CLIP-DPO: Vision-Language Models as a Source of Preference for Fixing Hallucinations in LVLMs

Yassine Ouali[1] Adrian Bulat[1,2] Brais Martinez[1] Georgios Tzimiropoulos[1,3]

[1]Samsung AI Center Cambridge, UK [2] Technical University of Iași, Romania
[3] Queen Mary University of London, UK

Abstract. Despite recent successes, LVLMs or Large Vision Language Models are prone to hallucinating details like objects and their properties or relations, limiting their real-world deployment. To address this and improve their robustness, we present `CLIP-DPO`, a preference optimization method that leverages contrastively pre-trained Vision-Language (VL) embedding models, such as CLIP, for DPO-based optimization of LVLMs. Unlike prior works tackling LVLM hallucinations, our method does not rely on paid-for APIs, and does not require additional training data or the deployment of other external LVLMs. Instead, starting from the initial pool of supervised fine-tuning data, we generate a diverse set of predictions, which are ranked based on their CLIP image-text similarities, and then filtered using a robust rule-based approach to obtain a set of positive and negative pairs for DPO-based training. We applied `CLIP-DPO` fine-tuning to the MobileVLM-v2 family of models and to LlaVA-1.5, in all cases observing significant improvements in terms of hallucination reduction over baseline models. We also observe better performance for zero-shot classification, suggesting improved grounding capabilities, and verify that the original performance on standard LVLM benchmarks is overall preserved.

1 Introduction

A Large Vision Language Model (LVLM) is a Large Language Model (LLM) combined with an independently-trained vision encoder, typically taken from VL embedding models like CLIP [56] and often kept frozen. LVLMs have recently become the state-of-the-art for vision language understanding [5, 16, 42, 65, 81]. Unlike the prior generation of models [34, 70], which were typically tuned for one or a small number of tasks, LVLMs allow free-form dialog in natural language with an image in the loop. This direction largely draws inspiration from the recent success of ChatGPT [50] and respectively, ChatGPT-4V [51], which adapts LLMs to follow human instruction and preferences via various forms of reinforcement learning from human feedback (RFLH) and/or supervised fine-tuning (SFT) on high-quality multi-turn instruction (*i.e.* chat) data. In a similar fashion, the current generation of LVLMs is trained as part of a two-step process [42]. First, the CLIP vision encoder is aligned with the LLM by training a few adaptation layers only and, then, by fine-tuning the model using SFT on a multi-modal

instructional dataset. Despite their remarkable success, a major limitation hindering the wider adoption of LVLMs is the high rate of hallucinated details (*e.g.* non-existing objects) that these models exhibit in their generated output text.

Motivation. Following the footsteps of LLM training, a natural solution to combat hallucinations is given by RFLH. As reinforcement learning-based training is generally expensive, often requiring the training of a policy model, Rafailov et al. [57] proposed a simplified framework, coined Direct Preference optimization (DPO), that allows direct training of the model using a binary cross-entropy objective. The data for DPO is typically obtained either from human preferences or using LLMs to construct preference data automatically. Thanks to its efficiency, DPO has been quickly adopted for LVLMs too [35, 36, 83]. However, these approaches require multiple LVLMs for generation, collecting additional data, and using GPT-4/GPT-4V API for ranking. Moreover, the GPT-4 provided scores are discrete (hence, of reduced granularity) and are themselves prone to hallucinations, which can perpetuate and exacerbate the already high rate of factual errors and poor image grounding [37, 71]. Finally, GPT-4/V is behind a paywall, and using it for generation/ranking is neither scalable nor cost-efficient.

Main idea. To fix hallucinations and address the high data construction and annotation cost exhibited by current methods, we propose a new DPO variant, called `CLIP-DPO`, that uses a pre-trained CLIP model [56] to rank the LVLM's self-generated captions to construct positive-negative pairs for DPO. Since the CLIP model was trained in a contrastive manner to measure the alignment between image-text pairs, it is naturally suitable for determining the quality of a given output from an LVLM, grounding it to the correct object or attribute. The dataset over which we operate is constructed by running the original pre-trained target LVLM on its own output obtained using prompting, removing both the need for (i) additional external data and (ii) ensembles of external LVLMs. The final data is filtered before training using robust rule-based filtering.

Main results. We applied `CLIP-DPO` fine-tuning on top of two state-of-the-art models: MobileVLM-v2 (3 models in total) and LlaVA-1.5 7B. We find that, in both cases, our approach is effective in reducing hallucinations, outperforming all baseline models (*i.e.* the models without `CLIP-DPO` fine-tuning) by a significant margin. Importantly, `CLIP-DPO` significantly outperforms our direct competitor HA-DPO [83], outperforms Qwen-VL [5] trained on significantly larger datasets (*i.e.* 1.4B samples for pre-training and 77M for multitask training; as opposed to `CLIP-DPO` training on just 0.7M samples). Finally, our model's enhanced object grounding capabilities are also illustrated for zero-shot image classification, all without degrading the original performance of the base LVLM model.

2 Related work

Large Visual Language Models (LVLMs). Following the unprecedented success of Large Language Models (LLMs) [7, 50, 68, 69, 80], several works have recently proposed to build multi-modal capabilities on top of them [10, 14, 16, 42].

LLaVA [42] and FROMAGe [31] directly pass the visual tokens produced by a pre-trained CLIP [56] vision encoder to an LLM, either fine-tuning the LLM or adapting it using LoRA [24]. Notably, LLaVA training includes a pre-training stage that aligns the CLIP features with the LLM input tokens using a simple projection layer (*i.e.* keeping the rest of the model frozen) using image captioning data. InstructBLIP [16] uses QFormer [33] to reduce the number of vision tokens before passing them to the LLM. As the quality and distribution of the training data plays a crucial role [84], a series of methods [9,10,55,76] introduced improved data construction pipelines. For example, ShareGPT4v [10] uses the API of GPT-4V [51] to first label, then train a model, and finally use it to re-annotate a new training set. Another line of work is improving efficiency by reducing the models' size (original LLaVA models have 7B and 13B parameters). LLaVA-Phi [87], MobileVLM [13] and its follow-up, MobileVLM-v2 [14], replace the LLaMA [68] and Vicuna [12] LLMs with the smaller 1.4B and 2.7B variants of MobileLLaMA and Phi [38]. Our work is orthogonal to the aforementioned methods and does not seek to improve the model's architecture. Instead, we propose `CLIP-DPO`, an improved training approach for LVLMs based on DPO.

Preference optimization. Instruction tuning can significantly improve LLMs' perceived output quality and usefulness by aligning their responses to a given task domain or human preferences. This is achieved either by direct fine-tuning on expert data [15, 47, 67] or via reinforcement learning [52, 64, 88]. The latter significantly simplifies the data collection process, but still requires complicated training algorithms based on REINFORCE [73] or PPO [61]. Recently, a much-simplified approach was proposed, Direct Preference optimization (DPO) [57], which bypasses the need to train a reward model and allows direct training using a cross-entropy loss. Multiple improved versions for LLMs have been proposed in the meantime [4, 75, 82]. Following this, a recent wave of works on combining DPO with LVLMs have been proposed [35, 36, 83]. Silkie [35] constructs a multi-modal instructional dataset automatically labeled by GPT-4V. Similarly, HA-DPO [83], aiming to reduce the rate of hallucinations, uses a GPT4 model to label and construct positive-negative pairs with and without hallucinations. The work of [36] follows a similar path by using a suite of LLMs and LVLMs (*i.e.* Gemini-Vision [66]) to generate and label the data. These methods are then primarily evaluated on the LLaVA benchmark and PoPE [72]. In contrast to the aforementioned works, the proposed `CLIP-DPO` (i) simplifies the pipeline, (ii) removes the need for paid APIs, (iii) removes the need for additional data, and (iv) removes the need for additional external LVLMs. Instead, we make use of a pre-trained CLIP model to rank the generated outputs from a small pool of efficient LVLMs, and show that the proposed `CLIP-DPO` training provides significant improvements for fixing LVLM hallucinations and, in general, for enhancing the discriminability and robustness of the model as demonstrated by image classification experiments.

LVLM hallucinations. Broadly speaking, in the context of LVLMs, we consider hallucinations to be incorrect or misleading generated text, contradicting the visual evidence provided by the input image. This is an undesirable characteristic

Original Caption: In this image, we see a black dog wearing an orange vest standing on the grass. The dog's head is down and it appears to be in a backyard.

- **Existence Hallucination:** In this image, we see a black dog wearing an orange vest standing on the grass, with a small orange ball near his feet. The dog's head is down and it appears to be in a backyard.
- **Attributes Hallucination:** In this image, we see a dark brown dog wearing an orange vest standing on the grass. The dog's head is down and it appears to be in a backyard.
- **Relation Hallucination:** In this image, we see a black dog wearing an orange vest sitting on the grass. The dog's head is down and it appears to be in a backyard.

Fig. 1: An example of the injected hallucinations. Given an image and its caption, we prompt GPT-4 to generate 3 types of hallucination: existence, attributes, and relation.

inherited from the pre-trained LLM used, and further exacerbated by the visual-language alignment process [5, 25, 29, 40]. Multiple solutions have been recently proposed by the community with varying degrees of success. The works of [25,77] attempt to address the data bias by constructing better-grounded annotated image-text pairs. The works of [5, 11] scale the resolution of the image encoder, as this was observed to reduce the amount of hallucination, but at the cost of a high increase in the computational cost. The works of [27,78] improve the vision encoder by adding extra informational paths. The closest work related to ours is that of [83], which, with the help of GPT-4V, constructs negative-positive pairs for DPO fine-tuning. Different from [83], we use a pre-trained CLIP model to perform the ranking, and no additional data or external LVLMs are required. For evaluation purposes, we use AMBER [71], the most comprehensive benchmark for hallucinations to date, encompassing both a generative and discriminative evaluation component. Importantly, unlike all prior benchmarks [22, 30, 37, 40, 65], AMBER is a high-quality dataset fully annotated by humans for both the generative and discriminative tasks.

3 Preliminaries

3.1 The Effectiveness of CLIP in Reducing Hallucinations

First, we probe the effectiveness of CLIP in accurately ranking correct captions over those with hallucinated content. This assessment will indicate the efficacy of CLIP-style models in reducing hallucinations with DPO-based training. To this end, initially, we select 1K captions from the Detailed Caption [1] dataset, chosen for its high-quality labels. Then, we instruct GPT-4 [2] to create three hallucinated captions for each image, corresponding to the following types of hallucination: (i) *existence*: new elements or objects are added to the caption that were not mentioned originally, (ii) *attribute*: the attributes, characteristics, or features of the original caption's elements are altered, and (iii) *relationship*: the spatial, contextual, or interactive relationships of the original caption's elements are altered. See Fig. 1 for an example of the injected hallucinations.

Next, we use LLaVA-1.5 7B [41] to compute the likelihood of all captions, including the original and the hallucinated captions. We then retain only the hallucinated captions for which the model predicts a higher likelihood than the

original, yielding 61 for existence, 76 for attribute, and 119 for relationship hallucinations out of the initial 1K captions. Finally, we compute the CLIP image-text scores for these likely yet incorrect captions. As illustrated in Fig.2, CLIP manages to accurately rank the original caption over the hallucinated ones in at least 59% of cases, reaching up to 83% for the existence type of hallucinations. These findings underscore the potential of using VL embedding models to provide a reliable training signal for hallucination reduction in LVLMs.

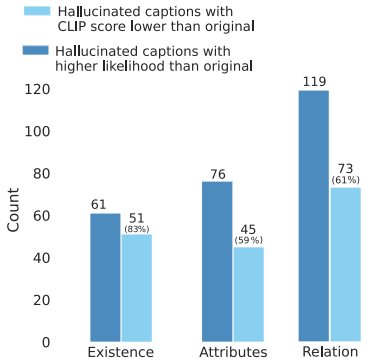


Fig. 2: In dark blue, we show the number of hallucinated captions per type that LLaVA-1.5 7B assigns a higher likelihood than the original out of 1K captions. Light blue shows the portion of samples corrected by CLIP.

3.2 Direct Preference Optimization

The standard pipeline for LLM and LVLM alignment consists of three steps: generation, annotation, and optimization. First, a set of N prompts $\{x_i\}_{i=1}^{N}$, each one consisting of an image and a text component in the case of LVLMs, are used to generate a set of response pairs y_i^1 and y_i^2 obtained from a pool of pre-trained LVLMs for each prompt x_i. Then, either human annotators or another set of LLMs (*i.e.* AI annotators) are used to rank the responses, resulting in a preferred y_i^+ and a less preferred response y_i^- for each prompt x_i, and thus a final preference dataset $\mathcal{D} = \{(x_i, y_i^+, y_i^-)\}_{i=1}^{N}$. Finally, Direct Preference Optimization [57] (DPO) can be applied to update the target policy π_θ parameterized by θ directly using the preference dataset $\mathcal{D}$. Specifically, the DPO optimization objective is defined as follows:

$$\max_{\pi_\theta} \mathbb{E}_{(x_i, y_i^+, y_i^-)\sim\mathcal{D}} \log \sigma \left(\beta \log \frac{\pi_\theta(y_i^+ \mid x_i)}{\pi_{\text{ref}}(y_i^+ \mid x_i)} - \beta \log \frac{\pi_\theta(y_i^- \mid x_i)}{\pi_{\text{ref}}(y_i^- \mid x_i)} \right), \quad (1)$$

with π_θ as the policy to be learned, π_{ref} as the reference SFT policy, and β as a hyperparameter to control the Kullback-Leibler (KL) divergence between the learned π_θ and reference π_{ref} policies [57]. The main benefit of DPO-based optimization is the direct alignment of the LVLM towards the preferences implicit in the preference data $\mathcal{D}$.

In contrast to previous DPO-based LVLM preference optimization works [35, 36, 83], next, we will introduce the proposed `CLIP-DPO`, which (i) simplifies the generation process by limiting the pool of LVLMs used for generation to small and efficient models, (ii) removes the need for external LLMs and LVLMs annotators accessed via paid APIs (*e.g.* GPT-3.5, GPT-4 or GPT-4V) by using CLIP as the ranking model, and (iii) removes the need for additional data by reusing the same data used during the SFT step.

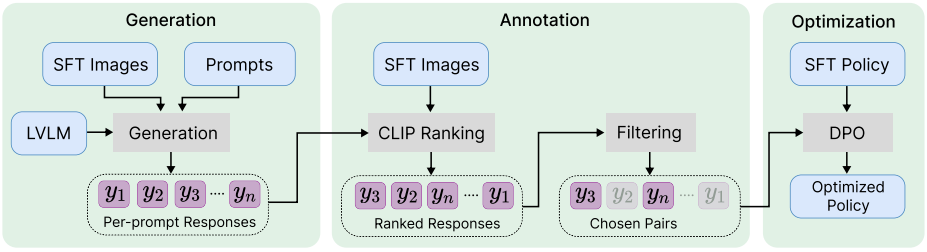


Fig. 3: `CLIP-DPO`. Starting from the initial SFT data pool and a set of prompts, an LVLM generates captions. These captions are first ranked using a CLIP model, then filtered to identify the most suitable positive and negative pairs for DPO-based optimization.

4 Method

In this section, we introduce our DPO variant for LVLMs preference optimization, named `CLIP-DPO`. As illustrated in Fig. 3, `CLIP-DPO` follows a three-step process: generation, annotation, and optimization, similar to the standard DPO-based training pipeline. While the final DPO-based optimization step remains unchanged, `CLIP-DPO` modifies the first two steps.

Firstly, we streamline the generation step by utilizing the same data pool (*i.e.* images only) as the SFT stage and generate data using either the model itself or a small pool of efficient LVLMs. Secondly, we introduce an annotation step tailored for CLIP-style embedding models, where we rank captions based on their image-text similarities. Then, given that the resulting captions and pairs may vary in quality, we implement a rule-based filtering strategy to remove unsuitable candidates. Finally, with the constructed preference data, the model undergoes DPO-based optimization using Eq. 1, as shown in Fig. 3.

As such, by leveraging VL embedding models for data annotation, we tap into the extensive information captured by these contrastively-pretrained models, which have been exposed to hundreds of millions of unique image-text pairs during training, unlike existing open-sourced LVLMs. Additionally, unlike the discrete GPT-4V ranking used in prior work, CLIP scores are continuous, enabling finer-grained comparisons. This allows us to assess the difficulty of a given pair based on the margin between the score values.

Next, we will elaborate on each module of our pipeline. Sec. 4.1 details the data generation module, and Sec. 4.2 details the data annotation process.

4.1 Data Generation

The first step of our pipeline consists of generating a set of per-image captions that will be later ranked by a pre-trained CLIP model, along with a subsequent filtering approach to select a set of positive and negative pairs, which are then used for DPO-based training. To this end, we start by selecting the pool of LVLM annotators and the data to be annotated. For the annotators and to reduce the

cost of the generation step, we select MobileVLM-v2 [14] family of models, given their efficiency and performance. As for the data, and to avoid introducing any new sources that might bias the results of `CLIP-DPO`, and to further reduce the method's cost, we opt for the initial pool of data used during the SFT stage of MobileVLM-v2 [14] models, see Table 1 for details. Then, we conduct our data generation pipeline that consists of two steps, the generation of *generic captions* and *per-image questions and answers*.

Dataset	Dataset Size
COCO	118K
GQA	72K
LLaVA-1.0 Pretraining	595K
LLaVA-1.5 Pretraining	558K
OCR VQA	80K
SAM	570K
SBU	845K
TextVQA	22K
VG	86K
Web-Celebrity	495
Web-Landmark	500
Wikiart	500
Total	2.9M

Table 1: Data pool used for data generation.

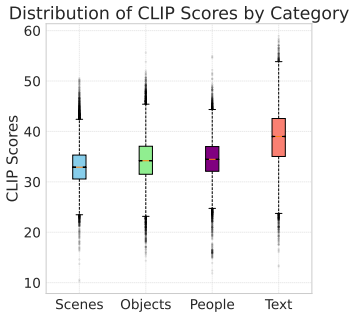


Fig. 4: The distribution of CLIP image-text scores per category.

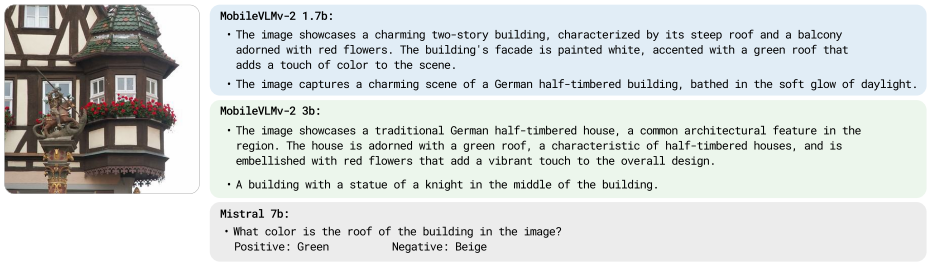


Fig. 5: Examples of generated generic captions produced by MobileVLM-v2 1.7B and MobileVLM-v2 3B. For MobileVLM-v2 3B's generated captions, we also show the produced question and positive and negative answers generated using Mistral 7B.

Generation of Generic Captions. We start by generating a set of 5 descriptive captions per image. For each of the MobileVLM-v2 models, we prompt with 5 different prompts (*e.g.* "identify the setting and note any characters or objects, focusing on visible details.") to increase the diversity of the generated captions. While these captions can be used for CLIP ranking and DPO-based training, they are still produced by generic prompts that are not image-specific. Next, we perform the second step of our data generation pipeline to obtain a set of per-image questions and answers.

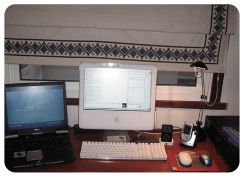
Generation of Per-Image Questions and Answers: To obtain a set of questions per image, we leverage an LLM, *i.e.* a variant of Mistral 7B Instruct-

Question 1: What is the central focus of the image?
- Question CLIP Score: 15.0
- Question Filter: Rejected

Question 2: What type of market is depicted in the image?
- Question CLIP Score: 26.7
- Question Filter: Kept

Positive: A fruit and vegetable market.
- Caption: type of market is depicted in the image a fruit and vegetable market.
- Positive CLIP Score: 29.1
- Question Filter: Kept

Fig. 6: Question Filtering. The sample is rejected if the CLIP image-question score is low for a given sample. If the question is relevant and not generic, it is kept. Next, we evaluate the quality of the answer by first parsing the question into a caption, appending the positive answer to it, and computing its CLIP score. If it is a high score, the sample is kept for DPO-based training.



Prompt: Provide a brief description of the given image.
Positive: An Apple computer with a laptop computer sitting on a desk.
Negative: A laptop and a desktop computer are sitting on a desk.

Prompt: Provide a depiction of the picture.
Positive: A man in orange shirt kicking a soccer ball.
Negative: A group of people kicking a soccer ball.

Prompt: What is the name of the paint brand featured on the bucket?
Positive: Homer's All Purpose Bucket.
Negative: Home Depot All-Purpose Bucket.

Fig. 7: Examples of the final positive and negative pairs obtained post-filtering, to be used for DPO-based training.

v0.2 [28][1], and feed it with the generated captions and prompt it to generate 2 questions for each image, together with positive and negative answers. The LLM is asked to generate the positive ones based on the fed captions and to generate plausible but incorrect negatives given the image description/caption.

For an example of the generated captions, questions, and answers, see Fig. 5.

4.2 Data Annotation

Starting with the generated captions, the subsequent data annotation step involves ranking them based on their CLIP image-text similarities and filtering them to obtain a final set of high-quality positive-negative pairs for DPO-based training. Our filtering pipeline includes two stages: *global filtering*, where we eliminate low-quality images and captions, and *pair-filtering*, where we select the best positive and negative per-image pairs for DPO-based training.

CLIP Ranking. The initial step of data annotation is CLIP ranking. In this step, we use a pre-trained CLIP model to compute the cosine similarities of the captions with their associated images and then sort them from highest to lowest.

[1] https://huggingface.co/teknium/OpenHermes-2.5-Mistral7B

Global Filtering. Next, we aim to filter our data and only retain a balanced and higher-quality subset. To achieve this, we start by analyzing the types of images in our data pool (see Tab. 1). We define a set of four generic categories: images of text, people, objects, and scenes. Using CLIP and a set of 10 descriptions per category, we create four class prototypes. Each image is then assigned to one of these prototypes based on the highest cosine similarity. As illustrated in Fig. 4, we observe that CLIP scores for the text category are the highest, while scores for people, objects, and scenes are relatively similar. Consequently, before applying CLIP-based filtering, we down-sample the portion of text images. We then filter out all generic captions below a given CLIP score (*e.g.* < 28.0). Additionally, since CLIP is primarily trained on short text, we remove long captions to ensure more precise CLIP scores. For questions, we compute their CLIP scores and remove all questions with low CLIP scores (*e.g.* < 25.0), thereby eliminating generic questions (*e.g.* "what is the main object in the image?") that are already covered by the generated captions.

Pair Filtering. Finally, starting from the remaining high-quality image captions and image question-answers, the final step will be selecting a set of positive and negative pairs for DPO-based training. For the questions, since we already have a set of positive and negative responses generated by our LLM, we only need to filter out the low-quality pairs. To achieve this, we use simple regex matching rules to extract an image description from the question, append the positive answer to it to create a synthetic caption, and compute its CLIP image-text scores. We then reject examples where the scores are low. See Fig. 6 for an example of the questions filtering process. As for the captions, we consider all possible pairs where the CLIP score difference between two captions is larger than a given threshold (*e.g.* > 2.0) and where the length of the two captions is similar to avoid introducing false preferences. We then order them based on the CLIP score difference between the positive and negative captions and select the top-ranking pair per image.

After the data annotation step, the resulting DPO training data consists of 750K pairs, of which 50K are question-answer pairs and the rest are caption pairs. For examples of the final pairs, see Fig. 7.

5 Implementation details

Network architecture: In this work, we consider two LVLMs architectures: MobileVLM-v2 [14] and LLaVA-1.5 [42]. Both models follow the same overall structure: a pre-trained CLIP vision encoder and a pre-trained LLM. The visual tokens produced by the frozen vision encoder are projected using either a linear layer or a small projection module and passed as input to the LLM. LLaVA opts for a pre-trained Vicuna LLM [12] while MobileVLM-v2 uses MobileLLaMA [13], except for their 7B variant, which also uses a Vicuna model. Both use the same ViT-L-14 @ 336px CLIP visual encoder [56]. For efficiency purposes, MobileVLM-v2 uses a projection module that halves the number of

visual tokens. We consider the following model variants in our comparisons: MobileVLM-v2 (1.7B, 3B and 7B), and LLaVA-1.5 7B. For data annotation, we use ViT-H/14 DFN [19] as our pre-trained CLIP model.

Training details: For all of our experiments, unless specified otherwise, we start from the pre-trained MobileVLM-v2 and LLaVA-1.5 models. The models are then fine-tuned for 1 epoch using DPO-based optimization on the constructed `CLIP-DPO` dataset. We use the following hyperparameters for fine-tuning the models: AdamW optimizer [45] with a batch size of 256, a learning rate of $5e-7$, decreased to 0 using a cosine scheduler, a warm-up of 0.01, and a weight decay set to 0. Both during training and testing, the input images are cropped and resized to 336×336px. The training was performed on 8 A100 GPUs using Pytorch [54]. For the larger models, *e.g.* LLaVA-1.5 7B, to fit them in memory, we use the Zero-3 strategy [58,59] and decrease the batch size to 64.

6 Results

We first evaluate the effectiveness of `CLIP-DPO` in reducing hallucinations on the recently introduced AMBER [71] benchmark in Sec. 6.1. Moreover, we demonstrate its enhanced grounding capabilities by reporting zero-shot image classification in Sec. 6.2. Finally, in Sec. 6.4, we show that the proposed `CLIP-DPO` training does not compromise LVLM performance by reporting results on the LLaVA benchmark [87].

6.1 Evaluation on hallucinations

We start by evaluating `CLIP-DPO` in terms of LVLM hallucination reduction, which is our main objective. We use AMBER [71], a comprehensive, high-quality, and LLM-free multidimensional benchmark for LVLM hallucination evaluation, which can be used to evaluate both generative and discriminative tasks. As the results from Tab. 2 show, MobileVLM-v2 trained with `CLIP-DPO` improves upon MobileVLM-v2 baselines across all model sizes, and sometimes quite significantly, especially for the 1.7B and 7B models. It can also be seen that `CLIP-DPO` significantly improves when applied on top of LLaVA-1.5 7B. Importantly, `CLIP-DPO` significantly outperforms HA-DPO [83], the main competing approach, improving the AMBER score 3.2 vs 7.8 when using LLaVA-1.5. Finally, our LlaVA-1.5 7B+`CLIP-DPO` even outperforms Qwen-VL [5], which is trained on significantly larger datasets (1.4B image-text pairs for pre-training and 77M for multitask training while `CLIP-DPO` is fine-tuned on just 0.7M samples), and even matches the performance of GPT-4V without using any GPT-4V model outputs during training. Overall, these results on a high-quality state-of-the-art benchmark such as AMBER clearly demonstrate the effectiveness of our `CLIP-DPO` approach for reducing hallucinations.

6.2 Zero-shot image classification

A primary reason for the hallucinatory behavior of LVLMs is the weak alignment between their visual features and the input LLM tokens. A direct way to

Model	Generative Task				Discriminative Task				AMBER
	CHAIR$_{\downarrow}$	Cover$_{\uparrow}$	Hal$_{\downarrow}$	Cog$_{\downarrow}$	Acc.	P.	R.	F1	
mPLUG-Owl	21.6	50.1	76.1	11.5	40.1	92.8	10.5	18.9	48.7
LLaVA	11.5	51.0	48.8	5.5	42.7	74.1	21.0	32.7	60.6
MiniGPT-4	13.6	63.0	65.3	11.3	63.6	90.5	50.4	64.7	75.6
CogVLM	5.6	57.2	**23.6**	**1.3**	69.0	88.9	60.9	72.3	83.4
mPLUG-Owl2	10.6	52.0	39.9	4.5	75.6	**95.0**	66.9	78.5	84.0
InstructBLIP	8.8	52.2	38.2	4.4	76.5	84.5	79.0	81.7	86.5
Qwen-VL	5.5	49.4	**23.6**	1.9	81.2	90.8	79.7	84.9	89.7
GPT-4V	**4.6**	**67.1**	30.7	2.6	**83.4**	84.9	**90.1**	**87.4**	**91.4**
LLaVA-1.5 7B	7.8	**51.0**	36.4	4.2	72.0	**93.2**	62.4	74.7	83.5
+ HA-DPO	7.2	33.6	19.7	2.6	68.3	68.1	**98.4**	80.5	86.7
+ CLIP-DPO	**3.7**	47.8	**16.6**	**1.3**	**77.8**	84.4	81.5	**82.9**	**89.6**
MobileVLM-v2 1.7B	**3.8**	39.6	**8.9**	**0.5**	65.4	**92.6**	51.9	66.5	81.4
+ CLIP-DPO	4.2	38.9	10.8	0.5	71.2	88.7	64.8	74.9	**85.3**
MobileVLM-v2 3B	4.8	38.7	**11.1**	0.7	73.5	**92.1**	65.7	76.7	86.0
+ CLIP-DPO	**4.7**	**41.5**	13.1	**0.5**	**76.7**	91.0	**71.8**	**80.3**	**87.8**
MobileVLM-v2 7B	4.4	**38.9**	10.4	0.6	71.9	**95.0**	60.8	74.1	84.9
+ CLIP-DPO	**4.0**	38.0	**10.1**	**0.5**	**77.3**	93.4	**70.8**	**80.5**	**88.3**

Table 2: Hallucination evaluation results on AMBER for both generative (leftmost set of columns) and discriminative tasks (rightmost set of columns). Our approach offers consistent and large performance improvements across 4 different LVLM models, and also beats HA-DPO by a large margin.

Method	StanfordCars	OxfordPets	OxfordFlowers	Imagenet	Food-101	Eurosat	Caltech-101	UCF-101	SUN397	Avg
LLaVA-1.5	23.2	34.0	7.3	37.9	45.3	49.5	82.7	50.5	43.1	40.0
+ HA-DPO	23.2	33.4	7.1	36.9	45.1	49.4	82.7	50.3	42.7	39.7
+ CLIP-DPO	**30.1**	**44.8**	**16.3**	**42.2**	**53.9**	**52.1**	**84.8**	**53.1**	**48.8**	**47.4**
MobileVLM-v2 1.7B	17.1	15.2	14.5	32.6	31.2	49.3	77.9	45.5	39.3	33.9
+ CLIP-DPO	**19.2**	**32.6**	**23.8**	**41.3**	**47.4**	**48.9**	**80.8**	**50.2**	**49.3**	**43.7**
MobileVLM-v2 3B	14.8	23.5	9.1	35.7	38.8	53.2	81.8	48.6	42.0	36.7
+ CLIP-DPO	**28.6**	**40.9**	**19.5**	**44.3**	**52.0**	**56.4**	**85.5**	**52.3**	**50.1**	**47.7**
MobileVLM-v2 7B	27.1	32.5	11.5	37.7	45.0	43.1	81.9	49.0	46.6	41.6
+ CLIP-DPO	**32.5**	**51.3**	**35.8**	**50.1**	**62.6**	**59.3**	**88.3**	**56.2**	**54.8**	**54.5**

Table 3: Zero-shot image recognition results in terms of Top-1 accuracy (%). While HA-DPO shows very similar performance to the base model, our `CLIP-DPO` improves the base model by a very large margin in all cases.

evaluate this is through simple zero-shot image classification, which is the go-to benchmark for contrastively trained VL models like CLIP [56]. The typical setup follows a closed-set classification problem, where the names of all possible classes are known a priori and are encoded into class prototypes using CLIP. A given image is then assigned to the class with the highest CLIP image-text scores. Herein, we follow the same protocol with the notable difference that, as our goal is LVLM evaluation, we first prompt the model to generate a free-form image caption describing the main object in the image, then encode the caption using the CLIP text encoder and assign the image to the class with the highest text-text (*i.e.* caption-class) CLIP score. Here, we opted for a different family of VL embedding model, *i.e.* SigLIP [79], to avoid evaluating using CLIP models similar to those used during the data annotation step. Following [56,85], we evaluate our approach on a suite of 9 diverse datasets: UCF-101 [63], SUN397 [74],

Method	LLM	Res.	GQA	SQA^I	VQA^T	POPE	MME^P	MMB^{dev}	Avg.
BLIP-2 [33]	Vicuna-13B	224	41.0	61.0	42.5	85.3	1293.8	–	-
InstructBLIP [16]	Vicuna-13B	224	49.5	63.1	50.7	78.9	1212.8	–	-
Shikra [9]	Vicuna-13B	224	–	–	–	–	–	58.8	–
Openflamingo [3]	MPT 7B	336	–	–	33.6	–	–	4.6	–
Qwen-VL [5]	Qwen 7B	448	59.3	67.1	**63.8**	–	1487.6	38.2	-
mPLUG-Owl [76]	LLaMA 7B	224	–	–	–	–	967.3	49.4	–
MiniGPT-v2 [8]	LLaMA 7B	448	60.3	–	–	–	–	12.2	–
MiniGPT-4 [86]	Vicuna 7B	224	32.2	–	–	–	581.7	23.0	–
InstructBLIP [16]	Vicuna 7B	224	49.2	60.5	50.1	–	–	36.0	–
ShareGPT4V [10]	Vicuna 7B	336	**63.3**	68.4	60.4	**85.7**	**1567.4**	**68.8**	**70.8**
MoE-LLaVA-1.6B×4 [39]	StableLM-1.6B	336	60.4	62.6	47.8	84.3	1300.8*	59.4	63.3
MoE-LLaVA-2.7B×4 [39]	Phi-2.7B	336	61.1	**68.7**	50.2	85.0	1396.4*	65.5	66.7
LLaVA-1.5 [41]	Vicuna 7B	336	**62.0**	66.8	58.2	**85.9**	**1510.7**	64.3	68.8
+ HA-DPO	Vicuna 7B	336	61.9	**69.2**	**58.3**	84.3	1505.6	**64.9**	**69.0**
+ CLIP-DPO	Vicuna 7B	336	59.3	67.6	56.4	85.8	1468.7	**64.9**	67.9
MobileVLM 1.7B [13]	MobileLLaMA 1.4B	336	56.1	57.3	41.5	**84.5**	1196.2	53.2	58.7
MobileVLM-v2 1.7B [14]	MobileLLaMA 1.4B	336	58.3	66.7	**52.1**	84.3	**1302.8**	**57.7**	**64.2**
+ CLIP-DPO	MobileLLaMA 1.4B	336	**58.6**	**68.4**	47.8	84.3	1331.1	57.6	63.8
MobileVLM- 3B [13]	MobileLLaMA 2.7B	336	59.0	61.2	47.5	84.9	1288.9	59.6	62.8
MobileVLM-v2 3B [14]	MobileLLaMA 2.7B	336	**61.1**	70.0	**57.5**	84.7	**1440.5**	63.2	68.1
+ CLIP-DPO	MobileLLaMA 2.7B	336	60.9	**72.3**	57.3	**85.1**	1425.7	**63.8**	**68.4**
MobileVLM-v2 [14] 7B	Vicuna 7B	336	**62.6**	74.8	**62.3**	**85.3**	**1560.7**	69.2	72.1
+ CLIP-DPO 7B	Vicuna 7B	336	62.5	**77.3**	62.2	84.7	1539.1	**70.4**	**72.3**

Table 4: Results on several vision language tasks from the LlaVA benchmark. Our approach is combined with LVLMs of different sizes, showing consistent performance in all cases.

Stanford Cars [32], Oxford Pets [53], Oxford flowers [49], ImageNet [17], Food 101 [6], Eurosat [23] and Caltech-101 [20]. As the results from Tab. 3 show, all LVLMs fine-tuned with `CLIP-DPO` significantly outperform their corresponding baselines, showcasing the increased discriminative properties. Note again that these improvements are obtained without affecting the model's performance on other tasks and datasets.

6.3 Additional evaluations

Herein, we evaluate the impact of `CLIP-DPO` training on other vision language tasks and, more specifically, on the popular LLaVA-Bench (GQA [26], ScienceQA [46], TextVQA [62], MME [21], MMBench [44]). As Tab. 4 shows, overall, `CLIP-DPO` training does not compromise performance. We further note that accuracy improvements on LLaVA-Bench are heavily tied to the addition of extra training data [10] or architectural changes [13,14,39,41]. As we are not using any additional data & models nor making architectural changes, it is not surprising that the performance after `CLIP-DPO` training remains largely in line with that of the original baseline model.

6.4 Qualitative Results

In Figure 8, we show some qualitative examples comparing the predictions of LLaVA-1.5 7B, LLaVA-1.5 7B+HA-DPO, and LLaVA-1.5 7B+`CLIP-DPO`. Over-

Provide a short description of the image.

LLava-1.5-7b: A large silver airplane is parked inside a hangar. The plane is positioned on a runway, and there is a ladder leading up to the cockpit. The hangar appears to be a large building, possibly a factory or a maintenance facility.

+ HA-DPO: A large silver airplane is parked inside a hangar.

+ CLIP-DPO: A fighter jet is parked inside a hangar, with a ladder attached to it. The jet is numbered 157.

Provide a short description of the image.

LLava-1.5-7b: A green bus is parked in a parking lot, waiting for passengers.

+ HA-DPO: A green bus is parked in a parking lot, waiting for passengers.

+ CLIP-DPO: A green Go3 bus is parked at a bus stop.

Fig. 8: Qualitative examples comparing our method with LLaVA and HA-DPO.

Loss variant	AMBER Score	Cls. Avg.	LLaVA bench.
DPO ($\beta = 0.1$) [57]	85.3	43.7	63.8
DPO ($\beta = 0.3$) [57]	83.3	42.1	64.0
IPO [4]	83.6	40.5	64.5
KTO [18]	83.5	39.5	64.7
SLIC [43]	83.2	42.7	64.0
cDPO ($\beta = 0.1, LS = 0.1$ [43])	83.1	43.2	63.9
cDPO ($\beta = 0.3, LS = 0.1$ [43])	83.1	40.7	64.5

Table 5: Effect of DPO loss variant on `CLIP-DPO` training. Results are reported in terms of AMBER Score, Average Top-1 (%) accuracy across all 9 image classification datasets, and the average on LLaVA benchmark. MobileVLM-v2 1.7B was used.

all, with `CLIP-DPO`, the model is more grounded in the visual content, less prone to hallucinations, and more precise and fine-grained in its descriptions.

7 Ablation studies

Herein, we ablate the effect of the proposed components. In all experiments, we use a MobileVLM-v2 1.7B model.

7.1 Effect of DPO loss variant

Our work primarily follows the original DPO formulation proposed in [57]. Here we also consider the following recently proposed variations: KTO [18], ITO [4], SLIC [43] and cDPO [48]. As Tab. 5 shows, aggregated, all losses tend to perform similarly, with DPO marginally outperforming the others.

7.2 Effect of CLIP scorer

Herein, we seek to explore alternatives to ViT-H/14 DFN [19] CLIP model used in previous experiments, analyzing the impact of the scorer used on the overall performance of the model. We consider a diverse set of alternatives, covering

Scorer	AMBER Score	Cls. Avg.	LLaVA bench.
ViT-L/14 [56]	79.8	42.3	63.0
SigLIP-L/16 [79]	85.0	43.9	63.2
ViT-H/14 [60]	84.3	44.2	63.7
ViT-H/14 DFN [19]	85.3	43.7	63.8

Table 6: Effect of CLIP scorer on `CLIP-DPO` training. Results are reported in terms of AMBER Score, Average Top-1 (%) accuracy across all 9 image classification datasets, and the average on LLaVA benchmark. MobileVLM-v2 1.7B was used.

multiple exploratory paths: equally-sized models trained on different data (ViT-H/14 [19] trained on LAION-2B instead of DFN-5B); smaller models (ViT-L/14 [56]) and models trained using different pre-training losses (SigLIP-ViT-L/16 [79]). As the results from Tab. 6 show, our approach is generally robust to the exact scorer used, with the notable exception of the ViT-L/14 [56] model. Notice that this difference is primarily manifesting on the AMBER benchmark. Intuitively, this showcases the importance of a powerful scorer for reducing the amount of hallucinations.

Limitations and broader impact

As an LVLM-based approach, our method is subject to the same general consideration (*i.e.* potential data bias, susceptibility to hallucinations, etc.). Moreover, as the LVLMs are trained on relatively small datasets compared to LLMs or CLIP, gaps within their knowledge domains are possible. This is especially important as neural networks tend to be overconfident outside their seen input distribution. As with all models from this category, we strongly recommend checking the models and the data carefully before deploying them. Despite these general aspects, our approach is shown to significantly reduce the amount of hallucinated content and improve the model's discriminability, hence resulting in more robust and reliable models.

8 Conclusions

In this work, we proposed `CLIP-DPO`, a simple method that reduces hallucinations in LVLMs based on using a pre-trained CLIP model [56] to rank the LVLM's self-generated captions in order to construct positive-negative pairs for DPO fine-tuning. In contrast to previous works, our proposed `CLIP-DPO` removes the need of (i) paid-for APIs, (ii) additional data and (iii) additional external LVLMs. Before training, the data is filtered using a newly proposed robust rule-based approach. When applied on top of established LVLMs, it is shown that `CLIP-DPO` fine-tuning significantly reduces hallucinations, outperforming the baseline models by significant margins and even matching the performance of GPT-4V on the AMBER benchmark. We also observe superior zero-shot object recognition, suggesting improved object grounding capabilities.

References

1. Detailed caption dataset. https://huggingface.co/datasets/echo840/Detailed_Caption (2024)
2. Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al.: Gpt-4 technical report. arXiv preprint arXiv:2303.08774 (2023)
3. Awadalla, A., Gao, I., Gardner, J., Hessel, J., Hanafy, Y., Zhu, W., Marathe, K., Bitton, Y., Gadre, S., Sagawa, S., et al.: Openflamingo: An open-source framework for training large autoregressive vision-language models. arXiv preprint arXiv:2308.01390 (2023)
4. Azar, M.G., Rowland, M., Piot, B., Guo, D., Calandriello, D., Valko, M., Munos, R.: A general theoretical paradigm to understand learning from human preferences. arXiv preprint arXiv:2310.12036 (2023)
5. Bai, J., Bai, S., Yang, S., Wang, S., Tan, S., Wang, P., Lin, J., Zhou, C., Zhou, J.: Qwen-VL: A versatile vision-language model for understanding, localization, text reading, and beyond. arXiv preprint arXiv:2308.12966 (2023)
6. Bossard, L., Guillaumin, M., Van Gool, L.: Food-101–mining discriminative components with random forests. In: European Conference on Computer Vision (2014)
7. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. Advances on Neural Information Processing Systems (2020)
8. Chen, J., Zhu, D., Shen, X., Li, X., Liu, Z., Zhang, P., Krishnamoorthi, R., Chandra, V., Xiong, Y., Elhoseiny, M.: Minigpt-v2: large language model as a unified interface for vision-language multi-task learning. arXiv preprint arXiv:2310.09478 (2023)
9. Chen, K., Zhang, Z., Zeng, W., Zhang, R., Zhu, F., Zhao, R.: Shikra: Unleashing multimodal llm's referential dialogue magic. arXiv preprint arXiv:2306.15195 (2023)
10. Chen, L., Li, J., Dong, X., Zhang, P., He, C., Wang, J., Zhao, F., Lin, D.: Sharegpt4v: Improving large multi-modal models with better captions. arXiv preprint arXiv:2311.12793 (2023)
11. Chen, Z., Wu, J., Wang, W., Su, W., Chen, G., Xing, S., Muyan, Z., Zhang, Q., Zhu, X., Lu, L., et al.: Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. arXiv preprint arXiv:2312.14238 (2023)
12. Chiang, W.L., Li, Z., Lin, Z., Sheng, Y., Wu, Z., Zhang, H., Zheng, L., Zhuang, S., Zhuang, Y., Gonzalez, J.E., et al.: Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. See https://vicuna. lmsys. org (accessed 14 April 2023) (2023)
13. Chu, X., Qiao, L., Lin, X., Xu, S., Yang, Y., Hu, Y., Wei, F., Zhang, X., Zhang, B., Wei, X., et al.: MobileVLM: A fast, reproducible and strong vision language assistant for mobile devices. arXiv preprint arXiv:2312.16886 (2023)
14. Chu, X., Qiao, L., Zhang, X., Xu, S., Wei, F., Yang, Y., Sun, X., Hu, Y., Lin, X., Zhang, B., et al.: MobileVLM V2: Faster and stronger baseline for vision language model. arXiv preprint arXiv:2402.03766 (2024)
15. Chung, H.W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, Y., Wang, X., Dehghani, M., Brahma, S., et al.: Scaling instruction-finetuned language models. arXiv preprint arXiv:2210.11416 (2022)
16. Dai, W., Li, J., Li, D., Tiong, A.M.H., Zhao, J., Wang, W., Li, B., Fung, P., Hoi, S.: InstructBLIP: Towards general-purpose vision-language models with instruction tuning. arXiv preprint arXiv:2305.06500 (2023)

17. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A large-scale hierarchical image database. In: IEEE Conference on Computer Vision and Pattern Recognition (2009)
18. Ethayarajh, K., Xu, W., Muennighoff, N., Jurafsky, D., Kiela, D.: Kto: Model alignment as prospect theoretic optimization. arXiv preprint arXiv:2402.01306 (2024)
19. Fang, A., Jose, A.M., Jain, A., Schmidt, L., Toshev, A., Shankar, V.: Data filtering networks. arXiv preprint arXiv:2309.17425 (2023)
20. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories. In: IEEE Conference on Computer Vision and Pattern Recognition - Workshops (2004)
21. Fu, C., Chen, P., Shen, Y., Qin, Y., Zhang, M., Lin, X., Yang, J., Zheng, X., Li, K., Sun, X., et al.: Mme: A comprehensive evaluation benchmark for multimodal large language models. arXiv preprint arXiv:2306.13394 (2023)
22. Gunjal, A., Yin, J., Bas, E.: Detecting and preventing hallucinations in large vision language models. arXiv preprint arXiv:2308.06394 (2023)
23. Helber, P., Bischke, B., Dengel, A., Borth, D.: EuroSAT: A novel dataset and deep learning benchmark for land use and land cover classification. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing **12**(7), 2217–2226 (2019)
24. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: LoRA: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685 (2021)
25. Hu, H., Zhang, J., Zhao, M., Sun, Z.: Ciem: Contrastive instruction evaluation method for better instruction tuning. arXiv preprint arXiv:2309.02301 (2023)
26. Hudson, D.A., Manning, C.D.: Gqa: A new dataset for real-world visual reasoning and compositional question answering. In: IEEE Conference on Computer Vision and Pattern Recognition (2019)
27. Jain, J., Yang, J., Shi, H.: Vcoder: Versatile vision encoders for multimodal large language models. arXiv preprint arXiv:2312.14233 (2023)
28. Jiang, A.Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D.S., Casas, D.d.l., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., et al.: Mistral 7b. arXiv preprint arXiv:2310.06825 (2023)
29. Jiang, C., Xu, H., Dong, M., Chen, J., Ye, W., Yan, M., Ye, Q., Zhang, J., Huang, F., Zhang, S.: Hallucination augmented contrastive learning for multimodal large language model. arXiv preprint arXiv:2312.06968 (2023)
30. Jing, L., Li, R., Chen, Y., Jia, M., Du, X.: Faithscore: Evaluating hallucinations in large vision-language models. arXiv preprint arXiv:2311.01477 (2023)
31. Koh, J.Y., Salakhutdinov, R., Fried, D.: Grounding language models to images for multimodal inputs and outputs. International Conference on Machine Learning (2023)
32. Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3D object representations for fine-grained categorization. In: IEEE International Conference on Computer Vision - Workshops (2013)
33. Li, J., Li, D., Savarese, S., Hoi, S.: BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. arXiv preprint arXiv:2301.12597 (2023)
34. Li, J., Li, D., Xiong, C., Hoi, S.: Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In: International Conference on Learning Representations (2022)

35. Li, L., Xie, Z., Li, M., Chen, S., Wang, P., Chen, L., Yang, Y., Wang, B., Kong, L.: Silkie: Preference distillation for large visual language models. arXiv preprint arXiv:2312.10665 (2023)
36. Li, S., Lin, R., Pei, S.: Multi-modal preference alignment remedies regression of visual instruction tuning on language model. arXiv preprint arXiv:2402.10884 (2024)
37. Li, Y., Du, Y., Zhou, K., Wang, J., Zhao, W.X., Wen, J.R.: Evaluating object hallucination in large vision-language models. arXiv preprint arXiv:2305.10355 (2023)
38. Li, Y., Bubeck, S., Eldan, R., Del Giorno, A., Gunasekar, S., Lee, Y.T.: Textbooks are all you need ii: phi-1.5 technical report. arXiv preprint arXiv:2309.05463 (2023)
39. Lin, B., Tang, Z., Ye, Y., Cui, J., Zhu, B., Jin, P., Zhang, J., Ning, M., Yuan, L.: Moe-llava: Mixture of experts for large vision-language models. arXiv preprint arXiv:2401.15947 (2024)
40. Liu, F., Lin, K., Li, L., Wang, J., Yacoob, Y., Wang, L.: Mitigating hallucination in large multi-modal models via robust instruction tuning. In: International Conference on Learning Representations (2023)
41. Liu, H., Li, C., Li, Y., Lee, Y.J.: Improved baselines with visual instruction tuning. arXiv preprint arXiv:2310.03744 (2023)
42. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. Advances on Neural Information Processing Systems (2024)
43. Liu, T., Zhao, Y., Joshi, R., Khalman, M., Saleh, M., Liu, P.J., Liu, J.: Statistical rejection sampling improves preference optimization. arXiv preprint arXiv:2309.06657 (2023)
44. Liu, Y., Duan, H., Zhang, Y., Li, B., Zhang, S., Zhao, W., Yuan, Y., Wang, J., He, C., Liu, Z., et al.: MMBench: Is your multi-modal model an all-around player? arXiv preprint arXiv:2307.06281 (2023)
45. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
46. Lu, P., Mishra, S., Xia, T., Qiu, L., Chang, K.W., Zhu, S.C., Tafjord, O., Clark, P., Kalyan, A.: Learn to explain: Multimodal reasoning via thought chains for science question answering. Advances in Neural Information Processing Systems (2022)
47. Mishra, S., Khashabi, D., Baral, C., Hajishirzi, H.: Cross-task generalization via natural language crowdsourcing instructions. arXiv preprint arXiv:2104.08773 (2021)
48. Mitchell, E.: A note on dpo with noisy preferences & relationship to ipo (2024)
49. Nilsback, M.E., Zisserman, A.: Automated flower classification over a large number of classes. In: Indian conference on computer vision, graphics & image processing. pp. 722–729 (2008)
50. OpenAI: Introducing chatgpt (2022)
51. OpenAI: Gpt-4v(ision) system card (2023)
52. Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al.: Training language models to follow instructions with human feedback. Advances in Neural Information Processing Systems (2022)
53. Parkhi, O.M., Vedaldi, A., Zisserman, A., Jawahar, C.: Cats and dogs. In: IEEE Conference on Computer Vision and Pattern Recognition (2012)
54. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch (2017)
55. Peng, Z., Wang, W., Dong, L., Hao, Y., Huang, S., Ma, S., Wei, F.: Kosmos-2: Grounding multimodal large language models to the world. arXiv preprint arXiv:2306.14824 (2023)

56. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning (2021)
57. Rafailov, R., Sharma, A., Mitchell, E., Manning, C.D., Ermon, S., Finn, C.: Direct preference optimization: Your language model is secretly a reward model. Advances on Neural Information Processing Systems (2024)
58. Rajbhandari, S., Ruwase, O., Rasley, J., Smith, S., He, Y.: Zero-infinity: Breaking the gpu memory wall for extreme scale deep learning. In: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (2021)
59. Rasley, J., Rajbhandari, S., Ruwase, O., He, Y.: Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (2020)
60. Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., et al.: Laion-5b: An open large-scale dataset for training next generation image-text models. Advances in Neural Information Processing Systems (2022)
61. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017)
62. Singh, A., Natarajan, V., Shah, M., Jiang, Y., Chen, X., Batra, D., Parikh, D., Rohrbach, M.: Towards VQA models that can read. In: IEEE Conference on Computer Vision and Pattern Recognition (2019)
63. Soomro, K., Zamir, A.R., Shah, M.: Ucf101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402 (2012)
64. Stiennon, N., Ouyang, L., Wu, J., Ziegler, D., Lowe, R., Voss, C., Radford, A., Amodei, D., Christiano, P.F.: Learning to summarize with human feedback. Advances in Neural Information Processing Systems (2020)
65. Sun, Z., Shen, S., Cao, S., Liu, H., Li, C., Shen, Y., Gan, C., Gui, L.Y., Wang, Y.X., Yang, Y., et al.: Aligning large multimodal models with factually augmented rlhf. arXiv preprint arXiv:2309.14525 (2023)
66. Team, G., Anil, R., Borgeaud, S., Wu, Y., Alayrac, J.B., Yu, J., Soricut, R., Schalkwyk, J., Dai, A.M., Hauth, A., et al.: Gemini: a family of highly capable multimodal models. arXiv preprint arXiv:2312.11805 (2023)
67. Thoppilan, R., De Freitas, D., Hall, J., Shazeer, N., Kulshreshtha, A., Cheng, H.T., Jin, A., Bos, T., Baker, L., Du, Y., et al.: Lamda: Language models for dialog applications. arXiv preprint arXiv:2201.08239 (2022)
68. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al.: Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023)
69. Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al.: Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288 (2023)
70. Wang, J., Yang, Z., Hu, X., Li, L., Lin, K., Gan, Z., Liu, Z., Liu, C., Wang, L.: Git: A generative image-to-text transformer for vision and language. arXiv preprint arXiv:2205.14100 (2022)
71. Wang, J., Wang, Y., Xu, G., Zhang, J., Gu, Y., Jia, H., Yan, M., Zhang, J., Sang, J.: An LLM-free multi-dimensional benchmark for MLLMs hallucination evaluation. arXiv preprint arXiv:2311.07397 (2023)

72. Wang, J., Zhou, Y., Xu, G., Shi, P., Zhao, C., Xu, H., Ye, Q., Yan, M., Zhang, J., Zhu, J., et al.: Evaluation and analysis of hallucination in large vision-language models. arXiv preprint arXiv:2308.15126 (2023)
73. Williams, R.J.: Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine learning **8**, 229–256 (1992)
74. Xiao, J., Hays, J., Ehinger, K.A., Oliva, A., Torralba, A.: Sun database: Large-scale scene recognition from abbey to zoo. In: IEEE Conference on Computer Vision and Pattern Recognition (2010)
75. Xu, J., Lee, A., Sukhbaatar, S., Weston, J.: Some things are more cringe than others: Preference optimization with the pairwise cringe loss. arXiv preprint arXiv:2312.16682 (2023)
76. Ye, Q., Xu, H., Xu, G., Ye, J., Yan, M., Zhou, Y., Wang, J., Hu, A., Shi, P., Shi, Y., et al.: mplug-owl: Modularization empowers large language models with multimodality. arXiv preprint arXiv:2304.14178 (2023)
77. You, H., Zhang, H., Gan, Z., Du, X., Zhang, B., Wang, Z., Cao, L., Chang, S.F., Yang, Y.: Ferret: Refer and ground anything anywhere at any granularity. arXiv preprint arXiv:2310.07704 (2023)
78. Zhai, B., Yang, S., Zhao, X., Xu, C., Shen, S., Zhao, D., Keutzer, K., Li, M., Yan, T., Fan, X.: Halle-switch: Rethinking and controlling object existence hallucinations in large vision language models for detailed caption. arXiv preprint arXiv:2310.01779 (2023)
79. Zhai, X., Mustafa, B., Kolesnikov, A., Beyer, L.: Sigmoid loss for language image pre-training. arXiv preprint arXiv:2303.15343 (2023)
80. Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., Dewan, C., Diab, M., Li, X., Lin, X.V., et al.: Opt: Open pre-trained transformer language models. arXiv preprint arXiv:2205.01068 (2022)
81. Zhao, H., Cai, Z., Si, S., Ma, X., An, K., Chen, L., Liu, Z., Wang, S., Han, W., Chang, B.: Mmicl: Empowering vision-language model with multi-modal in-context learning. arXiv preprint arXiv:2309.07915 (2023)
82. Zhao, Y., Joshi, R., Liu, T., Khalman, M., Saleh, M., Liu, P.J.: Slic-hf: Sequence likelihood calibration with human feedback. arXiv preprint arXiv:2305.10425 (2023)
83. Zhao, Z., Wang, B., Ouyang, L., Dong, X., Wang, J., He, C.: Beyond hallucinations: Enhancing LVLMs through hallucination-aware direct preference optimization. arXiv preprint arXiv:2311.16839 (2023)
84. Zhou, C., Liu, P., Xu, P., Iyer, S., Sun, J., Mao, Y., Ma, X., Efrat, A., Yu, P., Yu, L., et al.: Lima: Less is more for alignment. Advances in Neural Information Processing Systems (2024)
85. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Conditional prompt learning for vision-language models. In: IEEE Conference on Computer Vision and Pattern Recognition (2022)
86. Zhu, D., Chen, J., Shen, X., Li, X., Elhoseiny, M.: MiniGPT-4: Enhancing vision-language understanding with advanced large language models. arXiv preprint arXiv:2304.10592 (2023)
87. Zhu, Y., Zhu, M., Liu, N., Ou, Z., Mou, X., Tang, J.: LLaVA-phi: Efficient multi-modal assistant with small language model. arXiv preprint arXiv:2401.02330 (2024)
88. Ziegler, D.M., Stiennon, N., Wu, J., Brown, T.B., Radford, A., Amodei, D., Christiano, P., Irving, G.: Fine-tuning language models from human preferences. arXiv preprint arXiv:1909.08593 (2019)