Text Conditional Image Embeddings Using Prior Large Multimodal Models

Roey Ron Maor Lavi Gal Tchinio

Abstract

This paper investigates the hidden state representations within the LLaVA (Large Language and Vision Assistant) model, a multi-modal architecture that integrates language understanding with visual processing. Our research extends previous work on hidden representation research by specifically examining LLaVA's hidden states across various layers and experimental setups. We conduct a series of experiments to uncover what information these hidden representations hold about image and text prompts, and explore their potential in solving diverse multi-modal tasks. Based on our findings, we propose a novel method for extracting meaningful image embeddings conditioned on text input, leveraging LLaVA's hidden states. Our work contributes to a deeper understanding of multi-modal systems and offers new possibilities for leveraging hidden representations in practical applications. The results of this study have implications for improving model interpretability, enhancing multi-modal task performance, and informing future developments in multi-modal architectures. Our code and dataset are available at https://github.com/roey1rg/TCIE

1 Introduction

The study of hidden representations in neural networks has become increasingly important for understanding and interpreting the behavior of complex models. This line of research aims to uncover the internal mechanisms of these models and leverage this knowledge for various applications. Recent work in this area has focused on developing techniques to probe and analyze the intermediate layers of neural networks. (Alain and Bengio, 2018) introduced the concept of linear classifier probes, which are trained independently of the main model to assess the suitability of features at different layers for classification tasks. Expanding on this idea, (Ghandeharioun et al., 2024) proposed the Patchscopes framework, which utilizes the language model itself to explain its internal representations in natural language. Patchscopes also introduces novel possibilities, including using more capable models to explain the representations of smaller models and performing multihop reasoning for error correction.

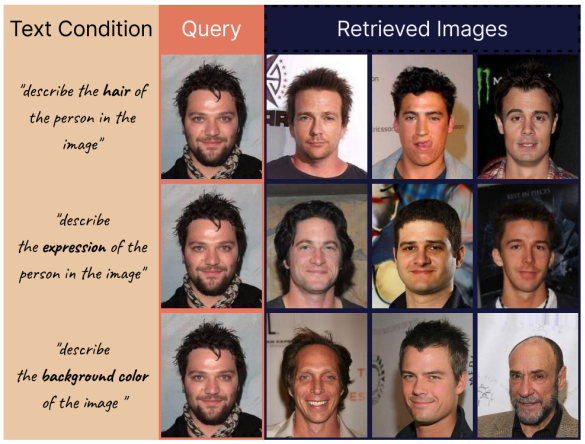

Figure 1: Search results for different text prompts

Our work builds upon these foundations insights by focusing specifically on the hidden state representations of the multi-modal LLaVA model (Liu et al., 2023). We constructed experiments to research what information they might hold about the image and text prompts on different layers, on various setups and their possibilities to help solve text conditioned image representation tasks. Based of our findings, we propose a method for extracting meaningful image embeddings conditioned on text input, leveraging the model's hidden states. We show how these embeddings can help solve tasks that current methods require specific training for.

LLaVA (Liu et al., 2023) (Large Language and Vision Assistant) is a multi-modal model that combines language understanding with visual processing. Its architecture consists of two main components: Language Model: (Originally LLaMA, we use 1.6 version which is Mistral based), a large language model trained on textual data. Vision

Encoder: Typically uses CLIP ViT-L/14 to process visual inputs. Visual tokens from the vision encoder are projected into the language model's embedding space. A learnable adapter layer connects the vision and language components. The model uses a projection layer to align visual and textual representations. LLaVA processes images and text simultaneously, allowing for tasks like visual question answering and image-based dialogue. The integration of visual and textual information occurs in the model's hidden layers, which is the focus of our research into hidden representations.

2 Related Work

2.1 Conditional Image Similarity

Conditional Image Similarity refers to the concept of determining how similar images are to each other based on specific, user-defined conditions or criteria. Unlike traditional image similarity measures that rely on a fixed notion of similarity, conditional similarity allows for dynamic and context-dependent comparisons. GeneCIS: A Benchmark for General Conditional Image Similarity (Vaze et al., 2023) This paper which emphasized the challenge of developing and evaluating models that can adapt to generic notions of similarity, introduces both a benchmark and method for solving the conditional image similarity task. The benchmark is specifically intended for zero-shot evaluation, aiming to assess models' ability to flexibly adapt to various similarity notions without fine-tuning.

GeneCIS approach to solve the conditional image similarity task, is by training over triplets of (Reference Image, Target Image, Caption). Their architecture consists of image encoder and text encoder, the two encoders output embeddings which are merged together by a 'Combiner' network which consists of combination of MLP's networks. The encoders initiated as CLIP image and text encoders and trained end to end together with the MLP's networks.

2.2 Composed Image Retrieval (CIR)

A super task to the one we are solving is the 'Zero Shot Composed Image Retrieval' (CIR) task. It involves retrieving image from a image corpus based on a reference image and textual query. We consider it super to the conditional image similarity task, because the query can also require removing or replacing objects and attributes of the reference image.

Most modern approaches such as (Alberto Baldrati, 2023) and (Pfister, 2023) solve CIR by converting the reference image into text that describes the given image. Then, combining the text that describes the reference image with the reference text query, to form a new text query which combines information about the reference image and desired output image. This new text query is encoded into shared embedding space (usually using CLIP). All of the corpus image embedding vectors are calculated, and the image with the nearest embedding vector is selected. Just like our method, training can be avoided with this approach. However, we believe that a drawback of this method is that by converting the reference image into text (usually by image encoder and text decoder), some information in the image that important to the input text query might be lost in the conversion.

3 Method

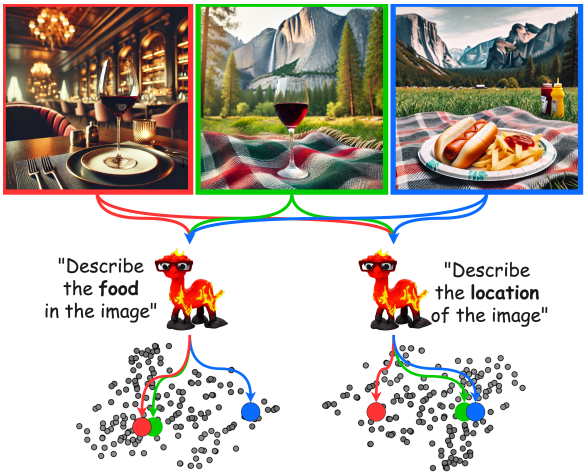

Figure 2: Our method for text conditioned image embedding. LLaVA's hidden states are being used as the text conditioned embeddings.

3.1 LLaVA hidden states as conditional image embeddings

Given a LLaVA model M, an image I, a text prompt T, token location $j \in \{1, \ldots, J\}$ where J is the number of input tokens, and layer number $l \in \{1, \ldots, L\}$ where $L = 32$ is the number of layers in the textual part of LLaVA, we can extract a hidden state $h = M(I, T; j, l)$ with $h \in \mathbb{R}^{4098}$. This allows us to obtain text-conditioned image embeddings. For example, in the task of conditioned image retrieval, one can select a fixed text T that describes the category or attribute of interest, calculate all the features of the search space, and return

the images with the highest similarity. The token location j and layer l are hyper parameters, and we will explore optimal settings for these in our experiments. Figure 3.1 illustrates our method.

3.2 Quadruplets dataset

For the purpose of optimizing and evaluating our method, we created a synthetic dataset composed of quadruplets, their definition and creation process will be described in the two following subsections. We name our dataset *QARD* which stands for Quadruplet Attribute Relationships Dataset

3.2.1 Quadruplet definition

A quadruplets of the form $(I_a, I_p, I_n, T_{a\sim p})$ corresponds to anchor image, positive image, negative image and a text instruction asking to describe a category for which I_a and I_p share a common attribute which is absent from I_n. In practice we created quintuplets (a tuple of 5 elements) of the form $(I_a, I_\gamma, I_\delta, T_{a\sim\gamma}, T_{a\sim\delta})$ which could be used to define two quadruplets: $(I_a, I_\gamma, I_\delta, T_{a\sim\gamma})$ and $(I_a, I_\delta, I_\gamma, T_{a\sim\delta})$. Figure 3 illustrates such a quintuplet.

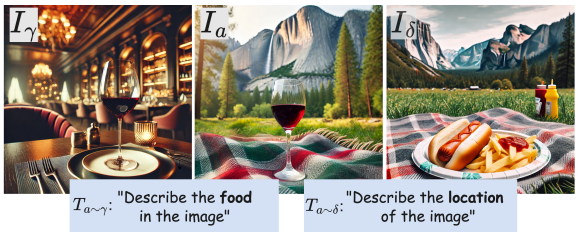

Figure 3: The anchor image I_a shares the same food category with I_γ (wine) and not with I_δ (hotdog). Similarly I_a shares the same location category with I_δ (Yosemite Park) and not with I_γ (restaurant).

3.2.2 Quadruplet dataset creation process

Figures 7 and 8 presents examples of successfully and flawed quintuplets. The quadruplets are being derived from the quintuplets, the creation of the quintuplets dataset involves a two-stage process. First, we utilized a commercial chat-bot, instructing it to generate quintuplets with text prompts for the images rather than the images themselves. This was achieved by describing the desired structure and logic of the quintuplets and providing a few examples. We found that Anthropic's *Claude 3.5 Sonnet* outperforms Open AI's *ChatGPT 4* and *4o* in this specific task. Second, we used SDXL (Rombach et al., 2021) to generate the images based on the generated text prompts. The instruction prompt for the chat-bot cab be found in the appendix A.1. The dataset contains 150 quintuplets which results in 300 quadruplets. The dataset was splitted to 100 train quadruplets (used for hyper-parameter tuning) and 200 test quadruplets.

4 Experiments

In the following subsections we will describe our experiments with LLaVA's hidden states. Similarly to a regular LLM, each hidden state corresponds to a token, while in LLaVA, the input tokens are composed both from image tokens and text tokens. The number of image tokens could be more than 1000, depends on the image size. Due to the large number of image tokens, we narrowed down our search space and focused only on the hidden states which corresponds to text tokens. Experimenting with the image token hidden states is left for future work.

4.1 Does the hidden states contain useful information about the input image?

First, we wanted to learn to what extent, the hidden states of LLaVA contains useful signal from an input image, given a general prompt such as "Describe the image". To do so, we used the Animals-10 dataset (Corrado, 2019) which is an image classification dataset, contains 10 classes of animals. A subset of the dataset with 400 training images and 100 test images was used where each of the 10 classes is equally represented. For each image in the dataset and for different hidden layers l and for different token locations j, we extracted the hidden states and trained a simple MLP classifier $C_{l,j}$ to predict the class of the image from the hidden state $h_{l,j}$. Figure 4 shows the classification accuracy calculated on the test for each hidden layer (vertical axis) and for each token location (horizontal axis). We can see the lowest layers (e.g. 0-3) are having low classification accuracy (regardless of the token locations), with layer 0 having the poorest performance due to the fact that no information at all flows to these hidden layers from the input image. Classification accuracy increases rapidly as we climb up with the layers, getting close to perfect. We can also see that different token locations are having different accuracy's with a general trend of better accuracy as we get closer to the end of the prompt. We also wanted to learn if the prompt affects the classification accuracy, therefore we repeated the experiment with the additional

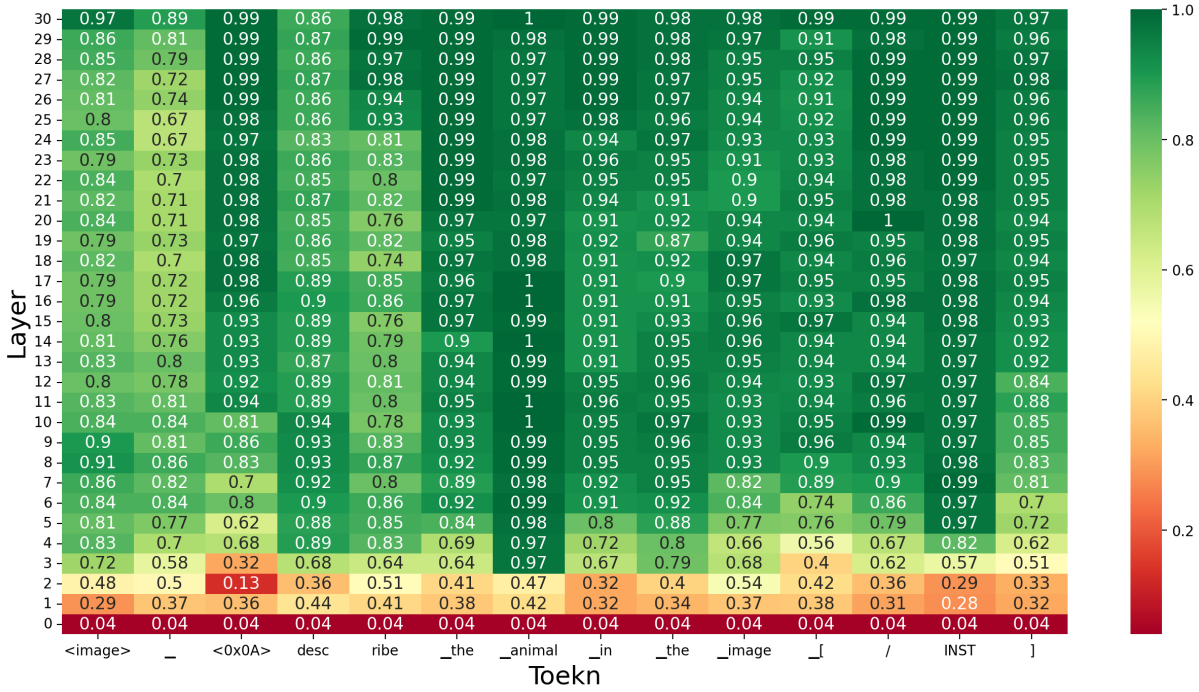

Figure 4: Classification accuracy on the test set, for hidden states from different token locations and layers.

prompt: "describe the background of image", Figure 5 shows the accuracy for different layers given the last token location for the two different prompts. We observe a slight improvement for some of the layers when using the hidden states of the more relevant prompt that instruct the network to describe the animal in the image. Even though the improvement is not very significant in this specific case, and together with the high accuracies we get, it led us to believe that the hidden states could be used as text conditional image embeddings.

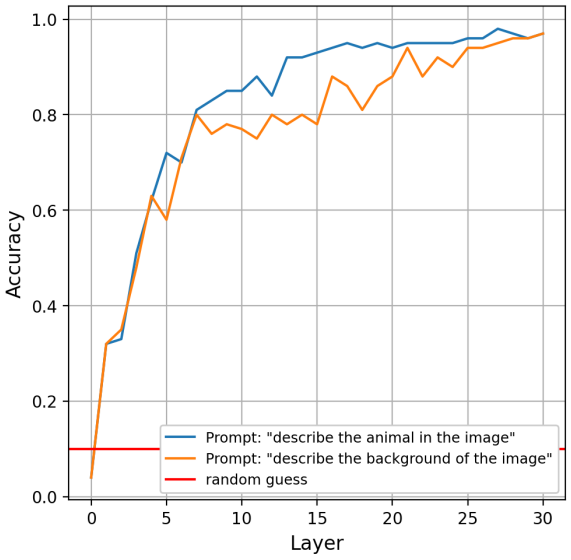

Figure 5: Classification accuracies of different hidden layers belong to the last token. The blue and orange curves are the results for the relevant and the less relevant prompts respectively.

4.2 Using hidden state as text conditioned image embeddings

As described in 3.1, hidden states are being extracted from LLaVA. For a fixed layer l and token location j, The straightforward way to asses the similarity between two images, given a condition text T, is to compute the similarity between their hidden states. For the image retrieval task we also propose a variation, where the entire search space is being indexed ahead using a generic text (e.g. "Describe the image") and only the query image hidden state is being conditioned by the text T.

4.2.1 Baseline

In order to evaluate our method we provide a baseline algorithm to which we will compare our method. Given a query image and a prompt, we generate a textual description using LLaVA, which we encode with the CLIP text encoder. Then, for a given test image, we use the CLIP image encoder to get it's encoding and calculate the similarity between the image and the textual encodings.

4.2.2 Datasets and Metrics

QARD: We initially apply our method and the baseline to the *QARD* dataset 3.2. Using the anchor image I_a and the text $T_{a\sim p}$ as the prompt, we compute the similarity scores for both the positive image I_p and the negative image I_n. The final score is determined by the ratio of successful instances (where the positive image has a higher similarity score) over the entire *QARD* dataset. We use the train split of the *QARD* dataset to select the best hidden layer and report the results on the test split.

CelebA: To evaluate the image retrieval task, we used the CelebA (Liu et al. (2015)) dataset and performed a user study. The CelebA dataset contains two hundred thousand images of faces, from which we used a subset of 1,200. We created 6 textual descriptions (to be used as the text condition), each focusing on a different attribute in the image, for example, "Describe the hair of the person in the image", then for each description we randomly chose 10 query images from the dataset. For each pair of text condition and a query image, we applied both our method and the baseline over the entire dataset (see examples in figure 9). We evaluated the results with a user study, in which we observed the top 3 images that were retrieved for each query image and text condition pair, and gave a score on a scale of 0-3 based on the number of retrieved images that matched both the query image and the text condition. For example, for the text condition "Describe the hair of the person in the image", we gave a point for every image in which the person's hair is similar to the query image's person's hair.

4.2.3 Main results

Figure 6 presents the accuracy over the *QARD* train split for different layers, optimal accuracy is achieved at layer 15, we therefore set the layer of the hidden layer hyper parameter to 15. Our main results are presented in Table 1, on our synthetic quadruplets dataset we achieved accuracy of 0.77 compared to 0.7 achieved by the baseline and to 0.66 achieved by the fast variation of our method. On the image retrieval task, where we conducted a user study, our method outperforms the baseline by a large margin. To understand why the baseline performed poorly over the CelebA task, we explored its results and intermediate text outputs and came up with two possible explanations. Firstly, we had to limit the expressiveness of the LLaVA output by adding to the prompt 'Limit your response to no more than 2 short sentences', as CLIP text encoder can only process 77 tokens. We noticed that the descriptions are very general, and sometimes repeat themselves for different query images. Secondly, we suffered from CLIP limitations such as negation, and attribute binding with few objects. For example, the embedding for the text 'small nose' was closer to the embeddings of an image where the nose covers the whole photo, rather than the embeddings of small nose of person farther away. Now, these explanations raise the question to why the baseline worked better over the quadruplets task. Our explanation to this gap lays in the observation that due to the creation process of the QARD dataset, it is possible to describe the similarity between the anchor image and the positive image only in a few words. While in the image retrieval task on CelebA, positive examples should have more subtle common features with the query image, which requires more expressiveness which is absent in the baseline, while our hidden states can encode subtle visual information more effectively.

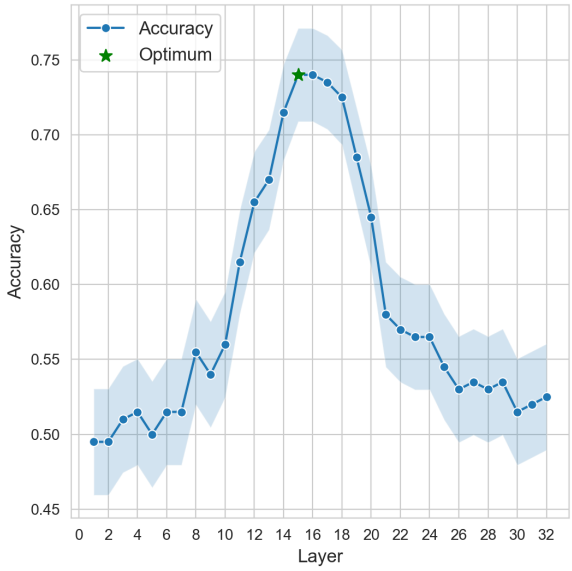

Figure 6: Accuracy on the quadruplets dataset for different layers calculated on the training set. Layer 15 was found to be optimal. the hidden state of the last token is being used.

5 Limitations

A big limitation in our method is that its very computation intensive. For large corpus, running Llava over all images can take significant amount of time. It is also required to run Llava over all corpus for

Table 1: Comparison of our method to the baseline on the quadruplets and the image retrieval tasks. The score for the quadruplets task is the success ratio of choosing the positive sample.

Method	Quadruplets task ↑	Image Retrieval ↑	#Inference ↓
Baseline	0.70	0.32	1
Ours	**0.77**	**0.74**	$1 + \|SearchSpace\|$
Ours - constant prompt for search space	0.66	-	1

each new text query. Most methods today that solve the 'composed image retrieval' task, require running image to text inversion only for the reference image, and they get the corpus embeddings via one time pre-process for as many queries desired with a simple image encoder (such as CLIP) over the corpus.

6 Future Work

In this paper, we did not explore the hidden representations belonging to the image tokens. Future work may find it worthwhile to investigate these hidden representations.

7 Conclusions

Exploiting LLaVA's hidden layers seems to be a promising direction for text conditional image embedding which could be used for various tasks, without the need to train task specific models. While we achieved accuracy of 0.77 on our synthetic quadruplets dataset, it should be taken into account that our dataset consists a non-negligible amount of flawed data points, therefore our achieved accuracy is a lower bound. Our findings with regard to the optimal hidden state, which is an intermediate layer rather than a final layer aligns with the finding of (Ghandeharioun et al., 2024) that shows that intermediate layers process and contain useful information while final layers are optimized to predict the next token.

References

Guillaume Alain and Yoshua Bengio. 2018. Understanding intermediate layers using linear classifier probes. 2018. *arXiv preprint arXiv:1610.01644*.

Marco Bertini Alberto del Bimbo Alberto Baldrati, Lorenzo Agnolucci. 2023. Zero-shot composed image retrieval with textual inversion. *ICCV 2023*.

Alessio Corrado. 2019. Animals-10 dataset. https://www.kaggle.com/datasets/alessiocorrado99/animals10.

Asma Ghandeharioun, Avi Caciularu, Adam Pearce, Lucas Dixon, and Mor Geva. 2024. Patchscope: A unifying framework for inspecting hidden representations of language models. *arXiv preprint arXiv:2401.06102*.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. In *NeurIPS*.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. 2015. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*.

Kuniaki Saito Kihyuk Sohn Xiang Zhang Chun-Liang Chen-Yu Lee Kate Saenko Tomas Pfister. 2023. Pic2word: Mapping pictures to words for zero-shot composed image retrieval. *arXiv 2302.03084*.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2021. High-resolution image synthesis with latent diffusion models.

Sagar Vaze, Nicolas Carion, and Ishan Misra. 2023. Genecis: A benchmark for general conditional image similarity. In *CVPR*.

A Appendix - Additional details

A.1 Text prompt instruction used to create the quintuplet dataset

Following is the prompt that was used to generate the quintuplets:

" Your mission today is to generate data points composed of 5-element quintuplets: Prompt anchor: A prompt that describes the anchor image. Prompt gamma: A prompt that describes the gamma image. Prompt delta: A prompt that describes the delta image. Text anchor-gamma: A text related to the commonality between the anchor and gamma image, which is not shared with the delta image. Text anchor-delta: A text related to the commonality between the anchor and delta image, which is not shared with the gamma image.

Example 1: Prompt anchor: A full image of a very tall man with blonde hair. Prompt gamma: A full image of a tall man with gray hair. Prompt y: A full image of a short man with blonde hair. Text

anchor-gamma: Describe the height of the person in the image. Text anchor-delta: Describe the hair color of the person in the image.

Example 2: Prompt anchor: A blue convertible parked near a beach. Prompt gamma: A red sports car driving on a highway. Prompt delta: A blue bicycle leaning against a tree in nature. Text anchor-gamma: Describe the type of vehicle in the image. Text anchor-delta: Describe the color of the vehicle in the image.

Example 3: Prompt anchor: An image of a glass of red wine on a picnic blanket in Yosemite park. Prompt gamma: An image of a glass of red wine on a table in a fancy restaurant. Prompt y: An image of a hotdog on a picnic blanket in Yosemite park. Text anchor-gamma: Describe the drink in the image. Text anchor-delta: Describe the location of the image. Additional instructions:

The rule: for each prompt in [gamma, delta], ancor-prompt should define a category of which they share a common attribute described in their prompts. Feel free to be creative and use your imagination to generate the prompts. Ensure the shared attributes between 'anchor' and 'gamma' are not shared with 'delta', and vice versa. Each text (anchor-gamma and anchor-delta) must clearly relate to the shared attribute and distinctly separate it from the non-shared image. Write down a short explanation for each quintuplet. Output your results in a list of dictionaries with keys 'anchor', 'gamma', 'delta', 'anchor-gamma', 'anchor-delta', 'explanation'. The commonality between the images should be clear, easy to understand, and not implicit. Generate 20 quintuplets. "

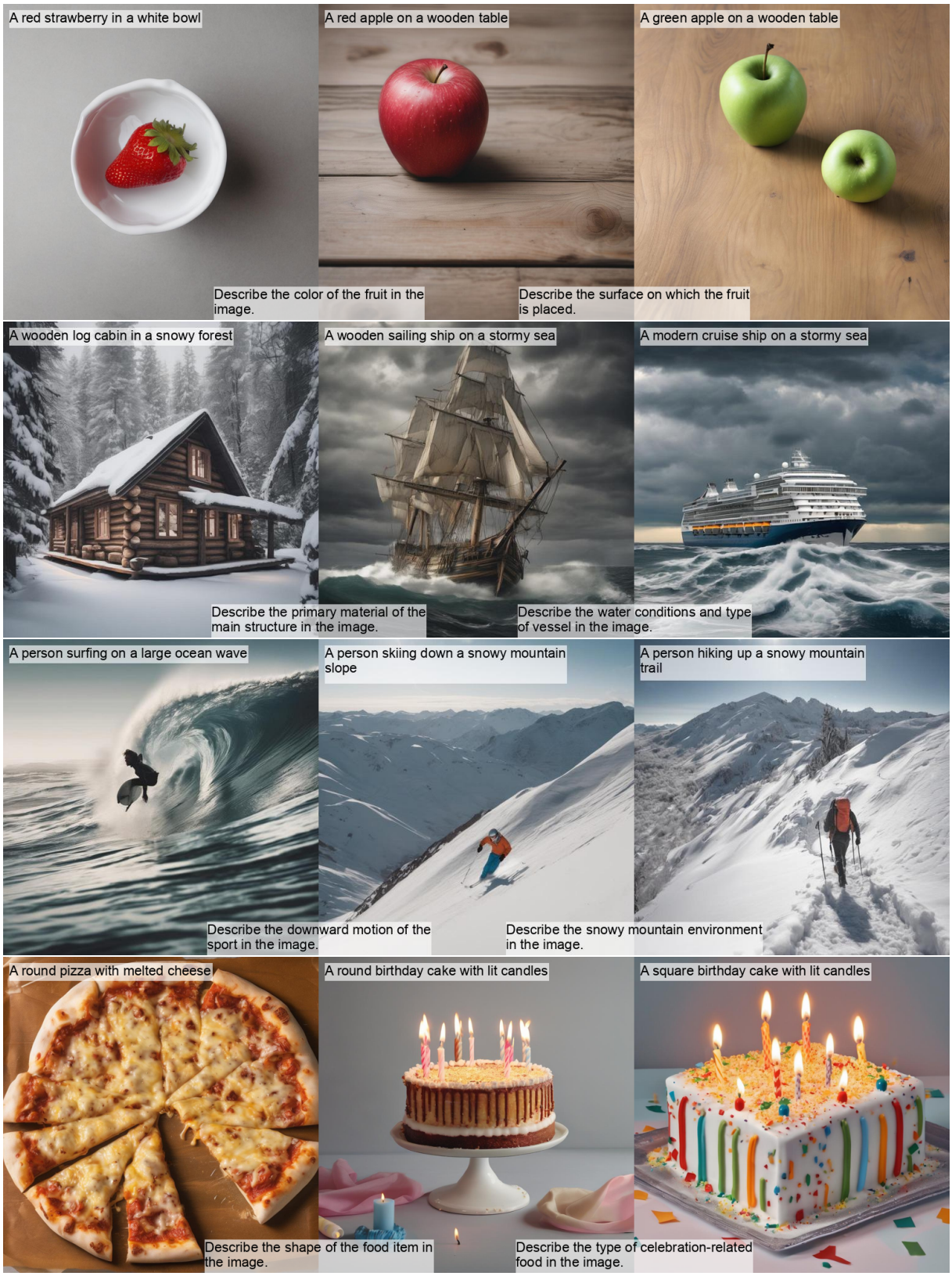

Figure 7: Samples of successfully generated data points

Figure 8: Samples of unsuccessfully generated quintuplet:
Top: The text-to-image model didn't comply to the prompt of the anchor image - the bowl isn't steaming
Second: The LLM didn't capture the professional activity being performed in the anchor image (cooking pasta) and created a description of a painter creating a portrait instead (left image).
Third: The text-to-image model created a person wearing a spacesuit instead of casual clothes (right image).
Bottom: The LLM didn't capture the type of camera in the anchor image (vintage Polaroid) and created a description depicting a modern digital camera (right image).

Figure 9: Samples of the image retrieval task: On the left is the query image, following by the top 5 images we retrieved. For each sample, the images in the top row were retrieved by our method, and the images in the bottom row were retrieved by the baseline using the following text conditions:
First sample's text condition: "Describe the expression of the person in the image"
Second sample's text condition: "Describe the hair of the person in the image"
Third sample's text condition: "Look on the lips of the person in the image"