

Report

Align LLaVA-1.6 7B VLM on a human preference dataset
(RLAIF-V) using ORPO (Monolithic Preference Optimization
without Reference Model) method

Sameer Sharma Poudel

May 28, 2025

Introduction

The goal of the task was to align a large multimodal model, called LLaVA: Large Language and Vision Assistant, on human preference dataset using Monolithic Preference Optimization without Reference Model (ORPO) method. LLaVA, being a multimodal model, works with multi format data in the form of language and vision. Among the available checkpoints, the 1.6 version of the model with CLIP vision encoder and Mistral large language model (LLM), which altogether has approximately 7 billion parameters, is used for the alignment task. Alignment typically refers to the process of making the output of a model more consistent with human values, preferences, or specific desired behaviors. The use of ORPO method obviates the need for a separate reward model or reference model, unlike its predecessors, during the alignment of the model on human preferences. As the ORPO method offers an end-to-end optimization with greater efficiency and scalability, it is gaining traction in machine learning research community and this coding challenge task also follows the same avenue. The following sections report about the essentials of the undertaken task.

The scripts for the task is available in the following link: [VLMAignment](#)

How is the problem tackled ?

Computing Resource

The model is trained on laptop with NVIDIA RTX A2000 8 GB GPU. Several attempts were made to train the model on free GPU instances available in Google Colab and Kaggle. However, the training sessions stopped many times even without training for the first epoch. Therefore, the training on local machine is preferred for consistent GPU performance.

Data Preparation

RLAIF-V human preference dataset is deployed for the model alignment. It is a large-scale multimodal feedback dataset. The dataset provides high-quality feedback with a total number of 83,132 preference pairs, where the instructions are collected from a diverse range of datasets including MSCOCO, ShareGPT-4V, MovieNet, Google Landmark v2, VQA v2,

OKVQA, and TextVQA. However, only 30,000 preference pairs are used for this task due to time, memory and computing resources constraint. Among them, 90% is used for training while the remaining 10% is used for validation during training. To avoid the device run out of memory, batch size of 2 is used while training and validation.

The input dataset is prepared in the format that the model expects. LLaVa model doesn't directly take images and text as input, but rather pixel values (which is a resized, rescaled, normalized and optionally splitted version of the receipt images), input ids (which are token indices in the vocabulary of the model), etc. This is taken care of by the processor. Since a single dedicated processor for both image and text is not available, the image and the text need to be processed separately. The CLIPImageProcessor performs several critical preprocessing steps to ensure that the image is in the right format and scale for CLIP vision encoder. Since the inputs to the model are text and image, the input prompt needs to be formatted such that it contains both text prompt and image. The input prompt is formatted following a chat template. A Mistral Instruct Conversation type chat template is used for this task. The formatted prompt is then tokenized with the help of the tokenizer. $<image>$, $[INST]$ and $[/INST]$ are added as special tokens and the token embedding size is resized. The padding side for the tokenizer is changed to right from the default left. The token for padding is also changed from $<unk>$ to $<s>$. The processing of the inputs from image processor and tokenizer yields *pixel_values* and *input_ids* respectively. The pixel values and input ids along with other inputs are fed into the model to obtain the output logits.

No additional pre-training or post-training steps have been carried out except the alignment training.

Fine-Tuning Techniques

As the model has 7 billion parameters, it is going to have quite an impact on the amount of memory used. For reference, finetuning a model using the AdamW optimizer, about 18 times the amount of parameters in GB of GPU RAM memory is required. So in this case, 18x7 billion bytes equivalent to 126 GB of GPU RAM is needed if all the parameters of the model are updated. That's huge and infeasible considering the computing resource that is at hand. Therefore, to optimize the training efficiency and resource utilization, Quantized Low-Rank Adaptation (QLoRA) technique is employed. QLoRA combines the following two techniques during finetuning:

- Quantization: The original model weights are frozen and quantized (e.g., to 4-bit precision) to drastically reduce memory usage during fine-tuning.
- Low-rank Adaptation: Small, trainable low-rank matrices known as adapters are added to the specific layers of the model to capture task-specific updates. During fine-tuning, only the low-rank adapters are updated.

The quantization of the model is configured with NF4 and double quantization. Similarly, for parameter-efficient fine-tuning, LoRA is configured with a rank of 8 and 16, an alpha value of 16 and the dropout rate of 0.05. The adapters are added to *q-proj*, *k-proj*, *v-proj*, *mm-proj*, *gate-proj*, *up-proj* and *lm-head* layers in the model. Upon the addition of new special tokens, the embedding layers of the new tokens are also set to be trainable. There are altogether 15,624,216 trainable parameters for the alignment task. Moreover, the following techniques are used for efficient finetuning:

- Mixed precision training
- Gradient accumulation
- Gradient checkpointing

The model is optimized using AdamW with a cosine annealing learning rate scheduler and accelerated with mixed-precision training using the *accelerate* library. Monolithic Preference Optimization without Reference Model (ORPO) loss is used to control the preference learning behavior of the model.

The organisation of the codebase

The codebase has several scripts in python files that are used for finetuning.

- **main.py:** It is the main script to set up the environment, load the model and data, and run the training loop. Workflow inside this file is as follows:
 - Initialize W&B tracking.
 - Set up CUDA device and check GPU availability.

- Load the pretrained model, tokenizer, and image processor.
 - Prepare the dataset, including training-validation splitting.
 - Initialize the optimizer and learning rate scheduler.
 - Prepare the model and dataloaders with Accelerate.
 - Train the model for the specified number of epochs.
 - Log training and validation metrics, including example generations.
 - Save checkpoints and the final fine-tuned model.
- **prepare_model_tokenizer.py:** This script prepares a model for Parameter-Efficient Fine-Tuning (PEFT) using QLoRA, as well as prepares and configures the tokenizer and model for fine-tuning with image and instruction tokens.
 - **dataloader.py:** This script defines a custom dataset class and related utilities for training a vision-language model using image and text data. It supports flexible tokenization, data loading, and efficient batching.
 - **train_and_validate.py:** This script defines functions for training and validating models during fine-tuning. It uses mixed precision training and gradient accumulation with the help of the accelerator from the Hugging Face Accelerate library. It also logs the essential metrics for training and validation.
 - **orpo.py:** This script defines the ORPOLoss class, which computes the ORPO loss for fine-tuning a large language model. The loss function compares the log probabilities of chosen and rejected responses to optimize the model for human preference alignment.
 - **utils.py:** This script contains the utility function to generate and log example inferences from the model.
 - **generator.py:** This script contains the class that generates text responses conditioned on image and text inputs using an autoregressive language model. It incorporates CLIP for vision-grounded verification, attempting to reduce hallucinations.

How to launch training/validation ?

In order to launch training/validation, the necessary libraries listed in *requirements.txt* file need to be installed first in the work environment. The specific version of the model for

this task is not directly accessible from Hugging Face library, so the official repository of the model should be cloned and the necessary packages need to be installed in order to load the desired version of the model. The configuration or arguments related to finetuning are then needed to be defined inside `get_args_parser()` function in `main.py` python file mentioned in previous section. To start the finetuning process, run the following command in command line interface: `python main.py`.

Training and Validation Result

As part of the task, two finetuning runs were carried out; the runs differ from each other mainly on LoRA rank and learning rate parameters, the other training parameters remain same. The training and validation statistics for these runs are tabulated in *table 1*. The training and validation loss are an ORPO loss. Validation accuracy measures the fraction of preference pairs in the validation set where the model assigns a higher probability (or score) to the chosen (positive) response than to the rejected (negative) response. As observed in the table, there is no significant difference between training loss and its counterpart validation loss evaluated at different points and tabulated row-wise in the table. However, a look at the losses in the first three rows for continued finetuning suggests that the training loss is decreasing while the validation loss is increasing. Another look at the validation accuracy gives an indication of decrement with the continued finetuning. The contrary trend of validation loss and accuracy thus doesn't give much information about the learning behavior of the model, so it is studied meticulously with the help of other metrics below.

The finetuning at first is carried out with rank 8 LoRA adapters and learning rate of 2E-4. While finetuning, the process crashed after running a little more steps after the first epoch. The finetuning process is then continued for further two more epochs using the saved checkpoint of the first epoch, keeping the training parameters same. Therefore, separate plots are provided for evaluation for this 2-step finetuning process. In the plots shown below, the values in y-axis are plotted against *global_step* in x-axis. The *global_step* is the total number of step (the number of times the training data goes through the model in batch in an epoch) across the epochs. For.e.g. in this case, with 27000 training data and batch size of 2, the number of steps in an epoch is 13500. If trained for two epochs continuously, the number of *global_step* at the end of epoch 1 and epoch 2 are 13500 and 27000 respectively.

To evaluate the training and validation, metrics such as mean log probability of the chosen

response, mean log probability of the rejected response, difference between mean log probability of the chosen response and the rejected response, ORPO loss and negative log likelihood loss are used. They are denoted as *pos_prob_mean*, *neg_prob_mean*, *log_odds*, *orpo_loss* and *nll_loss* respectively in the plots.

As seen in *figure 1* and *figure 2*, the mean log probablity of the chosen responses and the rejected responses remain between -7 and -10 for most part of the finetuning except occasional spikes across the global steps across the epochs. However, the decreasing trend is seen and slowly going beyond -10 near the end global steps in both figures for both chosen and rejected responses. The difference between the mean log probablity of the chosen responses and rejected responses, denoted as *log_odds*, can be observed in *figure 3* and *figure 4*. *log_odds* is seen to be slightly increasing near the end global steps in both figures. The slight decreasing trend of overall ORPO loss can be seen in *figure 5* and *figure 6*. Upon the observation of the trends in *figure 1* to *figure 6*, the following can be interpreted:

- **Both *pos_prob_mean* and *neg_prob_mean* decreasing:** This means the model is assigning lower probabilities (more negative log probablities) to both the chosen and rejected responses as training progresses ie. the model is becoming less confident in both responses.
- ***neg_prob_mean* decreasing more than *pos_prob_mean*:** This suggests that the model is pushing down bad responses more than good ones. Even though the model is losing confidence in chosen responses, it is losing even more confidence in the rejected responses.
- ***log_odds* has a slightly increasing trend:** This means the model still favors the chosen response more than the rejected one, but with lower absolute confidence.
- **ORPO loss decreasing:** A steady decline confirms the model is effectively minimizing the combined objective (language modeling loss + preference loss).

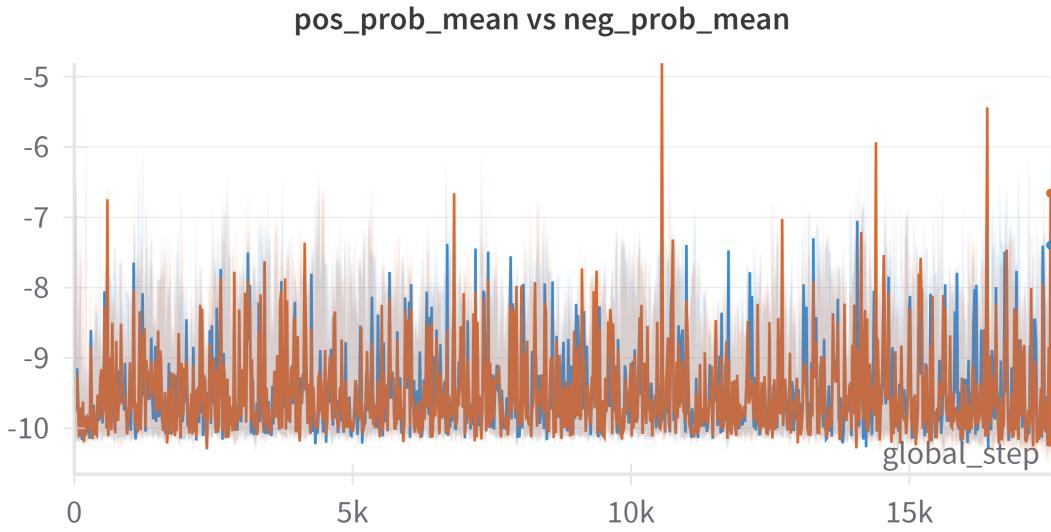
The generated response from the finetuned model for the given image and text prompt along with the chosen and rejected response can be seen in *figure 7*, *figure 8* and *figure 9*. As seen in all three figures, the model generates a factually incorrect response despite the factually correct chosen and rejected reference being provided. The model understands the text prompt but fails to constrain its response to the actual visual content. This suggests failure in visual grounding — the model ignores the actual image content and generates a

plausible-sounding but false response, and indicates a severe case of hallucination in response generation.

Although *neg_prob_mean* decreasing faster than *pos_prob_mean*, slight increasing trend of *log_odds* and slight decreasing trend of ORPO loss suggest the model inclination towards preference optimization, *pos_prob_mean* and *neg_prob_mean* both having more negative values at around -10 suggest the model having low confidence in generating responses. The lesser confidence of the model is evidenced by the factually incorrect responses generated by it.

The finetuning for the second time is carried out with LoRA adapters with increased rank of 16 and reduced learning rate of 2.5E-6. There is no observation of significant increasing or decreasing trends of *pos_prob_mean* and *neg_prob_mean* in *figure 10* and neither of *log_odds* in *figure 11*. However, a slight decreasing trends are observed for *nll_loss* and *orpo_loss* in *figure 12* and *figure 13* respectively. Furthermore, the generated response from the finetuned model can be seen in *figure 14* and *figure 15*. Upon the observation of results in figures from *figure 10* to *figure 15*, the model seems to have low confidence in generating factual response although its behavior inclines towards preference optimization and prediction of preferred response, similar to the result of first time finetuning.

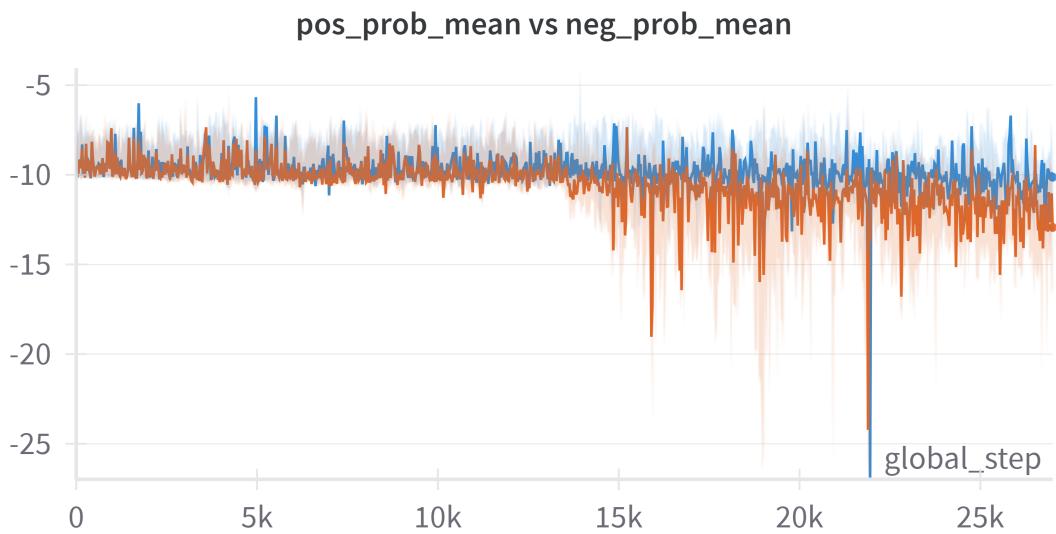
The fluctuating behavior in the plots is due to the low effective batch size.



*Figure 1: Mean log probability of the chosen response (*pos_prob_mean*) vs rejected response (*neg_prob_mean*) across global steps for 1 epoch*

Finetuning Remark	Epoch	Train Loss	Val Loss	Val Accuracy
Training 1: with 8 LoRA rank, 2E-4 LR	1	1.232	1.193	0.504
Training 1 cont.: with 8 LoRA rank, 2E-4 LR	1	0.991	1.197	0.538
Training 1 cont.: with 8 LoRA rank, 2E-4 LR	2	0.751	1.299	0.604
Training 2 : with 16 LoRA rank, 2.5E-6 LR	1	1.336	1.270	0.487

Table 1: Training and Validation Statistics

Figure 2: Mean log probability of the *chosen response (pos_prob_mean)* vs *rejected response (neg_prob_mean)* across global steps for 2 more epochs after continued finetuning with the checkpoint in figure 1

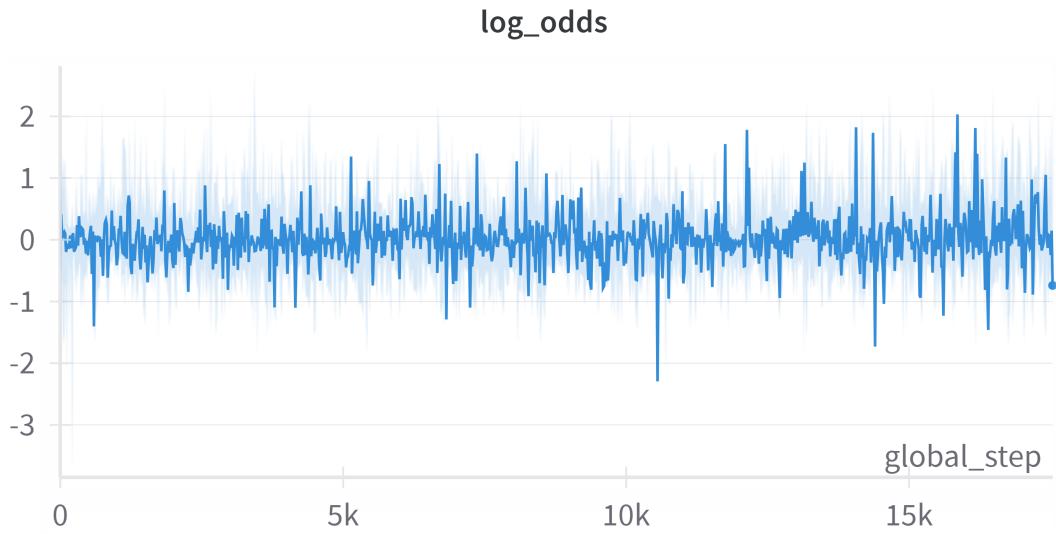


Figure 3: Difference between mean log probability of the chosen responses and the rejected responses across global steps for 1 epoch

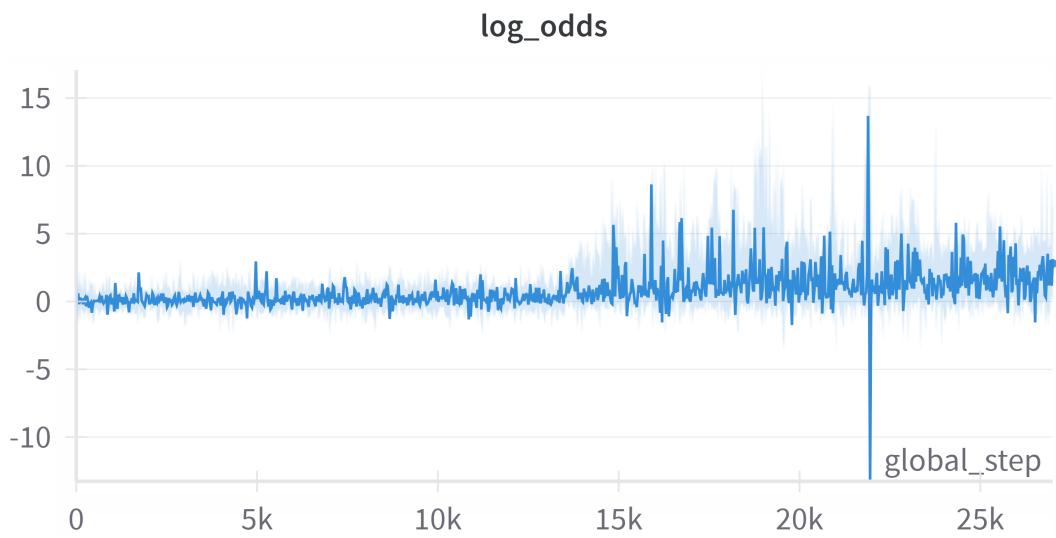


Figure 4: Difference between mean log probability of the chosen responses and the rejected responses across global steps for 2 more epochs after continued finetuning with the checkpoint in figure 3

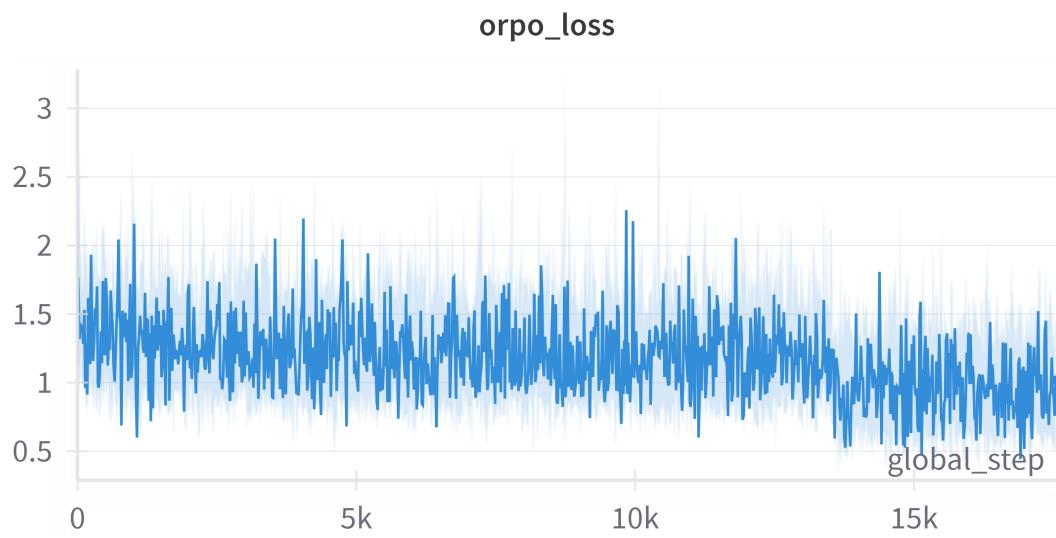


Figure 5: ORPO loss across global steps for 1 epoch

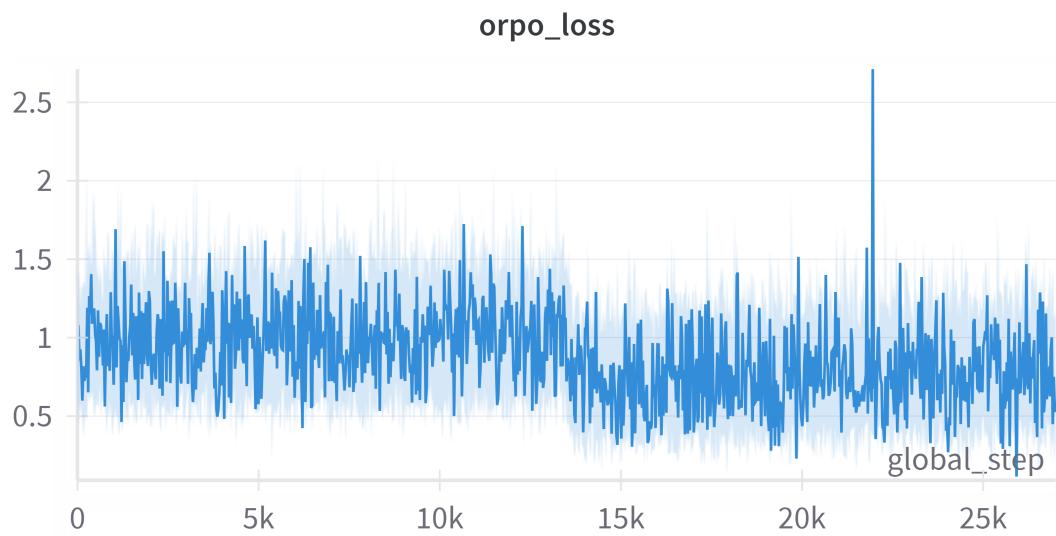


Figure 6: ORPO loss across global steps for 2 more epochs after continued finetuning with the checkpoint in figure 5

	image	prompt	response	chosen_ref	rejected_ref
1		[INST]<image> What team is this player playing for?[/INST]	[>[INST]<image>\nWhat team is this player playing for? [/INST]The player in the image is playing for the Atlanta Braves team. The Atlanta Braves are a professional baseball team based in Atlanta, Georgia, and they compete in Major League Baseball (MLB). The team's colors are red and white, and their logo features a stylized representation of a tomahawk, which is a weapon commonly used by Native American tribes. The presence of the Atlanta Braves jersey suggests that this player is a member of the team and supports its values and goals. The Braves have a rich history and tradition in baseball, dating back to their founding in 1871 as the Boston Red Stockings. Over the years, they have won numerous championships and are known for their strong fan base and loyal supporters. So, it's possible	[INST]<image> What team is this player playing for?[/INST]The player in the image is wearing a yellow uniform and appears to be a baseball player. However, without specific team logos or identifiable markings on the uniform, it's challenging to determine the exact team they are playing for based on this single image.	

Figure 7: The generated response from the finetuned model after 1 epoch

	image	prompt	response	chosen_ref	rejected_ref
1		[INST]<image> Create a detailed description describing the elements of the given worldwide landmark image. Identify the name of the landmark and describe the objects, colors, perspectives, etc.[/INST]	[>[INST]<image>\nCreate a detailed description describing the elements of the given worldwide landmark image. Identify the name of the landmark and describe the objects, colors, perspectives, etc.[/INST]The image features the Hoi An Ancient Town in Vietnam. The scene includes a traditional wooden boat floating on the water, surrounded by colorful buildings with pagodas and ornate details. The landmark is known for its unique architecture and lively atmosphere, reflecting its rich history and cultural significance. The perspective of the image gives a sense of being immersed in the town, showcasing the harmonious blend of nature and man-made structures.'	[INST]<image> Create a detailed description describing the elements of the given worldwide landmark image. Identify the name of the landmark and describe the objects, colors, perspectives, etc.[/INST]The image features the Teatro Greco di Taormina, an ancient Greek amphitheater located in Sicily, Italy. The landmark consists of large stone benches arranged in a circular pattern, surrounded by green grass and rocks. In the background, there's a picturesque view of mountains and clear blue sky with white clouds.	

Figure 8: The generated response at epoch 1 from the model after continued finetuning for 2 more epochs with the checkpoint in figure 7

	image	prompt	response	chosen_ref	rejected_ref
1		[INST]<image> Tell me what catches your eye in the image, and describe those elements in depth.[/INST]	[>[INST]<image>\nTell me what catches your eye in the image, and describe those elements in depth.[/INST]The image prominently features a well-designed bathroom countertop with a sink and faucet. The contrast between the black marble top and the silver fixtures creates an elegant and modern aesthetic. A neatly folded towel adds a touch of warmth to the scene, suggesting that the space is not only visually appealing but also functional for daily use. The combination of these elements makes the bathroom countertop a focal point in the image.'	[INST]<image> Tell me what catches your eye in the image, and describe those elements in depth.[/INST]The image prominently features a cartoon girl in a red tank top, blue shorts, and vibrant red shoes. Her brown hair is adorned with a matching red headband. She grasps a large flag, which is predominantly white with an iconic sun symbol on it. A distinguishing feature of her attire is the black belt cinched around her waist.	[INST]<image> Tell me what catches your eye in the image, and describe those elements in depth.[/INST]The image features a cartoon girl in a red tank top, blue shorts, and vibrant red shoes. Her brown hair is adorned with a matching red headband. She grasps a large flag, which is predominantly white with an iconic sun symbol on it. A distinguishing feature of her attire is the black belt cinched around her waist.

Figure 9: The generated response at epoch 2 from the model after continued finetuning for 2 more epochs with the checkpoint in figure 7

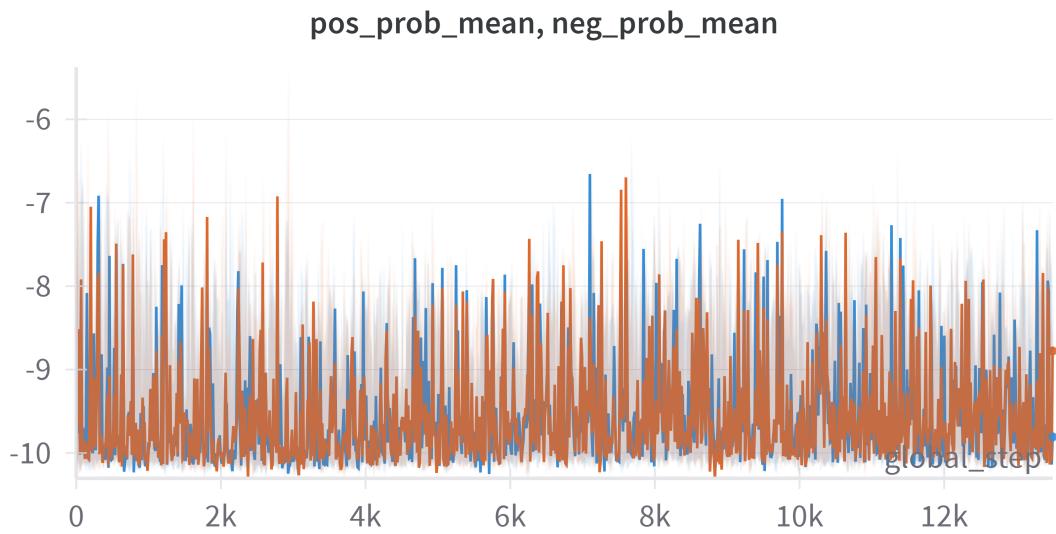


Figure 10: Mean log probability of the chosen response (`pos_prob_mean`) vs rejected response (`neg_prob_mean`) across global steps for 1 epoch for finetuning with 16 rank LoRA adapters and 2.5E-6 learning rate

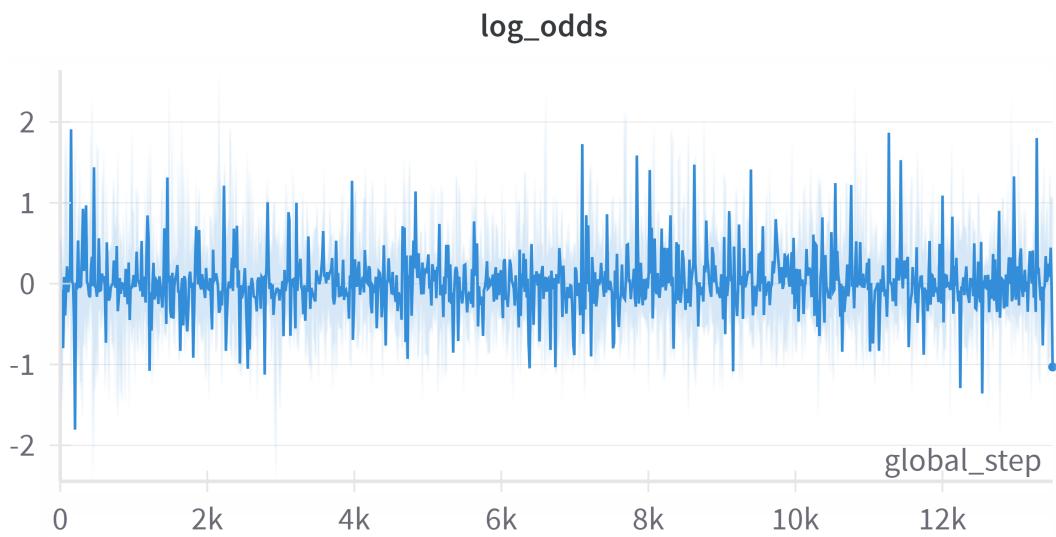


Figure 11: Difference between mean log probability of the chosen responses and the rejected responses across global steps for 1 epoch for finetuning with 16 rank LoRA adapters and 2.5E-6 learning rate

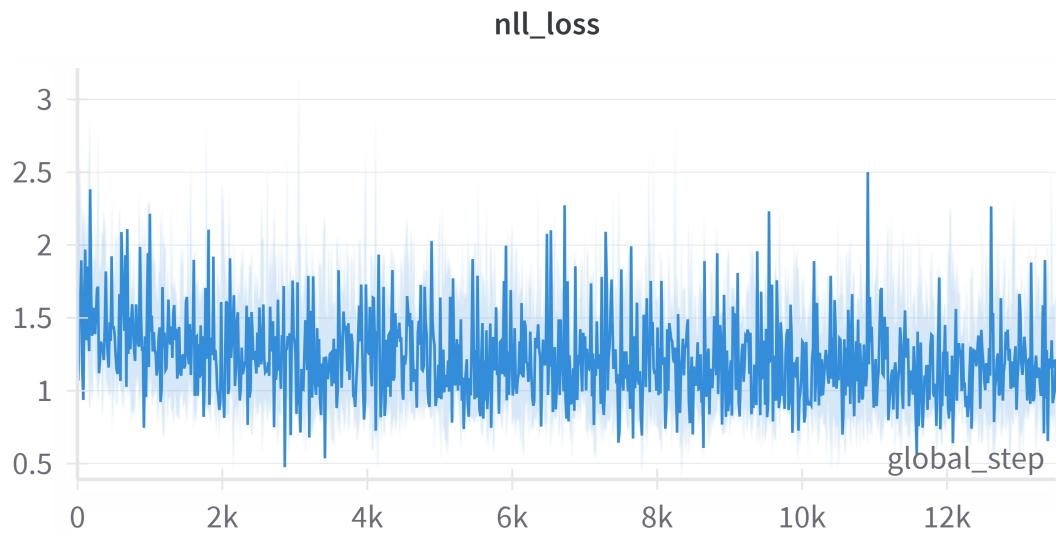


Figure 12: Negative Log Likelihood loss across global steps for 1 epoch for finetuning with 16 rank LoRA adapters and $2.5E-6$ learning rate

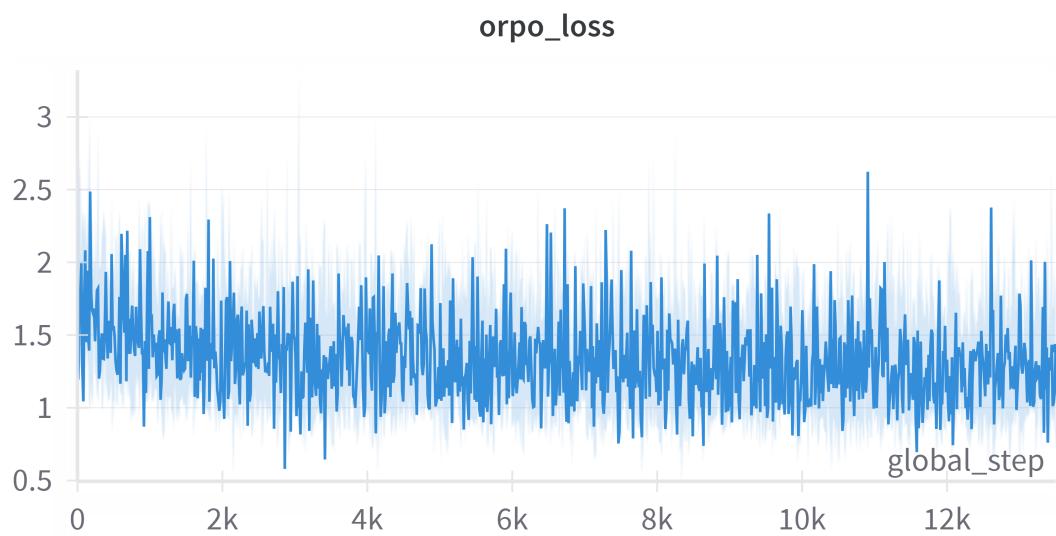


Figure 13: ORPO loss across global steps for 1 epoch for finetuning with 16 rank LoRA adapters and $2.5E-6$ learning rate

	image	prompt	response	chosen_ref	rejected_ref
1		[INST]<image> Is there anything in the suitcase?[/INST]	[INST]<image> Is there anything in the suitcase?[/INST]Yes, there is a suitcase in the image. The suitcase is placed on the ground near the woman who is sitting on the bench. It's likely that the suitcase contains personal belongings or items for her trip. The presence of the suitcase suggests that she might be traveling or has recently arrived at her destination.	[INST]<image> Is there anything in the suitcase?[/INST]Yes, there is something in the suitcase. The image shows a woman holding a piece of luggage while smiling for the camera.	[INST]<image> Is there anything in the suitcase?[/INST]Yes, there is something in the suitcase. The image shows a woman reaching into her luggage and grabbing someone's hand while standing on top of it.

Figure 14: The generated response from the finetuned model with 16 rank LoRA adapters and 2.5E-6 learning rate after 1 epoch

	image	prompt	response	chosen_ref	rejected_ref
2		[INST]<image> Identify and describe each object in the image in detail.[/INST]	[INST]<image> Identify and describe each object in the image in detail.[/INST]In the image, there are several objects that can be identified. The main focus is on a large, white airplane with a distinctive red and blue stripe on its body. The plane is in the process of taking off from a runway, suggesting it's either departing or arriving at an airport. In addition to the airplane, there are also multiple people present in the scene. Some of them appear to be standing near the runway, possibly airport staff or ground crew members who are involved in the operation of the aircraft. Their presence adds a sense of scale and activity to the image, highlighting the complex processes that take place at airports.	[INST]<image> Identify and describe each object in the image in detail.[/INST]The image showcases a scene from the Shanghai Stock Exchange. At the forefront, there's a large sign that reads "STAR MARKET" in vibrant shades of pink and blue. This sign is prominently displayed against a backdrop of a bustling trading floor. In the background, screens displaying stock prices fill the scene. These screens are scattered across the room, indicating a high level of activity on the exchange. The colors of these screens are predominantly white and green, adding to the overall modern and tech-driven ambiance of the setting. A tall column stands out amidst the sea of screens. It's adorned	[INST]<image> Identify and describe each object in the image in detail.[/INST]The image showcases a vibrant scene from the Shanghai Stock Exchange. Dominating the frame are several screens, each displaying an array of data in rows and columns. These screens are illuminated against a backdrop of a blue wall adorned with red text. Prominently displayed on the left side of the image is a large sign that reads "Star Market", written in bold white letters on a pink background. This sign appears to be the focal point of the space, drawing attention amidst the sea of data on the screens behind it. On the right side of the image, another sign can be seen, this one bearing Chinese characters that translate to "Shanghai Stock Exchange". This sign is smaller

Figure 15: The generated response from the finetuned model with 16 rank LoRA adapters and 2.5E-6 learning rate after 1 epoch

Future possible improvements/fixes

As observed in the training and validation result in the previous section, the model suffers from hallucination in response generation. The possible causes of the issue and the possible improvements or fixes are discussed in the following.

- **Dataset level:**

- The current dataset doesn't contain the hallucinated responses as the rejected responses. The dataset can be modified by adding the hallucinated responses as the rejected ones so that the model is penalized for generating hallucinating responses during finetuning.

- **Training level:**

- Low-rank adapters can sometimes fail to capture the visual-textual alignment, so the finetuning with higher ranks can be experimented to better capture visual-textual relationships.
- Fully finetuning the layers that map image features to the LLM's token space and if possible the vision encoder or at least the last few layers of the encoder.
- Adding auxiliary losses to penalize generations inconsistent with image content.
- Assigning higher learning rate for vision modules and lower for text modules.

- **Architecture level:**

- Feeding image features into multiple LLM layers (not just the first layer).
- Forcing the language model to attend to image tokens when generating key words.