Supplementary Materials

A. Related Works

Hallucination Mitigation in LVLMs The hallucination phenomenon in LVLMs can originate from either the visual encoder [9, 11] or the pretrained LLM [15, 16]. These components may fail to fully align visual and textual representations, leading to inconsistencies in generated outputs. To address this issue, various visual encoders have been developed to enhance the quality of processed images, ensuring more accurate and contextually relevant outputs [1, 4, 50]. Additionally, fine-tuning LVLMs on datasets specifically curated to address hallucination has proven effective in enhancing alignment [35, 40, 54]. Another promising approach is contrastive decoding, which leverages the difference between image-conditioned and image-free token probabilities during decoding stage to prioritize tokens that are grounded in the visual information [6, 14, 16].

In this paper, we primarily focus on addressing hallucination of LVLMs through preference alignment, where it is critical to construct informative and high-quality preference pairs to guide the model in generating grounded responses. Numerous methods have been proposed for constructing offline hallucination preference datasets. These include contaminating or removing image content to create negative samples [27, 40, 44], injecting hallucination into textual responses to generate negative samples [32, 54], and leveraging human annotators or external expert models, such as GPTs, to refine generated responses and construct positive samples [43, 53]. Some works also propose constructing preference dataset in an on-policy manner [47, 49, 55].

Preference Alignment. Preference alignment has emerged as a cornerstone methodology for enhancing the response quality of LLMs [2, 5, 25, 39]. Central to this approach is reinforcement learning from human feedback (RLHF) [2, 25], which involves training a reward model to capture human preferences and then using reinforcement learning algorithms, such as Proximal Policy Optimization (PPO) [34], to guide LLMs toward generating responses with higher rewards. However, RL-based methods often face challenges related to instability during training. Consequently, recent research has shifted toward developing simpler and more stable alternatives to RLHF. A notable approach is DPO [28], which implicitly optimizes the same objective as RLHF but achieves human preference alignment through a single cross-entropy loss, bypassing the need for learning the explicit reward model and the complex reinforcement learning stage.

The simplicity of DPO has inspired a wave of subsequent alternatives for hallucination mitigation in LVLMs [40, 48, 49, 53, 54]. Corresponding to how the dataset is constructed, the DPO algorithm can be tailored to address specific alignment challenges. For instance, some approaches focus on fine-grained preference feedback, enabling more nuanced alignment by capturing segment-level hallucination in responses [32, 43, 48]. Other than alignment on offline dataset, on-policy DPO [49] or its alternatives [47] emphasize aligning the model on its own generated outputs rather than the offline dataset. Furthermore, iterative DPO [49, 55] introduces an iterative updating paradigm similar to the standard RL process, progressively improving alignment over multiple iterations.

B. Theoretical Proof

B.1. Definition of on/off-policy in Preference Alignment

As stated in [8], the key distinction between on-policy and off-policy lies in whether the training data used to optimize the current policy is generated by the current policy itself. If the data is generated by the current policy, it is considered on-policy; otherwise, it is off-policy.

For a given completion y, if it is collected in an on-policy manner, i.e., $y \sim \pi_\theta(\cdot|x)$, then the current policy has a higher probability of generating that completion. In contrast, if the distribution that generated y differs from the current policy (which, in general, is a significant difference), the probability of the current policy generating y is relatively low, potentially approaching zero.

B.2. Proof of Remark 4.1

Proof. The next-token prediction task is typically trained via maximum likelihood estimation (MLE), which is equivalent to minimizing the cross-entropy loss. Let $\boldsymbol{x} = (x_1, x_2, \ldots, x_i)$ denote a token sequence prefix, and let $y = x_{i+1}$ denote the next token to be predicted. A large language model with vocabulary size $\mathcal{V}$ maps $\boldsymbol{x}$ to a d-dimensional feature vector $\phi(\boldsymbol{x}) \in \mathbb{R}^d$ via a deep neural network. The model then computes a logit vector $\boldsymbol{z} = \boldsymbol{W}^\top \phi(\boldsymbol{x}) \in \mathbb{R}^{\mathcal{V}}$ using a linear transformation $\mathbf{w} \in \mathbb{R}^{d \times \mathcal{V}}$, and applies the Softmax function to obtain the predicted probability vector $\boldsymbol{p} = \mathrm{Softmax}(\boldsymbol{z})$. The standard cross-entropy loss is used during training, and it is defined as:

$$\mathcal{L}_{CE}(\boldsymbol{p}, y) = -\boldsymbol{e}_y^\top \log \boldsymbol{p}, \tag{9}$$

where $\boldsymbol{e}_y \in \mathbb{R}^{\mathcal{V}}$ is a one-hot vector with a 1 at the y-th position and zeros elsewhere. We operate under the assumption of a linearly parametrized softmax policy [12, 30, 33], in which the feature extractor ϕ is fixed, and only the parameters of the read-out layer $\boldsymbol{W}$ are updated. Using stochastic gradient descent with learning rate η, the update rule is:

$$\boldsymbol{W}^{t+1} = \boldsymbol{W}^t - \eta \nabla_{\boldsymbol{W}} \mathcal{L}_{CE} = \boldsymbol{W}^t - \eta \phi(\boldsymbol{x})(\boldsymbol{p}^t - \boldsymbol{e}_y)^\top, \tag{10}$$

where t denotes the t-th training step.

To analyze the dynamics of the training process, we convert the discrete update into a continuous-time differential equation [3]. Let $\boldsymbol{W}(t)$ denote the parameters at continuous time t, then:

$$\frac{d\boldsymbol{W}(t)}{dt} = -\eta \phi(\boldsymbol{x}) \left(\boldsymbol{p}(t) - \boldsymbol{e}_y\right)^\top. \tag{11}$$

Let $\boldsymbol{z}(t) = \boldsymbol{W}(t)^\top \phi(\boldsymbol{x})$, then $\boldsymbol{p}(t) = \mathrm{Softmax}(\boldsymbol{z}(t))$. Differentiating $\boldsymbol{p}(t)$ with respect to time yields:

$$\frac{d\boldsymbol{p}(t)}{dt} = \frac{d\,\mathrm{Softmax}(\boldsymbol{z}(t))}{d\boldsymbol{z}(t)} \cdot \frac{d\boldsymbol{z}(t)}{dt}. \tag{12}$$

Since $\phi(\boldsymbol{x})$ is constant, we have:

$$\frac{d\boldsymbol{z}(t)}{dt} = \left(\frac{d\boldsymbol{W}(t)}{dt}\right)^\top \phi(\boldsymbol{x}) = -\eta \left[\phi(\boldsymbol{x})^\top \phi(\boldsymbol{x})\right] \left(\boldsymbol{p}(t) - \boldsymbol{e}_y\right). \tag{13}$$

Let $\beta = \eta \|\phi(\boldsymbol{x})\|^2$ for convenience. On the other hand, the Jacobian matrix of the Softmax function is:

$$\frac{d\boldsymbol{p}}{d\mathbf{z}} = \mathrm{diag}(\boldsymbol{p}) - \boldsymbol{p}\boldsymbol{p}^\top. \tag{14}$$

Substituting these results gives the time derivative of $\boldsymbol{p}(t)$:

$$\frac{d\boldsymbol{p}(t)}{dt} = -\beta \left[\mathrm{diag}(\boldsymbol{p}(t)) - \boldsymbol{p}(t)\boldsymbol{p}(t)^\top\right] \left(\boldsymbol{p}(t) - \mathbf{e}_y\right). \tag{15}$$

This is a nonlinear vector differential equation describing the evolution of the prediction probabilities $\boldsymbol{p}(t)$ under gradient descent training. To extract the dynamics of each component $\boldsymbol{p}_k(t)$, we expand the above expression as:

$$\frac{d\boldsymbol{p}_k(t)}{dt} = -\beta \boldsymbol{p}_k(t) \left[(\boldsymbol{p}_k(t) - \delta_{ky}) - \sum_{j=1}^{V} \boldsymbol{p}_j(t)(\boldsymbol{p}_j(t) - \delta_{jy}) \right], \tag{16}$$

where δ_{ky} is the Kronecker delta. When $k = y$, $\delta_{ky} = 1$; otherwise, it is 0.

To numerically solve this continuous-time system, we apply the Euler method. Let the time step be Δt, and define $\boldsymbol{p}^{(n)} := \boldsymbol{p}(t_n)$ at discrete time $t_n = n\Delta t$. The Euler update rule is:

$$\boldsymbol{p}^{(n+1)} = \boldsymbol{p}^{(n)} - \Delta t \cdot \beta \left[\mathrm{diag}(\boldsymbol{p}^{(n)}) - \boldsymbol{p}^{(n)} \boldsymbol{p}^{(n)\top} \right] (\boldsymbol{p}^{(n)} - \boldsymbol{e}_y), \tag{17}$$

and for the k-th component:

$$\boldsymbol{p}_k^{(n+1)} = \boldsymbol{p}_k^{(n)} - \Delta t \cdot \beta \cdot \boldsymbol{p}_k^{(n)} \left[(\boldsymbol{p}_k^{(n)} - \delta_{ky}) - \sum_{j=1}^{\mathbf{V}} \boldsymbol{p}_j^{(n)} (\boldsymbol{p}_j^{(n)} - \delta_{jy}) \right]. \tag{18}$$

We analyze the relative dynamics of the hallucination component during training. Let $\boldsymbol{p}^t \in \mathbb{R}^{\mathcal{V}}$ denote the predicted probability vector at training step t. Let y denote the ground-truth label and define the hallucination component as the non-ground-truth class with the highest probability:

$$\boldsymbol{p}_h^t := \max_{k \neq y} \boldsymbol{p}_k^t. \tag{19}$$

Let c denote an arbitrary non-hallucination and non-target class, i.e., $c \notin \{h, y\}$. We show that $\boldsymbol{p}_h^{t+1} > \boldsymbol{p}_c^{t+1}$ always holds during training under the continuous-time Euler approximation of gradient descent. As derived from Equation (18), and omitting the superscript (n) for clarity, we have:

$$\boldsymbol{p}_h^{(n+1)} = \boldsymbol{p}_h - \Delta t \cdot \beta \cdot \boldsymbol{p}_h \left[\boldsymbol{p}_h - \sum_{j=1}^{\mathcal{V}} \boldsymbol{p}_j (\boldsymbol{p}_j - \delta_{jy}) \right]. \tag{20}$$

Define the auxiliary quantity:

$$f := \sum_{j=1}^{\mathcal{V}} \boldsymbol{p}_j (\boldsymbol{p}_j - \delta_{jy}) = \|\boldsymbol{p}\|^2 - \boldsymbol{p}_y. \tag{21}$$

Let $s := \boldsymbol{p}_h + \boldsymbol{p}_y$ be the total probability mass on the hallucination and target components, and define the residual mass:

$$R := 1 - s = \sum_{k \notin \{h,y\}} \boldsymbol{p}_k. \tag{22}$$

By the definition of $\boldsymbol{p}_h = \max_{k \neq y} \boldsymbol{p}_k$, we have for any $k \notin \{h, y\}$:

$$\sum_{k \notin \{h,y\}} \boldsymbol{p}_k^2 \leq \boldsymbol{p}_h \sum_{k \notin \{h,y\}} \boldsymbol{p}_k = \boldsymbol{p}_h R. \tag{23}$$

Thus, we can bound f from below as follows:

$$\begin{aligned} f &= \boldsymbol{p}_h + \boldsymbol{p}_y - \left(\boldsymbol{p}_h^2 + \boldsymbol{p}_y^2 + \sum_{k \notin \{h,y\}} \boldsymbol{p}_k^2 \right) \\ &\geq \boldsymbol{p}_h + \boldsymbol{p}_y - \left(\boldsymbol{p}_h^2 + \boldsymbol{p}_y^2 + \boldsymbol{p}_h R \right) \\ &= \boldsymbol{p}_h + \boldsymbol{p}_y - \boldsymbol{p}_h^2 - \boldsymbol{p}_y^2 - \boldsymbol{p}_h(1 - \boldsymbol{p}_h - \boldsymbol{p}_y) \\ &= \boldsymbol{p}_y(1 - \boldsymbol{p}_y + \boldsymbol{p}_h) \geq 0. \end{aligned} \tag{24}$$

Therefore, the hallucination probability satisfies

$$\boldsymbol{p}_h \geq \|\boldsymbol{p}\|^2 - \boldsymbol{p}_y. \tag{25}$$

Next, we consider the update of a non-hallucination component p_c for $c \notin \{h, y\}$. Its update is given by:

$$\boldsymbol{p}_c^{(n+1)} = \boldsymbol{p}_c - \Delta t \cdot \beta \cdot \boldsymbol{p}_c \left[\boldsymbol{p}_c - \left(\|\boldsymbol{p}\|^2 - \boldsymbol{p}_y \right) \right]. \tag{26}$$

We now examine the difference between the updated hallucination and non-hallucination components:

$$\boldsymbol{p}_h^{(n+1)} - \boldsymbol{p}_c^{(n+1)} = (\boldsymbol{p}_h - \boldsymbol{p}_c) - \beta \left[\boldsymbol{p}_h(\boldsymbol{p}_h - f) - \boldsymbol{p}_c(\boldsymbol{p}_c - f) \right], \tag{27}$$

where $f = \|\boldsymbol{p}\|^2 - \boldsymbol{p}_y$.

Define the auxiliary function $g(x) := x(x - f)$, a quadratic function in x. Note that since $\boldsymbol{p}_h \geq f$, we have $g(\boldsymbol{p}_h) \geq 0$. Furthermore:

- If $\boldsymbol{p}_c \geq f$, then g is increasing on $[f, 1]$, and $\boldsymbol{p}_h \geq \boldsymbol{p}_c$ implies $g(\boldsymbol{p}_h) \geq g(\boldsymbol{p}_c)$;
- If $\boldsymbol{p}_c < f$, then $\boldsymbol{g}(p_c) < 0 \leq g(\boldsymbol{p}_h)$.

In both cases, we conclude that

$$g(\boldsymbol{p}_h) \geq g(\boldsymbol{p}_c), \tag{28}$$

which implies

$$\boldsymbol{p}_h^{(n+1)} - \boldsymbol{p}_c^{(n+1)} \geq \boldsymbol{p}_h^{(n)} - \boldsymbol{p}_c^{(n)} \geq 0. \tag{29}$$

This indicates that function $d(t) = \boldsymbol{p}_h^t - \boldsymbol{p}_c^t$ is non-decreasing at any training step t. Moreover, since $d(t) \geq 0$, it follows that $d(t + 1) \geq 0$.

□

C. Experiment Setup

In this section, we present the complete experimental configuration, including implementation details and parameter specifications.

C.1. Preparation of Classifier Training Data

We first construct a labeled dataset of model-generated samples with hallucination annotations for training the classifier. By incorporating ground truth annotations as auxiliary information, we simplify the classification task, enabling the model to make accurate judgments even when relying solely on the textual modality.

For classifier training, we extract data from the POVID dataset [54], using prompts as questions and chosen responses as ground truth answers, while using LLaVA-1.5-7B's outputs as model responses. For cost efficiency, we employ the pure text model DeepSeek-V3 for hallucination annotation. The detailed system prompt for annotation are listed as follows.

Please act as an impartial and objective judge and evaluate the quality of the response provided by a Large Multimodal Model (LMM) to the user question. Your evaluation should be mainly based on whether the response is informative, and whether the response contains any hallucination. Hallucination, in this context, refers to a situation where the LMM generates a response that includes information not present or implied in the image or previous conversation. A hallucination could be a false claim about an object, action, emotion, or any other detail that is not grounded in the image.

For clarity, consider these examples:

Example 1:
- Question: How many cats can you see in this image?
- Standard Human-Generated Answer: There are two cats visible in the image.
- LMM Response to Evaluate: There is one cat in the image.
- Analysis: The LMM's response is false because it claims there is only one cat in the image, when in fact there are two.
- Hallucination: Yes.

Example 2:
- Question: What can you see from this photo?
- Standard Human-Generated Answer: In this photo we can see branches. We can see a panda on a branch. The background is blurry.
- LMM Response to Evaluate: In the photo, I can see a large black and white panda sitting on a tree branch. The image captures the panda's unique appearance and its natural habitat, showcasing the beauty and rarity of these animals.
- Analysis: The LMM's response is not false because it does not make any false claims about the image contents. For example, it does not claim that the panda is eating bamboo.
- Hallucination: No.

Example 3:
- Question: Describe this image in details.
- Standard Human-Generated Answer: In this picture I can see a dog laying under a yellow umbrella on a beach, and there is a rope on the right side. In the background I can see few persons talking or lying. At the top of the image, I can see the blue and clear sky.
- LMM Response to Evaluate: The image features a brown dog laying on the sand at the beach, underneath a yellow umbrella. The dog is positioned near the water, and there are people in the background. The beach is sandy, and the sun is shining, creating a pleasant atmosphere. The dog is enjoying the shade provided by the umbrella, which is a common way to protect pets from the sun's harmful rays. The presence of people in the background suggests that the beach is a popular spot for leisure activities, such as sunbathing, swimming, and relaxing.
- Analysis: The LMM's response is false because it claims that the dog is positioned near the water, but there is no water in the image.
- Hallucination: Yes.

Example 4:
- Question: Where is this photo taken?
- Standard Human-Generated Answer: From the slogan "Welcome to Houston, Texas", we can infer that this photo is taken at an airport in Houston, Texas.
- LMM Response to Evaluate: This photo is taken at an airport, specifically at the entrance of a terminal. However, the specific location of this airport cannot be determined from the image.
- Analysis: Although the LMM's response does not explicitly mention the location "Houston, Texas" and does not perfectly answer the question, it makes no false claims about the image contents. Therefore, this response is not considered hallucination. It is OK if the LMM fails to directly answer the question, expresses uncertainty, or cannot include all the objects or details in the human-generated answer.
- Hallucination: No.

Example 5:
- Question: What environmental concerns are involved in this photo?
- Standard Human-Generated Answer: This photo shows a smokestack emitting smoke. It may cause severe air pollution which threatens human health.
- LMM Response to Evaluate: In the photo, there are several environmental concerns related to the smokestack emitting smoke. The smoke from the smokestack is a byproduct of industrial processes, which can contribute to air pollution, climate change, and negative impacts on human health. The smoke contains harmful pollutants, such as particulate matter, sulfur dioxide, nitrogen oxides, and carbon monoxide, which can lead to respiratory problems, heart disease, and other health issues. Additionally, the smoke contributes to the greenhouse effect, which can lead to global warming and climate change.

```
The smokestack's emissions also affect the environment, as they can harm wildlife, vegetation, and
ecosystems. Therefore, it is essential to address these environmental concerns by implementing measures
 to reduce emissions and promote sustainable practices in industrial processes.
- Analysis: Although the LMM's response is significantly longer than the standard human-generated
answer, it does not contain any false claims about the image contents. Instead, it provides additional
general information about the environmental concerns, which can be inferred from the smoke emission.
Such detailed analysis or reasoning should be considered as a positive aspect, as long as it contains
no false claims.
- Hallucination: No.

With these examples in mind, please help me evaluate whether the response by the LMM is informative,
and whether hallucination exists in it, based on the comparison between the LMM's response and the
factual information provided in the image contents, question, and the standard human-generated answer
below.

Please note that the standard human-generated answer may only contain factual information but may not
give a detailed analysis. Also, the standard human-generated answer may not be completely comprehensive
 in describing all the objects and their attributes, so please be a bit more cautious during
evalutation. LMM's detailed analysis or reasoning should be encouraged.

To evaluate the LMM responses, you must rate the response by choosing from the following options:
- Rating: 6, very informative with good analysis or reasoning, no hallucination
- Rating: 5, very informative, no hallucination
- Rating: 4, somewhat informative, no hallucination
- Rating: 3, not informative, no hallucination
- Rating: 2, very informative, with hallucination
- Rating: 1, somewhat informative, with hallucination
- Rating: 0, not informative, with hallucination

Just answer a number in range [0, 6], nothing else.
```

Listing 1. System prompt template for hallucination annotation with DeepSeek-V3

To transform image content into text while controlling information loss for DeepSeek-V3 annotation, we extract key objects through COCO labels as image content, since POVID images originate from COCO 2014 [18]. Apart from the system prompt, our input to DeepSeek-V3 includes the following content:

$$
\begin{aligned}
\textit{Input} = \#\#\# \textbf{ Image Contents } & \backslash n \ \{image\_content\} \ \backslash n \backslash n \\
\#\#\# \textbf{ Question } & \backslash n \ \{question\} \ \backslash n \backslash n \\
\#\#\# \textbf{ Standard Human-Generated Answer } & \backslash n \ \{gt\_answer\} \ \backslash n \backslash n \\
\#\#\# \textbf{ LMM Response to Evaluate } & \backslash n \ \{model\_answer\}
\end{aligned}
$$

This process generates 8.4K binary classification training samples, with our trained classifier achieving 90% consistency with DeepSeek-V3 judgments on the held-out validation set. **It is worth noting that although we obtained fine-grained score annotations during the labeling stage, we did not directly use these scores.** Instead, we mapped samples with scores from 0 to 2 as hallucinated, and those with scores from 3 to 6 as non-hallucinated. The purpose of fine-grained scoring during annotation was solely to ensure the interpretability and reliability of the labels.

C.2. Implementation Details

All models are trained using LoRA, with uniform settings of LoRA rank=128, LoRA alpha=256, and LoRA dropout=0.05. For multimodal models, we freeze the vision encoder and fine-tune only the intermediate projection layer and the subsequent language model. The optimizer is consistently set as the Adam optimizer with warmup, using default parameters ($\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 1e-6, weight\_decay = 0.0, warmup\_ratio = 0.05$), paired with a cosine learning rate schedule.

Table 4. Training hyperparameters of different stages.

Configuration	Classification	Iteration 1	Iteration 2
Global batch size	24	24	32
Peak learning rate	1e-4	1e-5	2e-6
Epochs	3	1	5
LoRA rank		128	
LoRA α		256	
LoRA dropout		0.05	
β_1		0.9	
β_2		0.999	
ϵ		1e-6	
Optimizer		AdamW	
Learning rate schedule		cosine decay	
Weight decay		0.0	
Warmup ratio		0.05	

Training for all 7B-sized models utilize DeepSpeed ZeRO-2, while training for the 13B models employed DeepSpeed ZeRO-3.

More specific hyperparameter settings are provided in Table 4. When fine-tuning Qwen2-VL-7B-Instruct as a hallucination classifier, we set the global batch size to 24 and the initial learning rate to 1e-4, training for a total of 3 epochs. For preference optimization, we perform two iterations of training. In the first iteration, we use off-policy data, with a global batch size of 24, an initial learning rate of 1e-5, and train for 1 epoch. We set the DPO coefficient $\beta = 0.5$, and incorporate the NLL loss into the objective as a regularization term, with a weight of 0.2. Adding the NLL loss helps the model better capture the detailed linguistic style of ground truth answers, encouraging the generation of longer responses. The larger β further strengthens the KL divergence constraint, contributing to training stability. In the second iteration, we optimize the objective defined in Equation (8), with a global batch size of 32, an initial learning rate of 2e-6, and a total of 5 training epochs. The DPO coefficient is set to $\beta = 0.1$ in this stage.

D. Additional Results

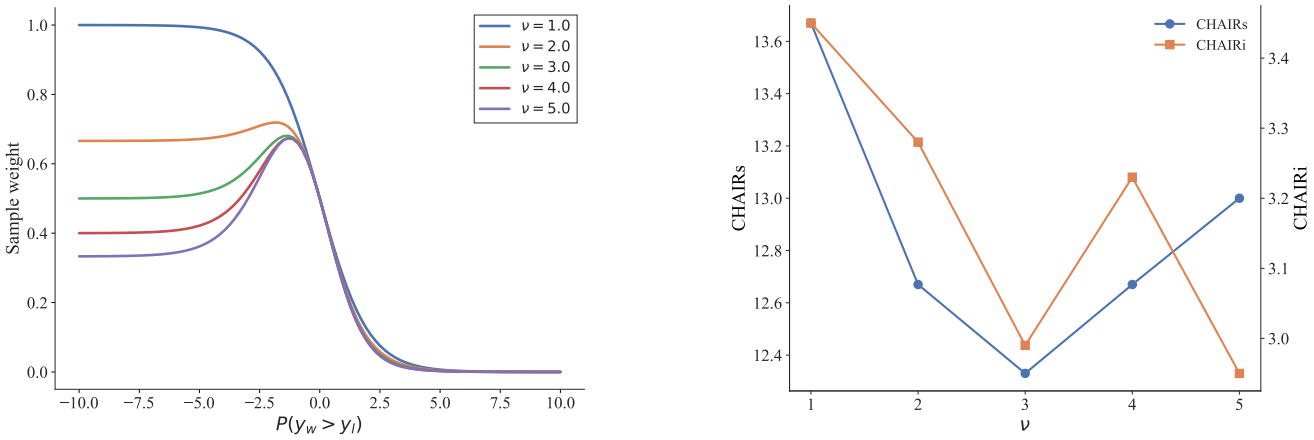
D.1. Ablation study on hyperparameters

We present additional experiment results examining key parameters in our framework. We mainly present results concerning the dynamic weight model parameters specified in Equation (7). First, we systematically vary ν to investigate its impact on the gradient difference term $(\nabla_{\theta,y_w} - \nabla_{\theta,y_l})$ in Equation (8). Notably, when $\nu = 1$, our loss function reduces to the standard DPO. Then we conduct ablation on different values of parameter ν on Object HalBench. The results are shown in Figure 5.

As shown in Figure 5a, the original DPO objective (with $\nu = 1.0$) assigns gradient weights in a sigmoid-like manner across different samples, consistent with the form of the $\sigma(\hat{r}(x, y_l) - \hat{r}(x, y_w))$ term. In contrast, our probability model places greater emphasis on samples near the decision boundary ($P(y_w \succ y_l) \approx 0$), while assigning lower weights to samples with large negative margins ($P(y_w \succ y_l) \ll 0$), as such instances are likely to be potentially noisy samples. Figure 5b demonstrates that the dynamic weighting achieves optimal performance at $\nu = 3$, leading us to adopt this value throughout our experiments.

D.2. General Benchmark Evaluations

To demonstrate that our method effectively reduces model hallucination without compromising general capabilities, we evaluate various hallucination mitigation approaches on both MMBench-EN and



(a) Gradient weight curve for different ν.

(b) Ablation on ν on Object HalBench.

Figure 5. Ablation study on parameter ν.

Table 5. Comparison of hallucination mitigation approaches on MMBench-EN and MMBench-CN

Model Size	Algorithm	Avg. Score ↑		Avg. Ranking ↓
		MMBench-EN	MMBench-CN	
7B	LLaVA-Instruct-1.5 [20, 21]	64.37	58.76	4.25
	LLaVA-RLHF [35]	51.40	39.52	7.0
	HA-DPO [53]	64.54	58.76	3.25
	POVID [54]	64.46	**60.82**	2.5
	RLAIF-V [49]	62.84	57.90	6.0
	OPA-DPO [47]	**65.73**	58.42	3.0
	Ours	65.48	59.36	**2.0**
13B	LLaVA-Instruct-1.5	67.77	63.75	**1.5**
	LLaVA-RLHF	60.10	52.66	4
	OPA-DPO	67.43	62.97	3
	Ours	**69.13**	63.49	**1.5**

MMBench-CN, as shown in the Table 5. The results indicate that our method outperforms baseline models not only in hallucination-related metrics but also in general visual question answering benchmarks. Compared to other algorithms, our method also achieves leading average rankings on both MMBench-EN and MMBench-CN.

E. Algorithm

We present our complete algorithm in Algorithm 1.

Algorithm 1: Robust Iterative Alignment

Input: Classifier $\mathcal{H}$, collected dataset $\mathcal{D} = \{(x, y^*)^i\}_{i=1}^N$, number of iterations T, number of generations K, batch size B, parameter ν for RK model, learning rate η.

Initialize: policy π_{θ_0}, preference data set $\mathcal{D}_{\text{pref}} = \emptyset$;

// Iterative DPO

for *t=1* **to** T **do**

  // Stage 1: Preference Data Construction

  for *i=1* **to** N **do**

    for *j=1* **to** K **do**

      generate response $y_j \sim \pi_{\theta_{t-1}}(\cdot \mid x_i)$ for x_i in $\mathcal{D}$; // generate K responses for each prompt

      Calculate the probability $P(h = 1 \mid x_i, y_j)$ through the hallucination classifier $\mathcal{H}(x_i, y_i^*, y_j)$.

    end

    Rank hallucination probablities $P(h = 1 \mid \cdot)$ for set $\{x_i, y_j\}_{j=1}^K$;

    Let the response with highest hallucination probability $P_{\max}$ be y_l;

    Let the response with lowest hallucination probability $P_{\min}$ be y_w;

    if $P_{\min} < 0.5$ *and* $P_{\max} \geq 0.5$ **then**

      $(x_i, y_w, y_l) \rightarrow \mathcal{D}_{\text{pref}}$.

    end

  end

  // Stage 2: Robust DPO Training

  for *each epoch* **do**

    Sample mini-batch $\mathcal{D}_m = \{(x, y_w, y_l)^m\}_{m=1}^B$ from $\mathcal{D}_{\text{pref}}$;

    Predict the probabilities $\pi_{\theta_t}(y_w \mid x)$ and $\pi_{\theta_t}(y_l \mid x)$ for (x, y_w, y_l) in $\mathcal{D}_m$ using the policy model;

    Predict the probabilities $\pi_{\theta_{t-1}}(y_w \mid x)$ and $\pi_{\theta_{t-1}}(y_l \mid x)$ for (x, y_w, y_l) in $\mathcal{D}_m$ using the reference model;

    Calculate the implicit reward $\hat{r}_w = \beta \log \frac{\pi_{\theta_t}(y_w|x)}{\pi_{\theta_{t-1}}(y_w|x)}$, $\hat{r}_l = \beta \log \frac{\pi_{\theta_t}(y_l|x)}{\pi_{\theta_{t-1}}(y_l|x)}$;

    Calculate pair-wise loss $\ell_{\text{pair}} = \log \sigma(\hat{r}_w - \hat{r}_l)$;

    Calculate sample weight $\gamma(x, y_w, y_l) = p(y_w \sim y_l \mid x) + \frac{2}{\nu+1}$; // Equation (7)

    $\theta \leftarrow \theta + \nabla_\theta \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}_m} \left[\text{sg}\left(\gamma(x, y_w, y_l)\right) \cdot \ell_{\text{pair}}\right]$; // Equation (8)

  end

  $\mathcal{D}_{\text{pref}} = \emptyset$.

end

Output: π_θ

Our algorithm implements an iterative model fine-tuning loop that progressively enhances output quality and reduces hallucination through three key phases per iteration. First, the generation phase produces multiple responses per prompt using temperature-controlled sampling to ensure diversity. The subsequent filtering phase applies our adaptive hallucination classifier to exclude hallucinated responses from preference training data. Following each iteration's data collection, we employ a robust reweighting mechanism that dynamically balances reward margin significance to prioritize uncertain boundary samples. This robust reweighting mechanism ensures stable fine-tuning against potential classifier annotation noise.

F. Future Works

In this work, we first provide a theoretical analysis of why on-policy data offers intrinsic advantages over off-policy data in hallucination mitigation for multimodal large language models. Building on this insight, we propose a tailored iterative DPO training framework. Both qualitative and quantitative experiments demonstrate that our approach substantially reduces hallucination rates.

Nevertheless, several issues remain open for further discussion. Due to computational constraints, we were unable to conduct broader experiments across diverse model architectures and scales to further validate the effectiveness of our method. In addition, the robustness of the hallucination classifier across data from different sources, as well as the influence of classifier performance on the overall pipeline, warrants deeper investigation. Since developing a robust hallucination classifier is not the primary focus of this paper, we leave this exploration into future work and we believe that incorporating a more powerful classifier would further enhance the performance of our method.