o}_{y_m}^\top \boldsymbol{W}\boldsymbol{x}_m\right). \tag{10}$$

Here, $\boldsymbol{o}_{y_m}$ is the classification vector which we use to get the final word prediction, $\boldsymbol{W}$ is a weight matrix and $Z_{m,\boldsymbol{W}}$ is the normalizing constant:

$$Z_{m,\boldsymbol{W}} = \sum_{y\in\mathcal{V}}\exp\left(\boldsymbol{o}_y^\top \boldsymbol{W}\boldsymbol{x}_m\right).$$

For the results in Eq. (10), we have assumed for simplicity that the classification is performed with linearly transformed encoding $\boldsymbol{W}\boldsymbol{x}_m$ instead of the more common nonlinear transformations in the transformer architecture. And the output probability is given by the logistic model, based on which we can obtain the joint probability of observing the entire continuation $y = (y_1, \cdots, y_M)$ given the starting input x as

$$\begin{aligned}\pi_{\boldsymbol{\theta}}(y|x) \equiv \pi_{\boldsymbol{W}}(y|x) &= \prod_{m=1}^{M}\pi_{\boldsymbol{W}}(y_m|x_m) \\ &= Z_{\boldsymbol{W}}^{-1}\exp\left(\sum_{m=1}^{M}\boldsymbol{o}_{y_m}^\top \boldsymbol{W}\boldsymbol{x}_m\right),\end{aligned}$$

where $Z_{\boldsymbol{W}} = \prod_{m=1}^{M} Z_{m,\boldsymbol{W}}$. We denote by $x_m^\pm$, $\boldsymbol{x}_m^\pm$ and $\boldsymbol{o}_{y_m}^\pm$ the positive/negative inputs, the corresponding embedding and classification vector for the positive/negative continuation respectively. Plugging this into (9), the first step DPO update has gradient

$$\begin{aligned}&\nabla_{\boldsymbol{W}}\mathcal{L}_{\text{DPO}}(\pi_{\boldsymbol{W}}; \pi_{\text{ref}})|_{\pi_{\boldsymbol{W}}=\pi_{\text{ref}}} \\ &= -\beta\mathbb{E}_{(x,y^+,y^-)\sim\mathcal{D}}\left[\sum_{m=1}^{M}\left(\boldsymbol{o}_{y_m}^+(\boldsymbol{x}_m^+)^\top - \boldsymbol{o}_{\boldsymbol{y}_m}^-(\boldsymbol{x}_m^-)^\top\right)\right].\end{aligned} \tag{11}$$

Note that the the normalization factors $Z_{m,\boldsymbol{W}}$ (and hence $Z_{\boldsymbol{W}}$) are omitted when we take the difference of the gradients of the log-probabilities. With N pairs of inputs in $\mathcal{D}$, and we consider the case $M = 1$, the DPO gradient will be an average over all the pairs:

$$\nabla_{\boldsymbol{W}}\mathcal{L}_{\text{DPO}}(\pi_{\boldsymbol{W}}; \pi_{\text{ref}})|_{\pi_{\boldsymbol{W}}=\pi_{\text{ref}}} = -\frac{\beta}{N}\sum_{i=1}^{N}(\boldsymbol{o}_{y_i}^+(\boldsymbol{x}_i^+)^\top - \boldsymbol{o}_{y_i}^-(\boldsymbol{x}_i^-)^\top), \tag{12}$$

where the extra index i mean i-th sample pairs. The Eq.(12) is the formulation (7) in our main paper, which is

$$\begin{aligned}\nabla_{\boldsymbol{W}}\mathcal{L}_{\text{DPO}} &= -\frac{\beta}{N}\sum_{i=1}^{N}\left(\boldsymbol{o}_{y_i^+}(\boldsymbol{x}_i^+)^\top - \boldsymbol{o}_{y_i^-}(\boldsymbol{x}_i^-)^\top\right) \\ &= -\frac{\beta}{N}\sum_{i=1}^{N}\big(\underbrace{\boldsymbol{o}_{y_i^+}(\boldsymbol{x}_i^+ - \boldsymbol{x}_i^-)^\top}_{\text{feature difference}} + \underbrace{(\boldsymbol{o}_{y_i^+} - \boldsymbol{o}_{y_i^-})(\boldsymbol{x}_i^-)^\top}_{\text{output difference}}\big).\end{aligned} \tag{13}$$

The gradient contains a feature difference term. Therefore, the gradient update can be interpreted as an attempt to eliminate feature differences to avoid hallucinated responses. For Nullu, it tries to approximate such difference via SVD and also attempts to eliminate it by null space projection, which shows the connection between Nullu and DPO.

4. Implementation Details of LVLMs

This section details the implementation of the evaluated LVLMs and the methods used for OH mitigation. The overall experimental setup is summarized in Table 2. Unlike the standard greedy method, which selects the most probable token at each decoding step, beam search maintains a fixed number of candidate sequences (beams) per step, ranking them based on the accumulated probability scores of the previous tokens $(y_{<t})$. In our experiments, the beam search method uses a *num-beams* setting of 3, specifying the number of candidate sequences retained at each step. We use the default code for implementation of these two baselines in HuggingFace Transformers Repository[**?**].[1]

Parameters	**Value**
Do-sample	False
Num-beams (for beam search)	3
Maximum New Tokens (CHAIR)	64
Maximum New Tokens (POPE)	64
Maximum New Tokens (MME)	64
Maximum New Tokens (OPOPE)	256
Maximum New Tokens (LLaVA-Bench)	1024

Table 2. Hyper-parameters for LVLMs.

The complete hyper-parameters for Nullu across different models in our experiments are as follows. Specifically, there are three major hyper-parameters that can be actively adjusted to optimize Nullu's effectiveness across different models:

1. Editing Layers ℓ: For all models, the editing layers are specified by $\ell \in \text{range}(16, 32)$.
2. The Selected Top-k singular vector: The number of top-k singular vectors selected varies by model. We use the value 4 for LLaVA-1.5 on both CHAIR and POPE. Similarly, we use 8 for MiniGPT-4 on the evaluated two datasets. For mPLUG-Owl2, we use 32 on CHAIR and 16 on POPE.
3. Num-beams: This parameter also differs across models. It is set to 3 for both LLaVA-1.5 and MiniGPT-4, while for mPLUG-Owl2, it is set to 1.

For the comparison of Nullu with SOTAs methods specifically designed for OH mitigation, the evaluation code is built based on the public repository of HALC [**?**][2]. Specifically, the hyper-parameters for HALC, VCD [**?**], DoLa [**?**] and OPERA [**?**] are reported in Table 3, Table 4, Table 5 and Table 6, respectively. For each baseline, we follow the official implementation and use the pre-trained models and configurations from their respective repositories to reproduce the reported results.

[1] https://huggingface.co/docs/transformers
[2] https://github.com/BillChan226/HALC

Parameters	**Value**
Amplification Factor α	0.05
JSD Buffer Size m	6
Beam Size	1
FOV Sampling	Exponential Expansion
Number of Sampled FOVs n	4
Exponential Growth Factor λ	0.6
Adaptive Plausibility Threshold	0.1

Table 3. HALC Hyperparameter Settings

Parameters	**Value**
Amplification Factor α	1
Adaptive Plausibility Threshold β	0.1
Diffusion Noise Step	500

Table 4. VCD Hyperparameter Settings

Parameters	**Value**
Repetition Penalty θ	1.2
Adaptive Plausibility Threshold β	0.1
Pre-mature Layers	$[0, 2 \cdots, 32]$

Table 5. DoLa Hyperparameter Settings

Parameters	**Value**
Self-attention Weights Scale Factor θ	50
Attending Retrospection Threshold	15
Beam Size	3
Penalty Weights	1

Table 6. OPERA Hyperparameter Settings

5. POPE Settings and Additional Results

Polling-based Object Probing Evaluation (POPE) [**?**], presents a streamlined approach to assess object hallucination. POPE interacts directly with the examined LVLM, which distinguishes it from CHAIR. Within this benchmark, LVLMs are queried to answer if a specific object exists in the given image. The ratio between queries probing existent objects and non-existent objects is balanced (i.e.,50% vs. 50%). It encompasses three sampling settings: *random, popular, and adversarial*, each distinct in constructing negative samples. In the *random* setting, objects absent from the image are chosen randomly. The *popular* setting selects missing objects from a high-frequency pool, while in the *adversarial* setting, co-occurring objects not present in the image are prioritized. We use the POPE benchmark aggregating data from MSCOCO [**?**]. For each experiment, we select 500 images under each sampling setting and generate 6 questions per image. The evaluation pivots on four key metrics: Accuracy, Precision, Recall, and the F1 score.

5.1. POPE Results

We conduct the comparison between the raw LVLMs and the one implemented with Nullu on POPE and provide the results in Table 8.

We also tested different OH methods on MiniGPT-4 and provided the results in Table 7. The results show that Nullu outperforms all other methods by a significant margin regarding the accuracy and F1 score across all three types of POPE VQA tasks (random, popular, adversarial). Our experiments show that the MiniGPT-4 tends to provide the answer with "yes", which leads to a high recall ratio for most tested OH methods. However, the Precision of these methods is generally lower than 60%, resulting in a lower F1 score. However, Nullu significantly improves the Precision of the MiniGPT-4, resulting in a noticeable improvement in the F1 score. Moreover, we also see that VCD also has a lower recall, indicating that the LLM bias of MiniGPT makes the model tend to provide the answer with "yes" when responding.

5.2. OPOPE results

While this interaction is not problematic for evaluating decoding-based baselines, it limits the applicability of POPE to post-hoc OH mitigation methods. This direct interaction also creates greater instability when the examined LVLM is based on smaller language backbones, such as LLaMA-7B, which has less robust chat capabilities. To address these issues, offline POPE (OPOPE) was introduced in HALC [**?**], where a comparison is made between this approach and other effective decoding methods.

Since OPOPE evaluates directly based on the caption generated for each image, it follows the caption generation procedure from CHAIR but differs in the subsequent metric calculation. When computing the OPOPE scores, we follow the processing procedure of CHAIR while adopting POPE's metric calculation methodology.

For every sampled 500 images in the validation split of MSCOCO. The captions generated by the models are tokenized separately and then each word is singularized. Subsequently, the words are mapped to MSCOCO objects using the synonym and double-word lists provided in [**?**].

Next, three hallucination test object lists are constructed following the sampling strategies proposed in the POPE method. We refer detailed explanations of the different options to its original paper[**?**]. Each list contains six objects, with a 1:1 ratio of ground-truth to nonexistent objects to ensure label balance. These lists are originally used to generate polling questions based on the template "*Is there a/an* $\{\}$ *in the image?*" in [**?**].

After obtaining the objects set from the generated captions and the three test objects list, we assess whether the captions include the ground-truth or nonexistent objects. The comparison results are used to compute scores as the score of the corresponding sampling strategy setting.

The primary metric in OPOPE is adjusted to enable more reliable comparisons. Since offline evaluations are less likely to include the exact hallucinated objects in descriptions, false negatives (FNs) and the resulting recall become less reliable. To address this, and in line with HALC, we adopt F-beta as the main metric for OPOPE instead of F-1, reducing the emphasis on FNs. Specifically, the F-beta score is defined as: $F_\beta = (1+\beta^2)\cdot(\text{precision}\cdot\text{recall})/(\beta^2\cdot \text{precision} + \text{recall})$, where $\beta = 0.2$ is used throughout our experiments following [**?**].

The detailed and comprehensive evaluation results under each sampling strategy incorporating OPOPE are presented in Table 9. From the results, we see that our method achieve 7 best results (denoted by bold) in 9 comparisons, which again demonstrates the effectiveness of our method.

6. MME Numerical Results

In Table 10, we present the performance of the edited LLaVA-1.5 baselines on the perception-related tasks of the MME benchmark.

The baselines demonstrate consistent performance patterns, with Nullu uniformly improving the perceptual competencies of the LVLM model. Specifically, the edited model shows improvement for tasks typically used to estimate hallucination capability [**?**], including color, existence, count, and position. Furthermore, likely due to Nullu's effect in alleviating language priors, the model exhibits enhancements across all tasks, particularly in OCR, achieving an additional 84.35-point improvement in the total score. Furthermore, Table 11 showcases the performances on recognition-related tasks within the MME benchmark. The results suggest that implementing Nullu while mitigating hallucination issues and enhancing perceptual capabilities does not compromise the inherent reasoning abilities of LVLM. This is evident from the consistent overall recognition scores, which indicate that the model's fidelity remains unaffected by the intervention. Nullu significantly surpasses the original model, demonstrating a comprehensive performance improvement in reducing OH while maintaining generation quality.

Setting	Model	Method	Accuracy	Precision	Recall	F_1 Score
Random	MiniGPT4	Greedy	64.33	58.66	97.13	73.14
		Beam Search	62.10	57.15	96.67	71.84
		DoLa	64.27	58.82	95.10	72.68
		VCD	57.90	55.69	77.27	64.73
		HALC	64.87	59.04	97.13	73.44
		Nullu	77.23	76.54	78.53	**77.53**
Popular	MiniGPT4	Greedy	56.63	53.66	97.13	69.13
		Beam Search	56.47	53.58	96.67	68.95
		DoLa	56.58	53.72	95.10	68.65
		VCD	55.30	53.59	79.20	63.92
		HALC	57.00	53.88	97.13	69.31
		Nullu	70.13	67.24	78.53	**72.45**
Adversarial	MiniGPT4	Greedy	55.17	52.81	97.13	68.42
		Beam Search	55.50	53.02	96.67	68.48
		DoLa	55.85	53.28	95.10	68.29
		VCD	52.90	51.99	75.60	61.61
		HALC	55.53	53.02	97.13	68.60
		Nullu	66.70	63.50	78.53	**70.22**

Table 7. POPE results with random, popular and adversarial samplings compared to existing OH mitigation methods.

Setting	Model	Method	Accuracy	Precision	Recall	F_1 Score
random	LLaVA-1.5	Original	88.98	88.65	89.43	89.03
		Nullu	89.45	91.41	87.10	**89.20**
	MiniGPT4	Original	64.33	58.66	97.13	73.14
		Nullu	77.23	76.54	78.53	**77.53**
	mPLUG-Owl2	Original	81.83	77.80	89.07	83.06
		Nullu	83.33	79.10	90.60	**84.46**
popular	LLaVA-1.5	Original	84.58	81.61	89.43	85.32
		Nullu	85.37	84.25	87.10	**85.63**
	MiniGPT4	Original	56.63	53.66	97.13	69.13
		Nullu	70.13	67.24	78.53	**72.45**
	mPLUG-Owl2	Original	75.77	70.35	89.07	78.61
		Nullu	77.47	71.75	90.60	**80.08**
adversarial	LLaVA-1.5	Original	77.97	72.79	89.43	80.24
		Nullu	79.40	75.51	87.10	**80.88**
	MiniGPT4	Original	55.17	52.81	97.13	68.42
		Nullu	66.70	63.50	78.53	**70.22**
	mPLUG-Owl2	Original	72.77	67.17	89.07	76.58
		Nullu	74.03	68.05	90.60	**77.72**

Table 8. Results on POPE. Original denotes direct sampling for LVLMs, whereas Nullu refers to edit the model with the proposed method.

Setting	Model	Method	Accuracy	Precision	Recall	F Score
Random	LLaVA-1.5	Greedy	81.52	98.41	64.07	96.42
		Beam Search	81.67	98.67	64.20	**96.67**
		DoLa	81.38	98.11	64.00	96.14
		OPERA	81.62	98.57	64.17	96.58
		VCD	80.57	98.41	62.13	96.25
		HALC	79.58	98.21	60.27	95.89
		Nullu	81.18	98.05	63.63	96.05
	MiniGPT-4	Greedy	72.42	98.49	45.53	94.25
		Beam Search	72.65	98.70	45.90	94.51
		DoLa	72.45	98.58	45.57	94.34
		OPERA	72.57	98.77	45.70	94.52
		VCD	72.35	98.19	45.53	93.97
		HALC	72.08	98.62	44.80	94.25
		Nullu	72.68	99.06	45.80	**94.82**
	mPLUG-Owl2	Greedy	79.45	97.74	60.30	95.46
		Beam Search	79.45	97.52	60.43	95.27
		DoLa	78.33	97.60	58.10	95.09
		OPERA	78.31	97.73	57.96	95.21
		VCD	78.19	98.23	57.42	95.61
		HALC	77.83	97.72	57.00	95.10
		Nullu	80.30	98.40	61.60	**96.19**
Popular	LLaVA-1.5	Greedy	78.93	91.17	64.07	89.71
		Beam Search	79.30	91.98	64.20	90.47
		DoLa	78.72	90.69	64.00	89.26
		OPERA	79.22	91.80	64.17	90.30
		VCD	77.57	89.87	62.13	88.35
		HALC	77.47	91.87	60.27	90.05
		Nullu	79.80	94.06	63.63	**92.36**
	MiniGPT-4	Greedy	70.80	92.01	45.53	88.53
		Beam Search	71.32	93.35	45.90	89.77
		DoLa	70.90	92.33	45.57	88.82
		OPERA	71.10	92.82	45.70	89.27
		VCD	70.33	90.30	45.53	86.98
		HALC	70.92	93.80	44.80	90.00
		Nullu	71.97	96.08	45.80	**92.19**
	mPLUG-Owl2	Greedy	76.00	87.90	60.30	86.38
		Beam Search	75.90	87.50	60.43	86.02
		DoLa	75.20	88.36	58.10	86.60
		OPERA	75.02	88.06	57.96	86.33
		VCD	74.86	88.16	57.42	86.37
		HALC	75.77	91.34	57.00	89.26
		Nullu	78.20	92.22	61.60	**90.49**
Adversarial	LLaVA-1.5	Greedy	76.97	86.36	64.07	85.22
		Beam Search	77.27	86.92	64.20	85.75
		DoLa	76.85	86.18	64.00	85.05
		OPERA	77.03	86.40	64.17	85.26
		VCD	75.88	85.71	62.13	84.48
		HALC	76.57	89.44	60.27	**87.80**
		Nullu	77.58	88.27	63.63	86.98
	MiniGPT-4	Greedy	70.43	90.65	45.53	87.32
		Beam Search	70.98	92.06	45.90	88.63
		DoLa	70.50	90.85	45.57	87.50
		OPERA	70.78	91.63	45.70	88.21
		VCD	69.82	88.43	45.53	85.32
		HALC	70.52	92.22	44.80	88.60
		Nullu	71.10	92.73	45.80	**89.21**
	mPLUG-Owl2	Greedy	74.23	83.58	60.30	82.36
		Beam Search	73.78	82.51	60.43	81.37
		DoLa	73.52	83.98	58.10	82.55
		OPERA	73.17	83.45	57.96	82.06
		VCD	72.85	83.01	57.42	81.61
		HALC	74.02	86.41	57.00	84.72
		Nullu	76.90	88.76	61.60	**87.28**

Table 9. Detailed OPOPE results with random, popular and adversarial samplings.

Model	Method	*Existence*	*Count*	*Position*	*Color*	*Posters*		***Perception Total***
LLaVA-1.5	Original	$181.67_{\pm 2.36}$	$118.33_{\pm 12.47}$	$104.44_{\pm 10.39}$	$152.78_{\pm 5.67}$	$117.23_{\pm 4.79}$	Original	$1246.36_{\pm 5.79}$
	Nullu	$\mathbf{190.00}_{\pm 4.08}$	$\mathbf{121.11}_{\pm 7.74}$	$\mathbf{105.56}_{\pm 8.20}$	$\mathbf{156.67}_{\pm 9.81}$	$\mathbf{127.55}_{\pm 4.20}$		
Model	Method	*Celebrity*	*Scene*	*Landmark*	*Artwork*	*OCR*		
LLaVA-1.5	Original	$111.67_{\pm 3.90}$	$144.83_{\pm 1.50}$	$130.65_{\pm 5.26}$	$108.92_{\pm 2.99}$	$75.83_{\pm 5.89}$	Nullu	$\mathbf{1330.71}_{\pm 19.77}$
	Nullu	$\mathbf{115.59}_{\pm 6.60}$	$\mathbf{147.92}_{\pm 1.36}$	$\mathbf{131.66}_{\pm 1.09}$	$\mathbf{113.00}_{\pm 2.07}$	$\mathbf{121.67}_{\pm 8.25}$		

Table 10. Results on all MME perception-related tasks.

Model	Method	*Common Sense Reasoning*	*Numerical Calculation*	*Text Translation*	*Code Reasoning*	***Recognition Total***
LLaVA-1.5	Original	$111.19_{\pm 4.68}$	$59.17_{\pm 7.73}$	$79.17_{\pm 8.25}$	$71.67_{\pm 11.24}$	$321.19_{\pm 2.15}$
	Nullu	$\mathbf{112.14}_{\pm 3.55}$	$\mathbf{65.00}_{\pm 16.20}$	$\mathbf{81.67}_{\pm 7.73}$	$\mathbf{92.50}_{\pm 15.94}$	$\mathbf{351.31}_{\pm 2.61}$

Table 11. Results on all MME recognition-related tasks.

7. Analysis about HalluSpace

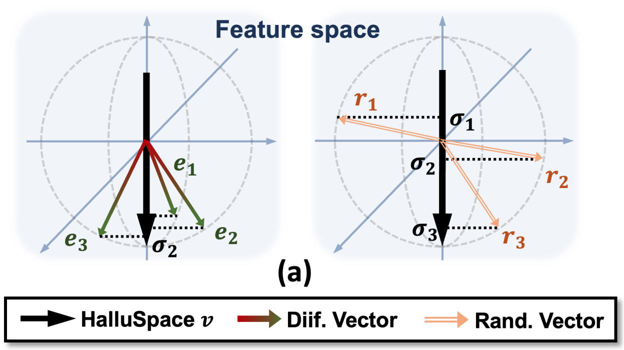


Figure 1. The illustration of difference vectors and random vectors in the feature space.

This section provides a more comprehensive study about the question **Does HalluSpace represent the hallucination biases?**. In other words, can the HalluSpace learned from the prepared hallucinated pairs adequately represent the true OH during the test? Ideally, if HalluSpace effectively represents these biases, the difference vectors from test samples with few OH issues after editing should have larger projected components when mapped onto HalluSpace than these random ones. Indeed, if the HalluSpace represents the OH problematic direction, the aforementioned difference vectors from test samples should gather together around this direction. This is further illustrated in Figure. 1.

To evaluate this, we select 100 test samples from CHAIR where Nullu successfully mitigates OH issues. We compute difference vectors $\boldsymbol{e}_i$ for each sample between the raw and edited LLaVA features. Moreover, we generate 100 random vectors $\boldsymbol{r}_i$ as a comparison baseline. All these vectors are normalized to avoid the effects of norms. Moreover, we use σ_i to represent the projected components. Figure. 1 shows the distribution of vectors on a normalized sphere.

Given $\boldsymbol{V}_4$ (rank-4), each projected component σ_i resides within $\mathbb{R}^4$. We then calculated $\sigma_i = \boldsymbol{e}_i \boldsymbol{V}_4$ for all selected samples and random vectors ($\sigma_i = \boldsymbol{r}_i \boldsymbol{V}_4$), averaging $||\sigma||$ across samples. The results is provided in Table 12. The table shows that the average $||\sigma||$ of difference vectors across layers is significantly larger ($10\times$) than that of random vectors. Since the selected test samples were successfully edited to avoid OH, this evidence suggests that HalluSpace captures directions in the feature space associated with hallucination and inaccuracies in LVLM responses.

8. LLaVA-Bench

8.1. Prompt for GPT-4V Aided Evaluation

As we leverage LLaVA-Bench [**?**] to qualitatively evaluate the overall performance using GPT-4V Aided Evaluation[3], in this section, we main describe the prompt used for evaluation. The assessments using GPT-4V are based on the accuracy and level of detail in the responses generated by LVLMs, following the approach described in [**?**]. The specific prompt structure is detailed in Table 13. During the evaluation, we collect the responses from two different LVLMs and then use the responses to replace the "{*Response*}" in the prompt, which is then sent to GPT-4V for scoring. Next, we analyze the GPT-4V outputs to assess the accuracy and detailedness of the LVLMs' responses. We further provide an evaluation example in Table 14 to further illustrate this process.

8.2. More case studies

Additional case studies on the LLaVA-bench are presented in Figure 3 and Figure 4 to illustrate the effectiveness of our approach. Note that the case in Figure 4 provides an example that the proposed Nullu can correctly generate an HTML

[3] https://openai.com/research/gpt-4v-system-card

Layers	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
Diff.	0.269	0.266	0.270	0.270	0.279	0.278	0.284	0.279	0.287	0.283	0.288	0.292	0.291	0.293	0.300	0.386
Rand	0.023	0.025	0.027	0.026	0.026	0.021	0.022	0.022	0.023	0.025	0.027	0.026	0.025	0.029	0.030	0.021

Table 12. Norm average of difference vectors and random vectors at different layers in the LVLM.

Description:
AI that scores image description accuracy and detailedness.

Instructions:
You are an AI designed to evaluate and score the performance of two AI assistants in describing a given image. Your primary focus is on the accuracy and detailedness of their descriptions. You will assess the accuracy by checking for hallucinations - any part of the description that is inconsistent with the image content. For detailedness, you will consider how rich the response is in necessary details, excluding any hallucinated parts. You will provide scores on a scale from 1 to 10 for each assistant separately, based on these criteria. After scoring, you will offer an explanation for your evaluation, ensuring it is free from bias and not influenced by the order of presentation of the responses.

Input format:

[Assistant 1]
{Response 1}
[End of Assistant 1]

[Assistant 2]
{Response 2}
[End of Assistant 2]

Output format:

Accuracy:
Scores of the two answers:
Reason:

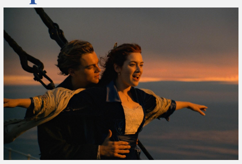
Detailedness:
Scores of the two answers:
Reason:

Table 13. The configuration to build an image-description evaluator with GPT-4V

script that meets the need in the text prompts, which also corresponds to the experimental results on MME, where the edited model is shown that can achieve better performance in Code Reasoning tasks, further demonstrating the effectiveness of our method. The generated HTML website is shown in Figure 2.

Figure 2. The generated website of Nullu.

Input:



[Assistant 1]

In the context of the movie, this scene is significant as it captures a romantic moment between the two main characters, Jack and Rose, who are sharing a passionate kiss while being on a boat in the ocean. This scene is crucial as it highlights the emotional connection between the two characters and adds to the overall narrative of the movie. The kiss takes place during a storm, which adds to the intensity and drama of the scene, further emphasizing the strong bond between the two characters and their willingness to face challenges together.

[End of Assistant 1]

[Assistant 2]

In the context of the movie "Titanic", this scene of a man and a woman (Rose DeWitt Bukater and Jack Dawson) hanging onto a rope while being on a boat or ship at sea signifies their romantic connection and the beginning of their relationship. It's a pivotal moment in their story, as they first met and fell in love onboard the Titanic. This scene, along with others, portrays their journey of love, loss, and redemption, which ultimately led to their tragic fate. The romantic and emotional connection between Rose and Jack, as well as their shared experiences onboard the Titanic, form the core of their relationship and the movie's themes.

[End of Assistant 2]

GPT4-V Output:

Accuracy:

Scores of the two answers:
Assistant 1: 1
Assistant 2: 7
Reason: Assistant 1 inaccurately describes the scene as featuring a passionate kiss during a storm, which is not depicted in the image. The image shows no kiss or stormy weather; instead, it captures a moment where the characters are standing at the bow of the ship with arms outstretched. This is a significant misinterpretation of the content, leading to a low accuracy score.
Assistant 2, while not perfectly accurate, describes the scene with less deviation from the image. The mention of them "hanging onto a rope" and it being "a pivotal moment in their story" slightly strays from what is visually depicted, as they are not hanging but rather standing with arms extended and supported. The response is closer to the actual content, thus earning a higher score.

Detailedness:

Scores of the two answers:
Assistant 1: 3
Assistant 2: 5
Reason: Assistant 1 provides a narrative with emotional context, describing the intensity and drama of a supposed stormy scene. However, this description does not match the calm, sunset setting in the image and introduces elements like a kiss that aren't present. While rich in narrative detail, it mismatches the actual visual content significantly.
Assistant 2 provides more contextual information about the characters and their relationship within the movie, mentioning key themes of love, loss, and redemption. Although it erroneously mentions them "hanging onto a rope" and meeting for the first time in this scene, it still delivers a more aligned description of the emotional and thematic elements present in the actual scene depicted, hence a slightly higher score.

Table 14. Evaluation example for the image-description evaluator with GPT-4V.

Text prompts: Describe this photo in detail.

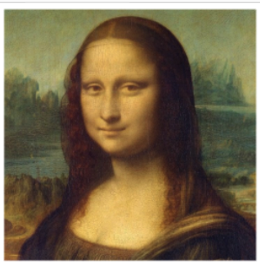
Raw: The image shows **a tall, white building with a large, round dome** on top. The building is surrounded by trees and has a large, open area in front of it. There are several other buildings visible in the background, including one that appears to be a hotel or office building. The sky is clear and blue, with a few clouds scattered across it.

Nullu: This is a photograph of the Space Needle in Seattle, Washington. The Space Needle is a towering structure with a rotating observation deck and a revolving restaurant. It was built for the 1962 World's Fair and has become an iconic symbol of Seattle and the Pacific Northwest. The Space Needle stands at over 600 feet tall and offers stunning views of the city and Puget Sound.

Text prompts: Describe this photo in detail.

Raw: The image features a close-up of two large subway sandwiches, each with a variety of toppings. The sandwiches are **placed on a table**, and their toppings include lettuce, tomatoes, and cheese. The sandwiches are cut in half, making it easy to see the delicious ingredients inside. The presentation of the sandwiches is appetizing and inviting, showcasing the fresh and flavorful options available at Subway.

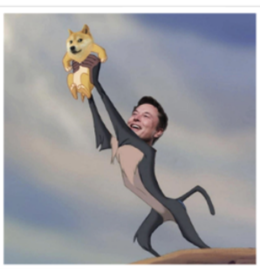
Nullu: The image features a close-up view of a Subway Series sandwich, showcasing its delicious ingredients. The sandwich is cut in half, revealing its layers of meat, lettuce, and cheese. The sandwich is generously filled with meat and cheese, making it look very appetizing. The Subway Series logo is also visible on the sandwich, emphasizing its branding. The sandwich is presented on a black background, drawing attention to its mouthwatering appearance.



Text prompts: Describe this photo in detail.

RAW: The image features a painting of a woman with a smile on her face, wearing a veil. She is the main subject of the painting, and her facial expression and pose create a sense of warmth and charm. The painting is set in a landscape, with a hillside visible in the background. There are also **two other people** in the painting, **one located on the left side and the other on the right side**. These figures appear to be smaller and less prominent than the main subject, adding depth and interest to the composition.

Nullu: The painting portrays a woman's face, which resembles that of Leonardo da Vinci's famous Mona Lisa. She wears a veil over her face and a Renaissance-style dress. The woman's face and expression evoke a sense of mystery and enigma. The painting's background features a landscape, including a body of water, a mountainous region, and a rocky outcrop. The woman's face and the landscape's elements work together to create a harmonious and visually appealing composition.



Text prompts: How does this meme reflect or comment on Elon Musk's public image, personality, or actions?

Raw: The meme of Elon Musk holding a dog in a **spacesuit** reflects or comments on his public image, personality, or actions by portraying him as a playful and eccentric individual. The image of Elon Musk holding a dog in a **spacesuit** is a lighthearted and humorous take on his work in the space industry and his ambitious plans for colonizing Mars. It also highlights his eccentric personality, as he is known for his unconventional ideas and actions. The meme serves as a way to entertain and engage with his fans and followers, while also showcasing his unique perspective on space exploration and technology.

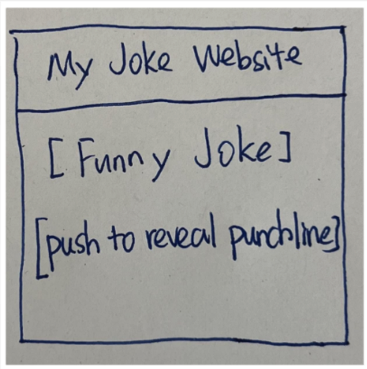
Nullu: The meme of Elon Musk holding a dogecoin dog in the air reflects or comments on his public image, personality, or actions by humorously portraying him as a dog lover and a fan of the cryptocurrency dogecoin. This meme is lighthearted and playful, poking fun at Musk's eccentricities and his sometimes controversial behavior. It also highlights the widespread fascination with and obsession with Musk, as well as the popularity of dogecoin, a cryptocurrency that gained significant attention and value during its peak in 2021.

Text prompts: Which iconic movie scene is being parodied in the meme?

Raw: The meme is parodying the iconic scene from the movie "The Lion King," where Simba holds up his father, **Mufasa**, in the air.

Nullu: The meme is parodying the iconic movie scene from "The Lion King," where Simba is lifted into the air by Rafiki.

Figure 3. More examples of Nullu on LLaVA-Bench.



Text prompts: Write brief HTML/JS to turn this mock-up into a colorful and interactive website, where the joke is replaced by a real joke.

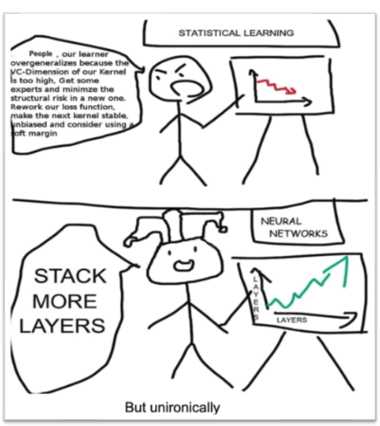
RAW:
Sure, here is the HTML code for the image:
```
<img src="image.jpg" alt="Image of a joke">
```
This will display the image with the alt text `Image of a joke`. If you want to replace the joke with a real one, you can add it to the `src` attribute like this:
```
<img src="image.jpg" alt="Image of a joke">
```
I hope this helps! Let me know if you have any other questions.

Nullu:
Sure, here is a simple HTML/JS code to turn the mock-up into a colorful and interactive website:
```
<!DOCTYPE html>
<html>
 <head>
  <title>My Joke Website</title>
 </head>
 <body>
  <h1>Welcome to My Joke Website</h1>
  <img src="image.jpg" alt="Joke Image">
  <p id="joke">Click here to reveal the joke</p>
  <button id="reveal-joke">Reveal Joke</button>
  <script>
   // Get the button element
   var revealJokeButton = document.getElementById('reveal-joke');

   // Add an event listener to the button
   revealJokeButton.addEventListener('click', function() {
    // Get the joke text element
    var jokeText = document.getElementById('joke');

    // Show or hide the joke text
    if (jokeText.style.display === 'none') {
     // Show the joke text
     jokeText.style.display = 'block';
    } else {
     // Hide the joke text
     jokeText.style.display = 'none';
    }
   });
  </script>
 </body>
</html>
```
This code creates a simple HTML page with a button that, when clicked, shows or hides a joke text element with the id `joke`. The button is given an event listener that listens for a `click` event and, when triggered, changes the display style of the joke text element from `none` to `block` or vice versa.

Note that you will need to replace `image.jpg` with the actual image file you want to use as the joke image.



Text prompts: Give a detailed description of this meme.

RAW: The meme is a cartoon depicting a man standing in front of a graph, which shows a correlation between people and layers. The man is **holding a sign that says** "stack more layers." The graph shows a positive correlation between the number of layers and the performance of the system. However, the man is unaware of the negative impact of this correlation, as the graph also shows that the system becomes less efficient as more layers are added. The cartoon humorously highlights the potential pitfalls of over-engineering and the importance of balancing complexity with efficiency.

Nullu: The meme is a cartoon that discusses statistical learning and neural networks. It starts with a man overseeing the C-Dimension of his kernel, which is a mathematical concept. The man is trying to get some structure into the data. The cartoon then moves on to describe the process of stacking more layers in neural networks. The man is shown stacking more layers, and the cartoon ends with the phrase "But unironically."

Figure 4. More examples of Nullu on LLaVA-Bench.