

hw3_r13921068

DLCV HW3 Report

學號：r13921068

姓名：吳家萱

Problem 1: Zero-shot Image Captioning with LLaVA

1. Paper reading - Please read the paper "Visual Instruction Tuning" and briefly describe the important components (modules or techniques) of LLaVA.
 - LLaVA (大型語言和視覺助理) 是一種用於視覺和語言理解的大型多模態模型，他結合了一個 Vision Encoder 和一個大型語言模型 (LLM)，結構如下：
 - Vision Encoder:
 - LLaVA 使用 CLIP 的視覺編碼器 (ViT-L/14)，將輸入的圖像轉換為視覺特徵 (visual features)，論文中使用了最後一層 Transformer 之前和之後的網格特徵 (grid features)
 - Projection Layer:
 - 為了將圖像特徵連接到 LLM 的詞嵌入空間 (word embedding space) 中，LLaVA 使用一個簡單的線性層，這個層會將視覺特徵 (Z_v) 轉換為語言嵌入標記 (H_v)，其維度與 LLM 中的詞嵌入空間相同，也就是把圖像特徵轉換成 LLM 能理解的語言
 - Language Model:
 - LLaVA 選擇 Vicuna 作為它的 LLM ($f_\phi(\cdot)$)，因為 Vicuna 在公開可用的模型檢查點中具有最佳的語言指令遵循能力，我理解成他是 LLaVA 的「大腦」，負責理解和生成語言
 - Multimodal instruction-following data:
 - 由於缺乏視覺-語言指令遵循數據，LLaVA 的訓練利用 ChatGPT/GPT-4 將圖像-文本對轉換成適當的指令遵循格式，透過使用圖像的標題和邊界框作為 GPT-4 的提示、讓 GPT-4 生成三種類型的指令遵循數據：對話、詳細描述和複雜推理、將生成的數據用於 LLaVA 的微調，來達到數據重組
 - Two-stage instruction tuning:
 - LLaVA 採用兩階段指令微調過程來訓練模型:
 1. Pre-training for Feature Alignment:
讓 Vision Encoder 和 LLM 的權重都被凍結，只有投射矩陣 W 是可訓練的，模型在過濾後的 CC3M 數據集上進行預訓練，目的是將圖像特徵與預先訓練好的 LLM 詞嵌入對齊，我的理解是在教 LLaVA 如何「看懂」圖像
 2. Fine-tuning End-to-End:
讓 Vision Encoder 的權重保持凍結，而投射層和 LLM 的權重都進行微調，目標是進一步提高模型的指令遵循能力

• 總結：

- LLaVA 主要利用現有的圖像-文本配對數據，透過多模態指令遵循數據來進行訓練，尤其是在模型架構方面，LLaVA 使用 CLIP 的視覺編碼器提取圖像特徵，並透過一個線性投射層將其轉換為語言嵌入標記
- 訓練過程分為兩個階段：
 1. 特徵對齊預訓練階段，凍結 Vision Encoder 和 LLM 的權重，僅訓練Projection Layer，以對齊圖像特徵與語言模型的詞嵌入空間
 2. End-to-End Fine-tuning 階段，凍結 Vision Encoder 的權重，微調Projection Layer 和語言模型的權重，以進一步提升模型的指令遵循能力
- 2. Prompt-text analysis - Please come up with two settings (different instructions or generation config). Compare and discuss their performances.
- 我擁有最好的 performance 設計為：
- instruction:

*"Write a single sentence that describing the main action or the main content in the image, focusing on who or what is in the scene and the action what they are doing. Examples:
A man in a baseball game running to base and others trying to tag him out.
A box with half a dozen glazed and frosted donuts.
A person at a table is eating a small pizza."*

- generation_config:

max_new_tokens	do_sample	temperature	num_beams
50	False	0.7	2

在此設計中，我的分數可以分別達到以下成績：

CIDEr	CLIPScore
1.2077657265782578	0.7760260009765625

- 另外，我也實驗過當我的 instruction 沒有給予 caption example 時，我的 performance 就無法足夠好，如下設定：
- instruction:

"Provide a direct description of the main subject and what they are doing. Focus on what they are doing, do not mention their appearance, clothing, or colors. Avoid stating unnecessary details like 'the main subject is'. Be concise and focus on the action."

- 在 generation config 中，我嘗試過的方法如下：
 - 如果 do_sample 為 True，雖然可以增加 sampling 的多樣性，但跑的時間會過久，因此維持 False
 - temperature 可以增加創意性，我嘗試過 0.1~0.7，最後覺得 0.7 最合適
 - num beams 我試過 2~7，雖然 beams 的數量越大可以有越多候選人，但大概只有 2 或 3 時有機會在 30 分鐘內跑完，其餘都會超過
- generation_config:

max_new_tokens	do_sample	temperature	num_beams
50	False	0.1	2

在此設計中，我的分數分別為以下成績：

CIDEr	CLIPScore
1.0772888361500528	0.7530999755859376

- 總結：

- 在嘗試各種不同設計的排列組合後，我認為 instruction 是最為關鍵的，尤其是要在 instruction 提及生成的範例，以及需要特別說明 “Write a single sentence that describing the main action or the main content in the image”，否則就會生成較多無關緊要的描述，導致成績下降
-

Problem 2: PEFT on Vision and Language Model for Image Captioning

- Report your best setting and its corresponding CIDEr & CLIPScore on the validation data.
Briefly introduce your method. (TA will reproduce this result)

- 我擁有最好的 performance 設計為：
- Encoder: timm - vit_huge_patch14_clip_quickgelu_224
- Decoder projection layer:
 - nn.Linear(1280, cfg.n_embd),
nn.ReLU(),
nn.Linear(cfg.n_embd, cfg.n_embd)
- Training parameters:

epoch	optimizer	scheduler	criterion
7	AdamW	torch.optim.lr_scheduler.OneCycleLR	nn.CrossEntropyLoss
	weight_decay=1e-2	max_lr=4e-3, pct_start=0.7, anneal_strategy='cos'	ignore_index=-100, label_smoothing=0.1
batch size	trainable params	transform	LoRA rank
16	9870336	timm.data.create_transform(**data_config, is_training=False)	56

- 在此設計中，我的 validation 分數可以分別達到以下成績：

CIDEr	CLIPScore
1.0951861762670705	0.7374256896972656

- Report 2 different attempts of LoRA setting (e.g. initialization, alpha, rank...) and their corresponding CIDEr & CLIPScore.

- 在其他設計維持不變，只調整 LoRA rank 時，當 $r = 1$ ，performance 如下：

CIDEr	CLIPScore
0.9211345289020344	0.7192660522460937

- 在其他設計維持不變，只調整 LoRA rank 時，當 $r = 28$ ，performance 如下：

CIDEr	CLIPScore
1.0073739734843061	0.7346510314941406

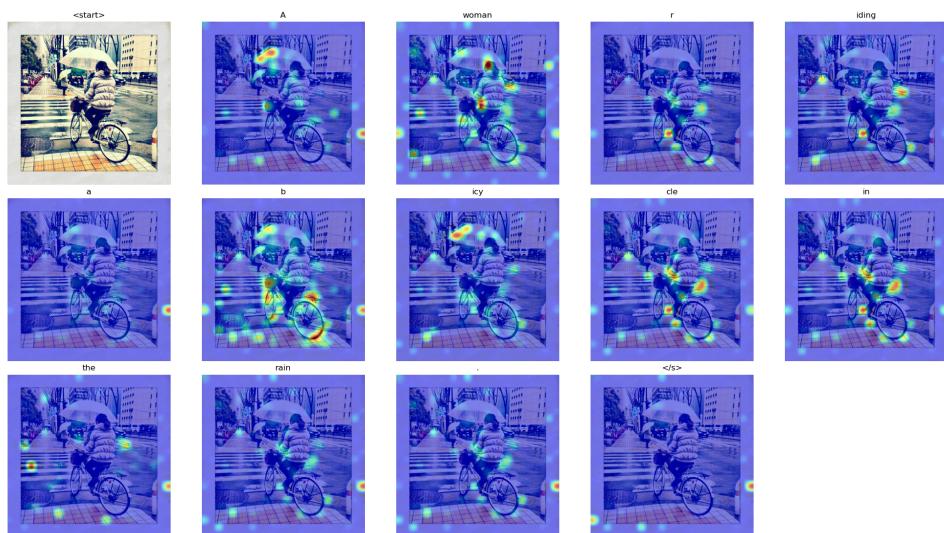
- 總結：

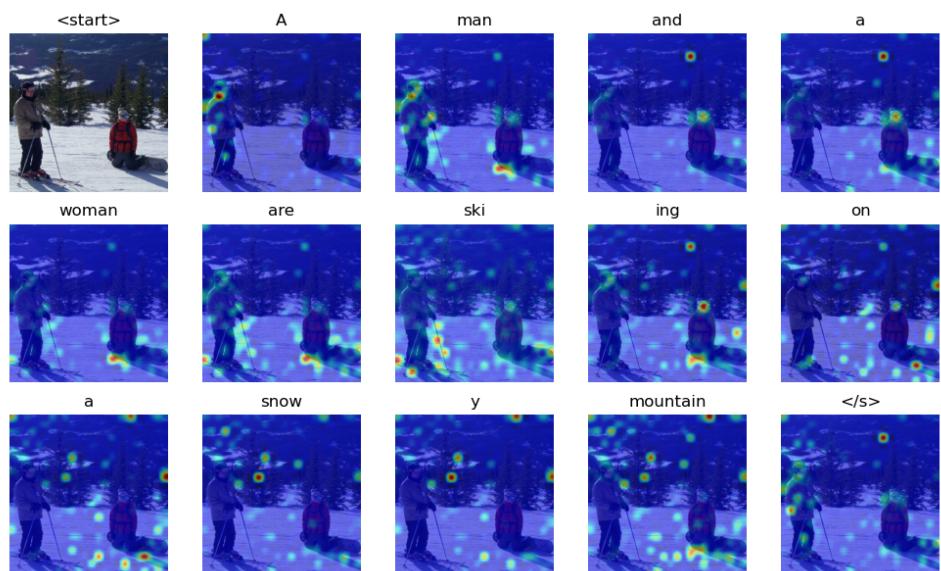
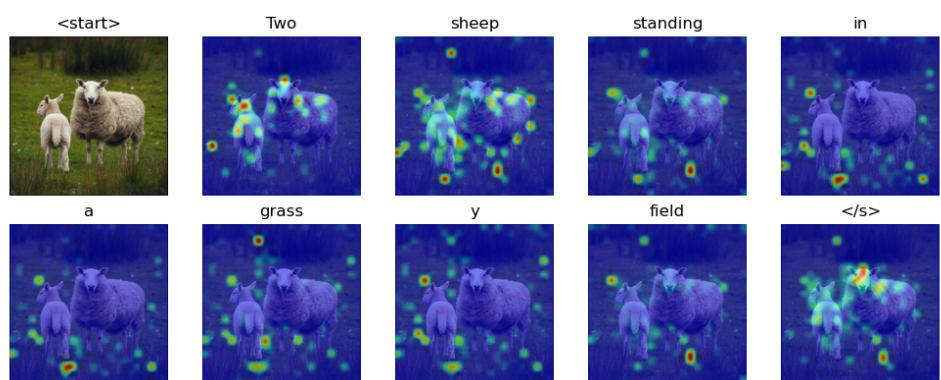
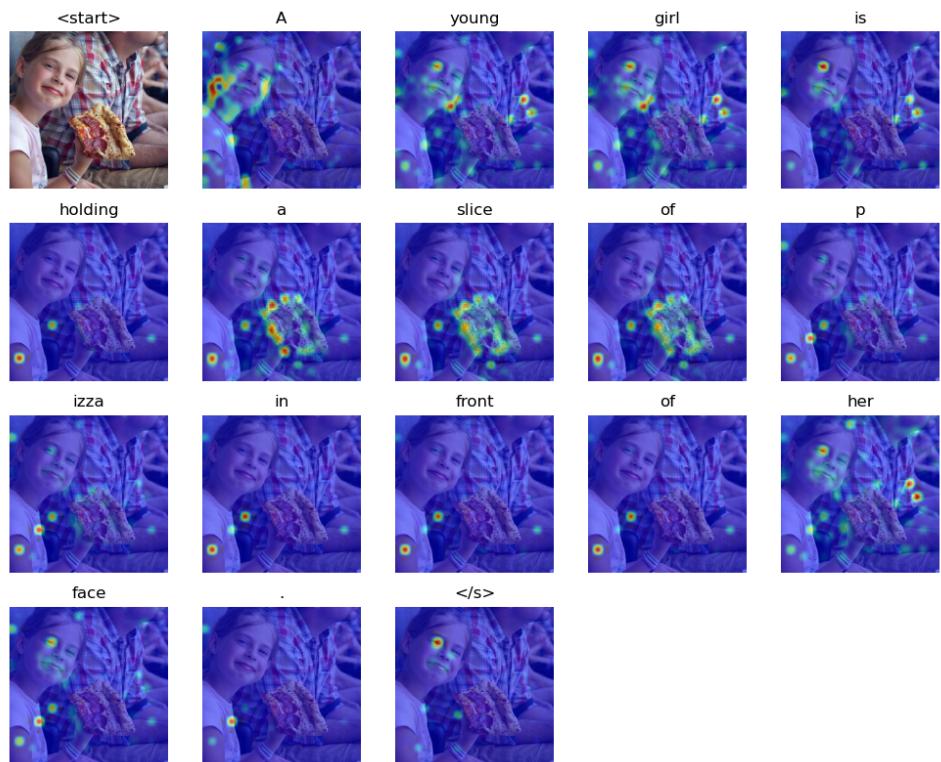
- 在其他參數保持不變的情況下，LoRA rank 越大，參數量越多 (在 10M 以下)，performance 會有所提升

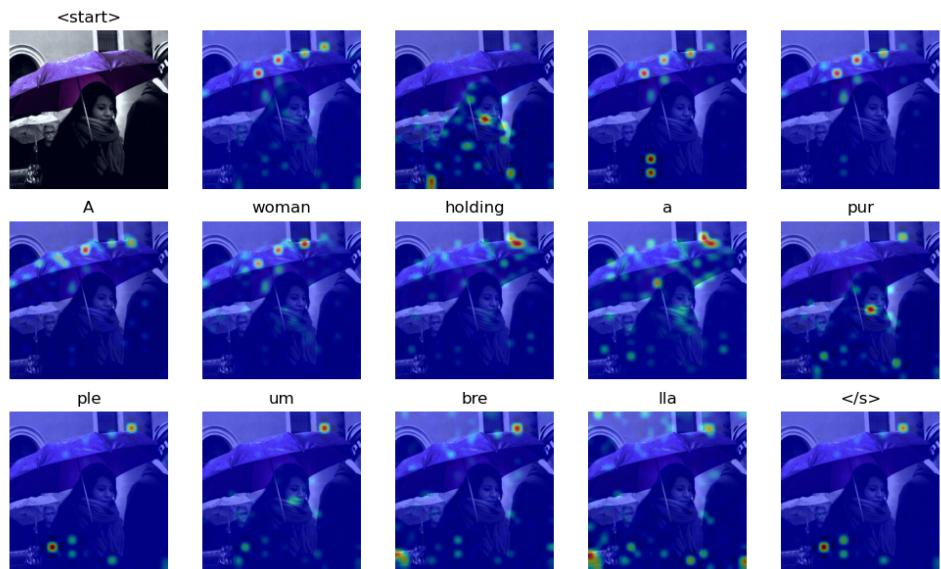
Problem 3: Visualization of Attention in Image Captioning

- Given five test images ([p3_data/images/]), and please visualize the predicted caption and the corresponding series of attention maps in your report with the following template:

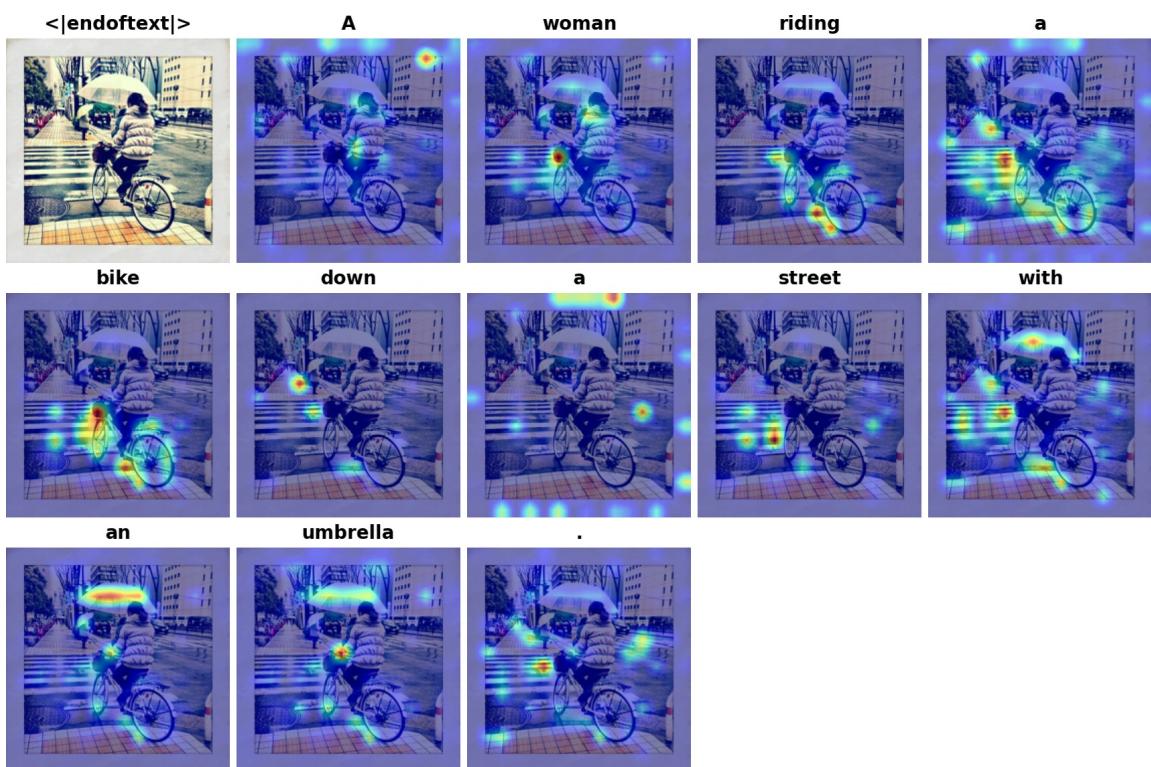
 - 在 Problem 1 中，我的 visualization 如下：

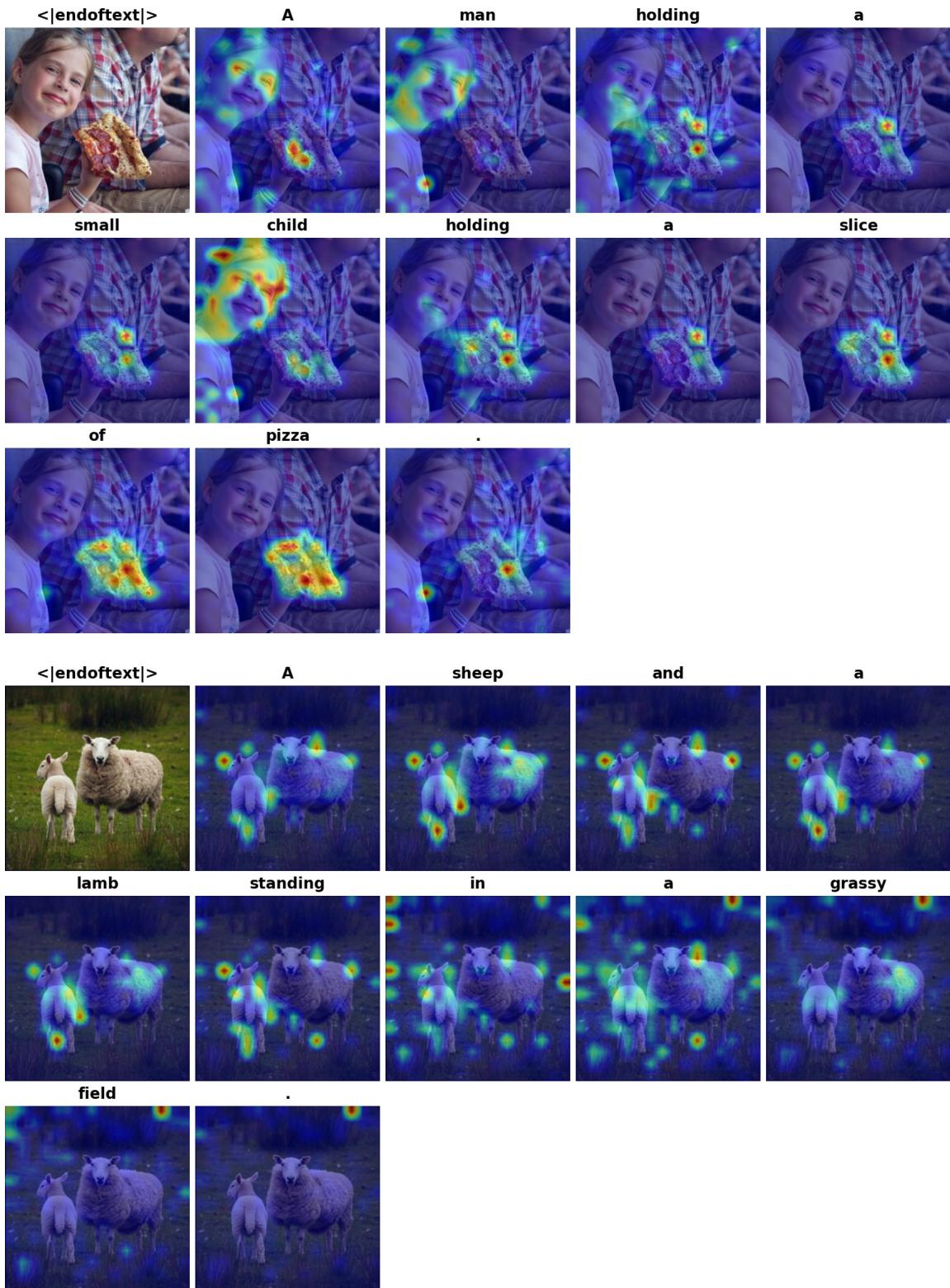


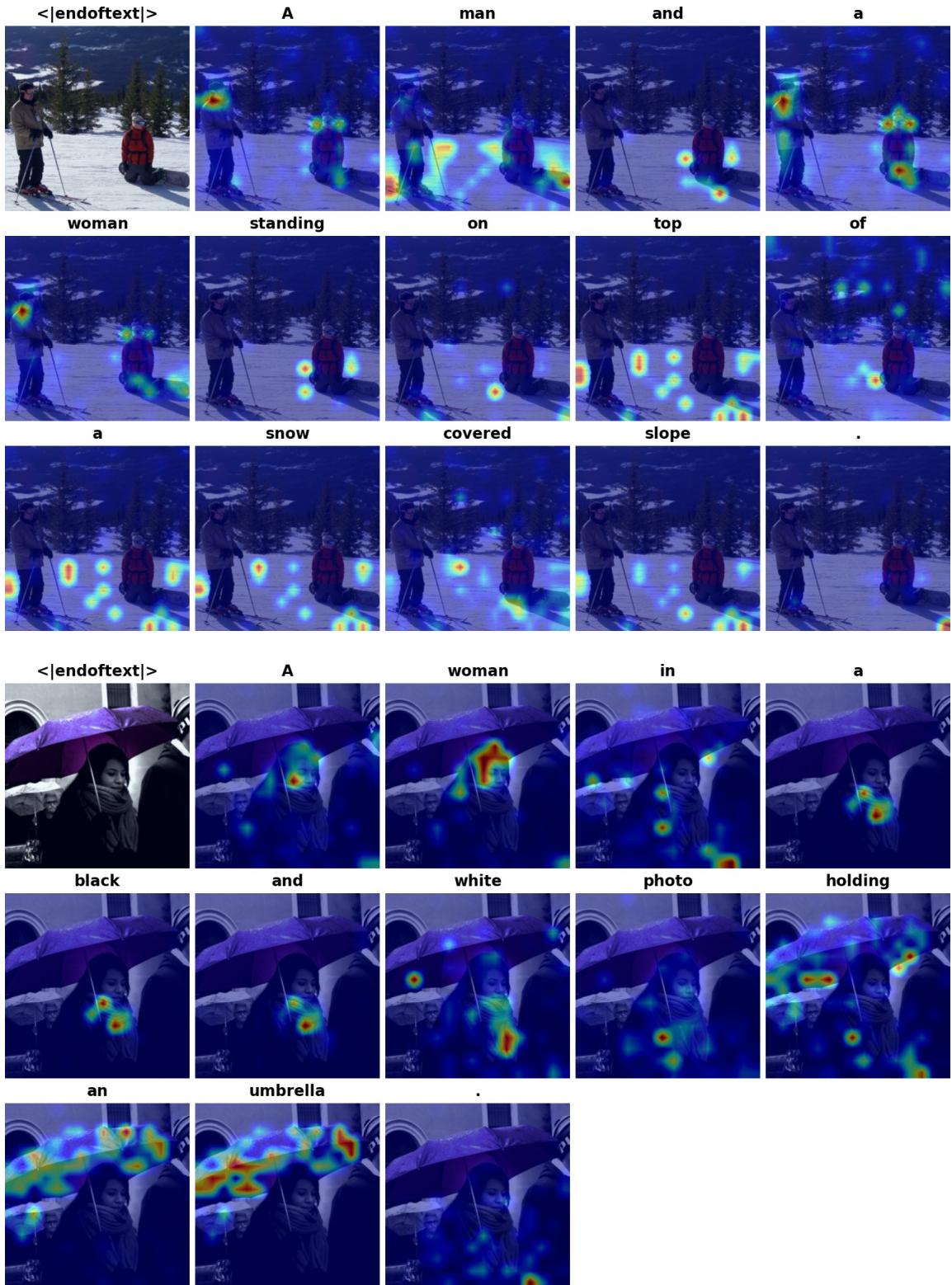




- 在 Problem 2 中，我的 visualization 如下：



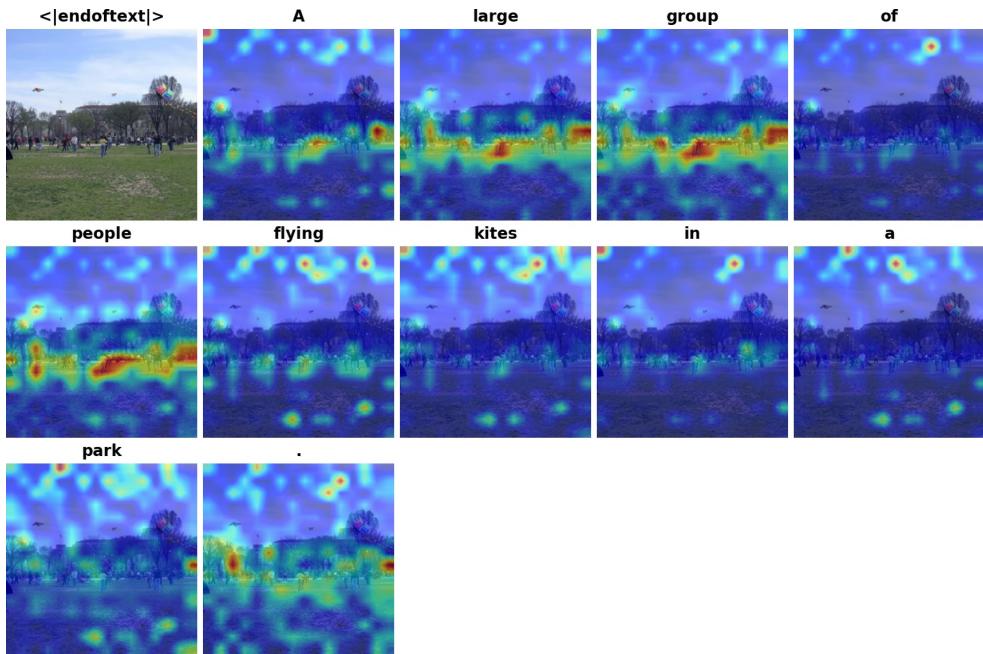




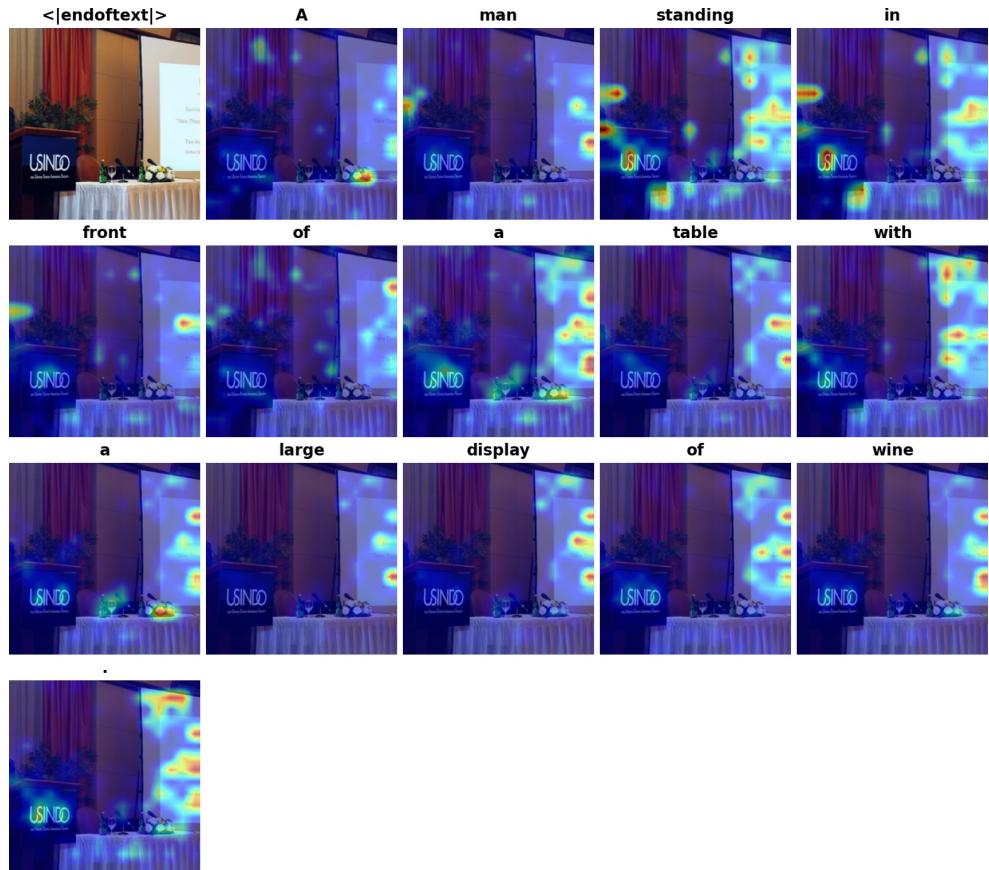
2. According to CLIPScore, you need to:
 - i. visualize top-1 and last-1 image-caption pairs
 - ii. report its corresponding CLIPScore
- 我的 top-1 & last-1 image-caption 分別為：

top-1	last-1
000000001086.jpg	000000001523.jpg
CLIPScore	CLIPScore
1.06201171875	0.42205810546875

- top-1 000000001086.jpg visualization :



- last-1 000000001523.jpg visualization :



2. Analyze the predicted captions and the attention maps for each word according to the previous question. Is the caption reasonable? Does the attended region reflect the corresponding word in the caption?

- 在 top-1 000000001086.jpg 中，我覺得生成的 caption 十分合理，這張圖的 caption 為：“*A large group of people flying kites in a park.*”，原圖的場景確實呈現了在公園空地上有多人放風箏，而 attended region 也都有抓到 caption 的一些重要 token，例如 “large”、“group”、“people” 都有聚焦在圖片中人群聚集的區域
- 在 last-1 000000001523.jpg 中，我覺得生成的 caption 差強人意，這張圖的 caption 為：“*A man standing in front of a table with a large display of wine*”，但可以明顯觀察到，原圖中並沒有出現人物，而 attended region 也沒有抓到 caption 的 token，其大部分的熱圖都聚集在投影幕上，然而 caption 中並沒有提及任何有關投影幕的 token word

Reference

[llava-hf/llava-1.5-7b-hf huggingface sample code](#)

[Visual Instruction Tuning Paper](#)

[LoRA: Low-Rank Adaptation of Large Language Models Github](#)

[The timm \(PyTorch Image Models\) Leaderboard](#)

[GenerationConfig huggingface](#)

[Processors huggingface](#)

Claude 3.5 Sonnet

GPT o1