Deep Learning for Computer Vision

HW3

電子所 ICS 組, R13943015, 張根齊

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.

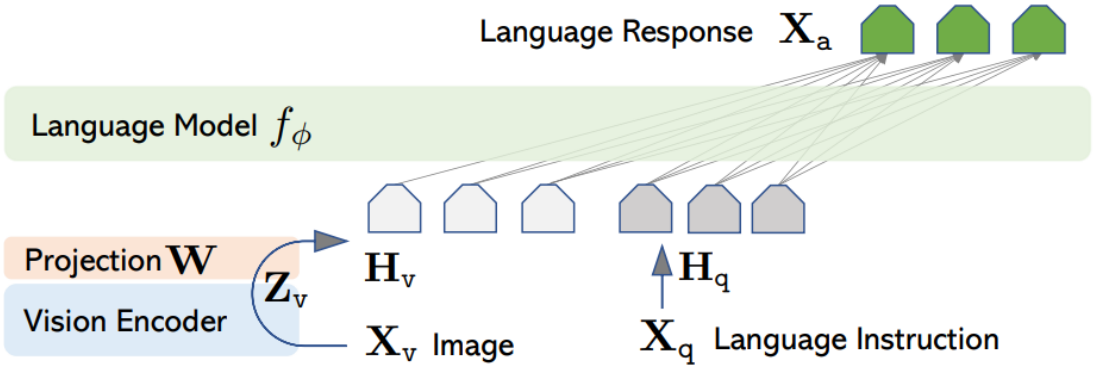


Figure 1: LLaVA network architecture.

LLaVA is a multimodal model that combines visual and textual information using a vision encoder (ViT) and a language model (LLaMA). An input image Xv is processed through the vision encoder to obtain visual features Zv, which are then passed through a projection layer W to generate the image embedding Hv. And a language instruction Xq is converted into text embeddings Hq using a tokenizer. The language model then combines the visual embedding Hv with the text embedding Hq to generate a language response Xa. Additionally, LLaVA uses multimodal instruction tuning by using GPT-4 to transform datasets into paired data (e.g., VQA pairs) and fine-tunes on these pairs to enhance its instruction-following capabilities.

2. Prompt-text analysis: Please come up with two settings (different instructions or generation config). Compare and discuss their performances.

Settings	Max token	Instructions	Gen config	Time(on L4)	CIDEr	CLIPScore
A	40	A short image caption.	Beam = 5	37 min	1.181	0.779
B	20	A short image caption.	Beam = 5	35 min	1.195	0.776
C	25	A short image caption.	Beam = 3	25 min	1.152	0.787
D	25	A image caption.	Beam = 3	29 min	0.718	0.793
E	20	A short image caption.	Beam = 3	21 min	1.119	0.775
F	20	A short image caption.	topK=10, topP=0.9 temperature=0.7	11 min	1.000	0.765

From comparing Set A with Set B and Set C with Set E, we can observe that increasing the maximum token length impacts inference time; generating more tokens leads to longer inference times. Comparing Set B with Set E reveals that a higher beam width can improve performance but also results in increased inference time. And from Sets E and F, we notice that using methods like top-k sampling and top-p sampling achieves faster inference times, but this comes at the cost of significantly reduced performance. From Set C and Set D, we can observe that using instruction without “short” cause model generate long caption, which results in poor performance and long inference time.

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

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

1. Best setting and its corresponding CIDEr & CLIPScore:

Pretrained vision encoder	
vit_large_patch14_clip_224.laion2b (https://huggingface.co/timm/vit_large_patch14_clip_224.laion2b_ft_in12k_in1k)	
Projection layer	
`self.projection_layer = nn.Linear(1024, 768)`	
transform	

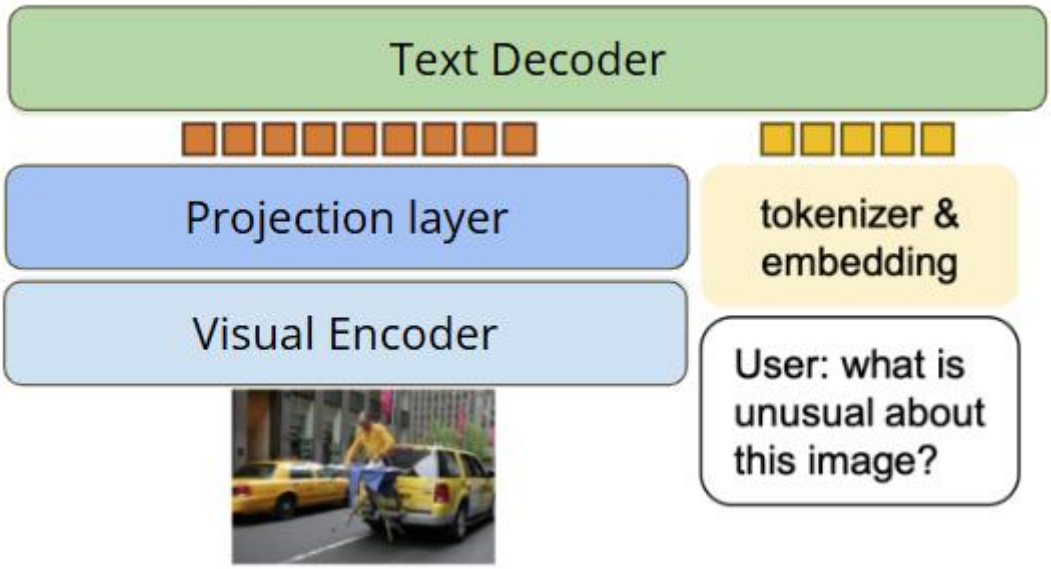
`train_transform = transforms.Compose([`
`  transforms.RandomHorizontalFlip(p=0.5),`
`  transforms.RandomResizedCrop((vision_encoder_size, vision_encoder_size), scale=(0.7, 1.0)),`
`  transforms.RandomRotation(30),`
`  transforms.ToTensor(),`
`  transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225])`
`])`	

Epoch, Loss, Optimizer, lr scheduler	

`# number of epoch`
`num_epochs = 10`
`# CE`
`criterion = nn.CrossEntropyLoss(ignore_index=PAD_GT_TOKEN)`
`# Initialize optimizer`
`optimizer = optim.Adam(model.parameters(), lr=0.004, weight_decay=1e-6)`
`# Different base learning rate and update strategy #!!!!!!`
`scheduler = CosineAnnealingLR(optimizer, T_max=num_epochs * len(train_loader), eta_min = 0)`	

Lora rank	
16	
Generation config	
greedy	
CIDEr	CLIPScore
0.9680648947073769	0.7323910522460938

2. method:



First, the input image is passed through the visual encoder to obtain the image features. Then, these features are passed through a projection layer to reshape it's dimensions to match the token embedding dimension, resulting in the image embedding. The input text is tokenized to obtain token embeddings. Next, the image embedding and token embeddings are concatenated and fed into the text decoder (language model). Finally, the output of the text decoder is obtained, but only the text part is used.

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

Using "vit_large_patch14_clip_224.openai_ft_in12k_in1k" as vit pretrained model.

Using greedy autoregressive inference method.

Result at epoch 5:

Lora rank	CIDEr	CLIPScore
16	0.9131250745714828	0.7201461791992188
32	0.9085339789333721	0.7233885192871093

For low lora rank, it's CIDEr score is higher, but it's CLIPScore is lower.

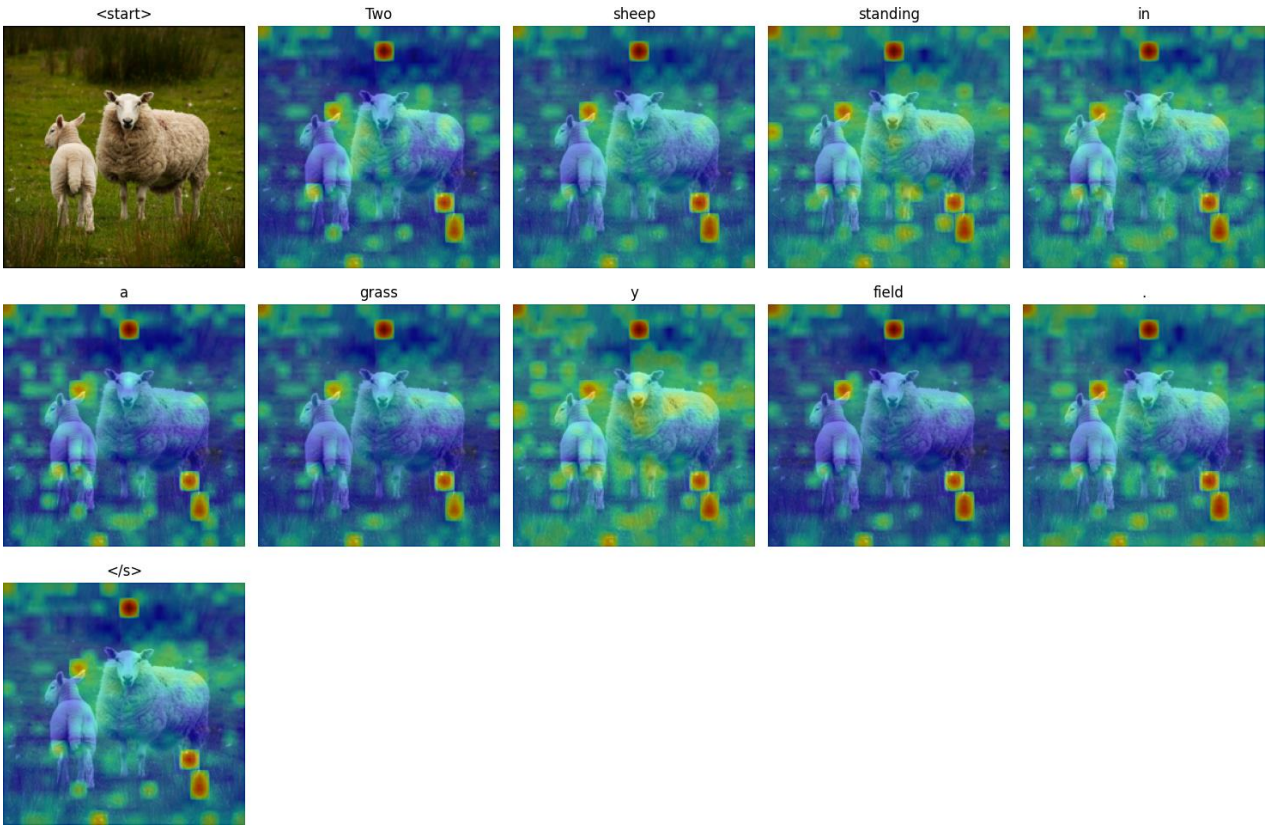
Problem 3: Visualization of Attention in Image Captioning

1. 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:

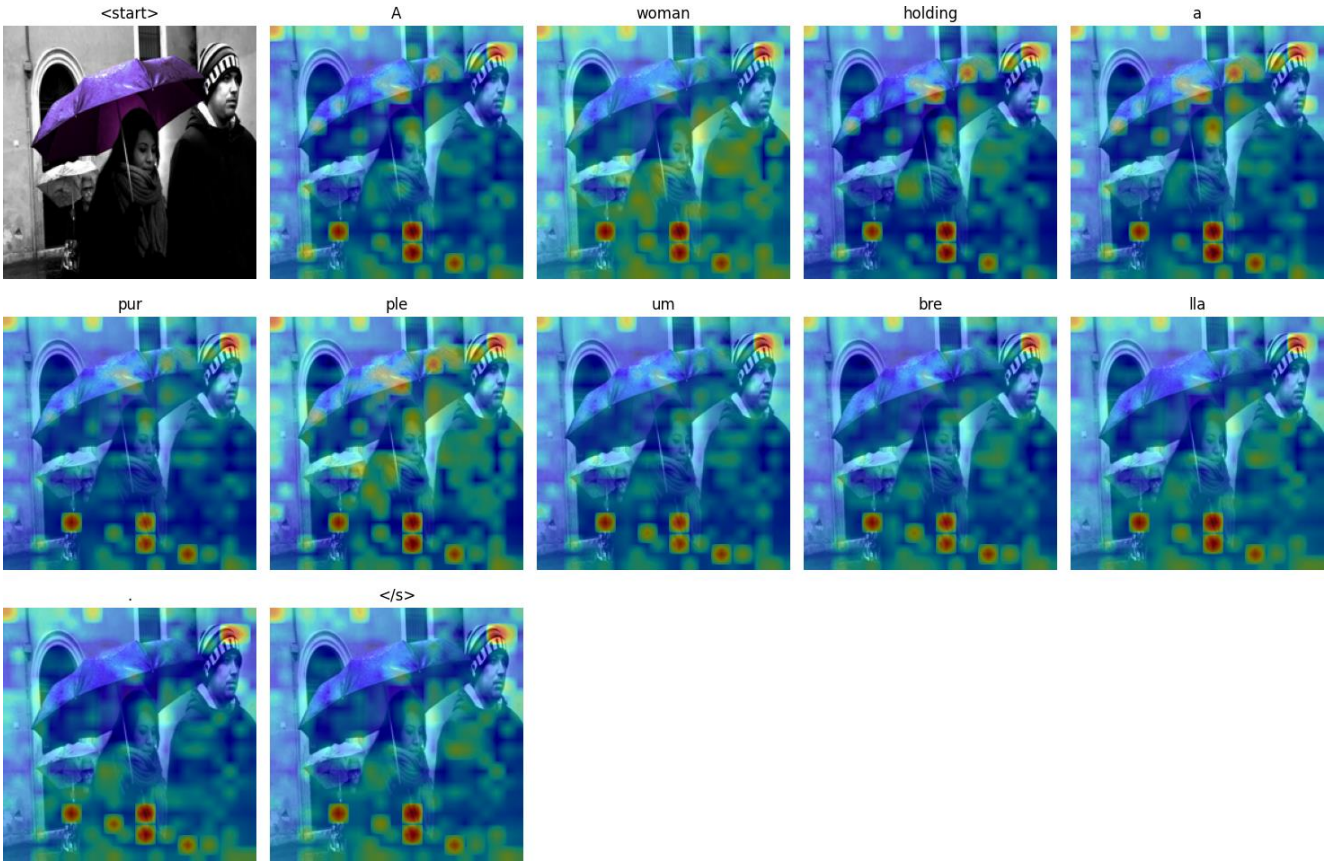
(you need to visualize 5 images for both problem 1 & 2)

Problem 1:

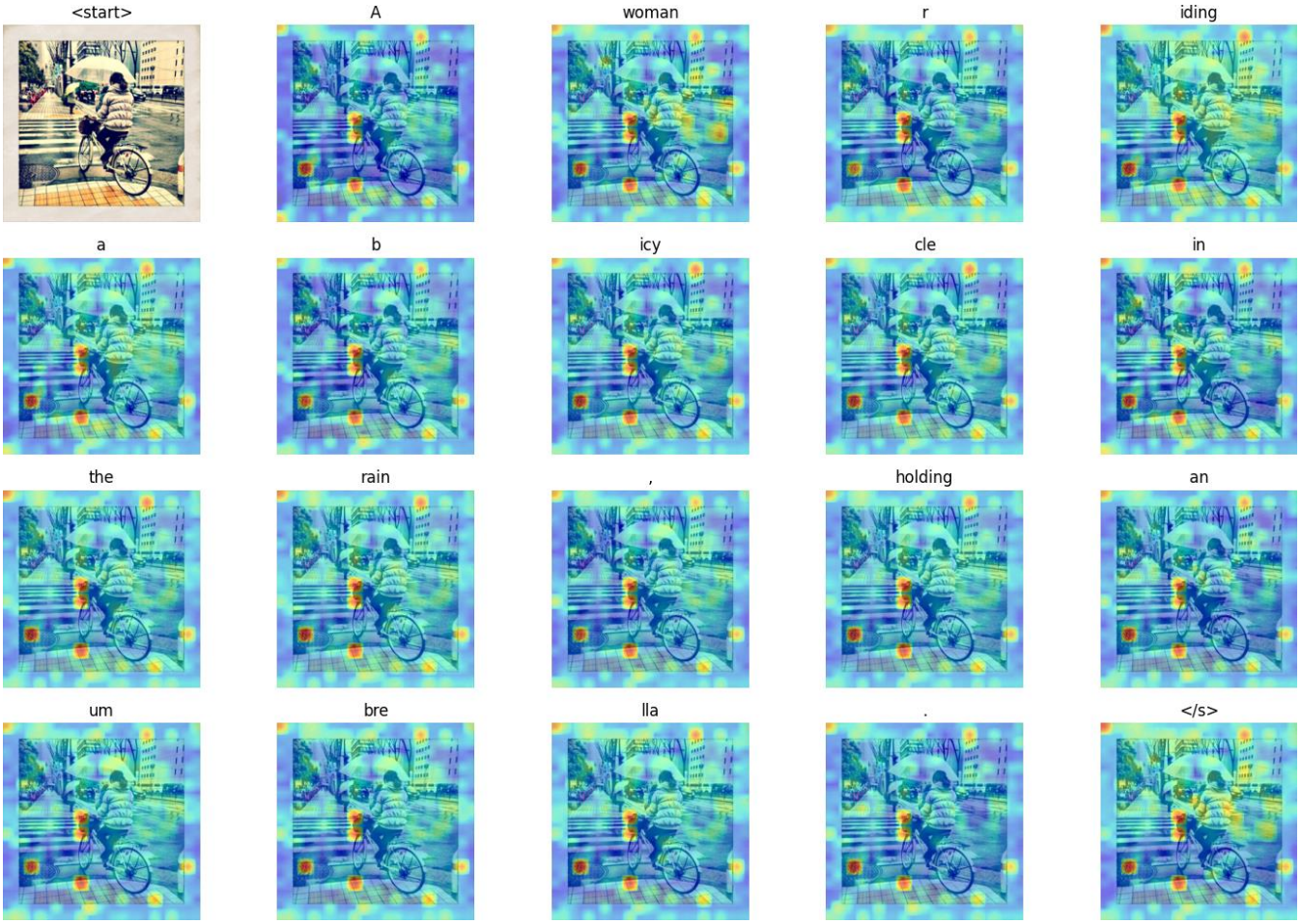
- Sheep:



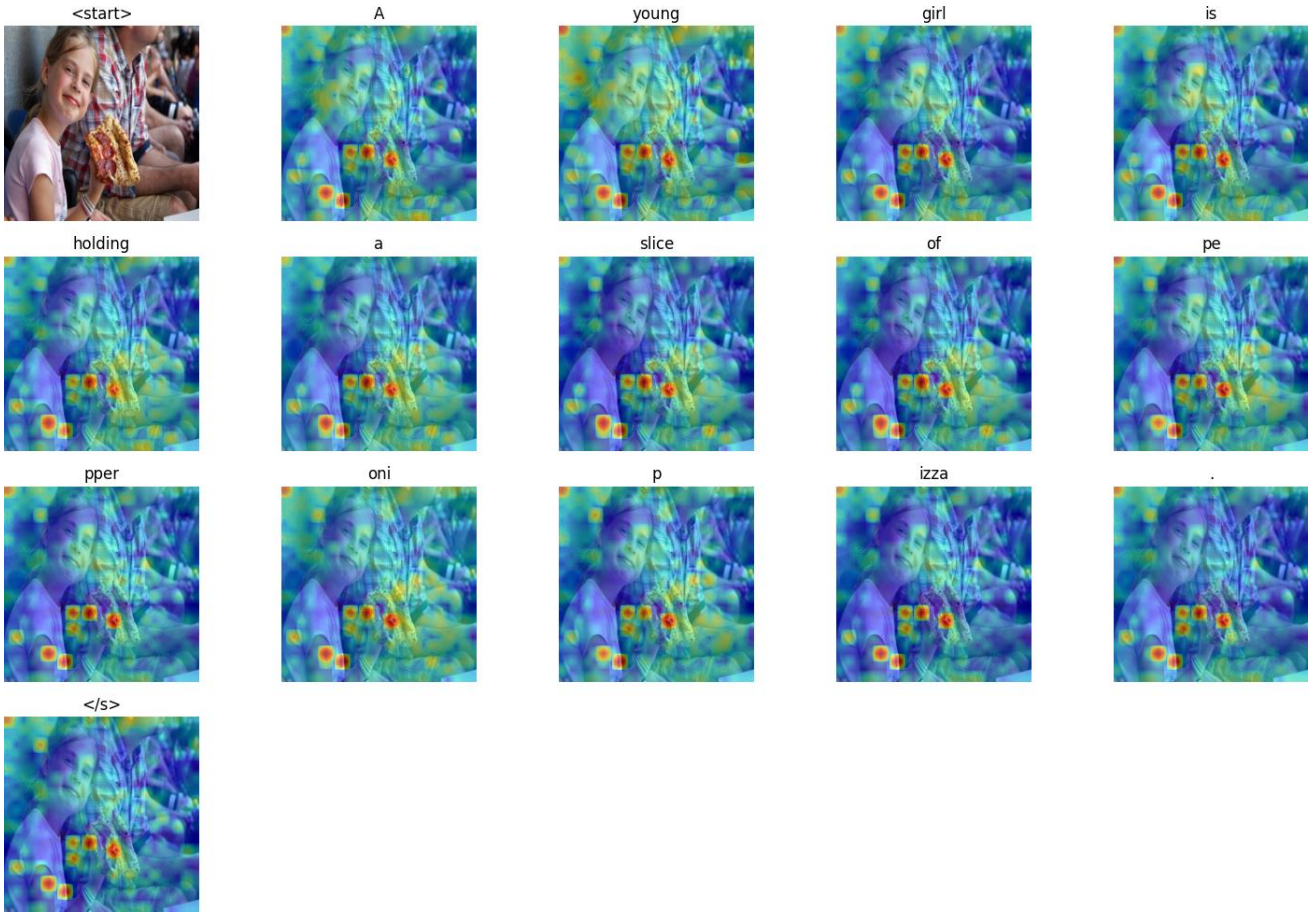
- Umbrella:


<start>
A
woman
holding
a
pur
ple
um
bre
lla
.
</s>

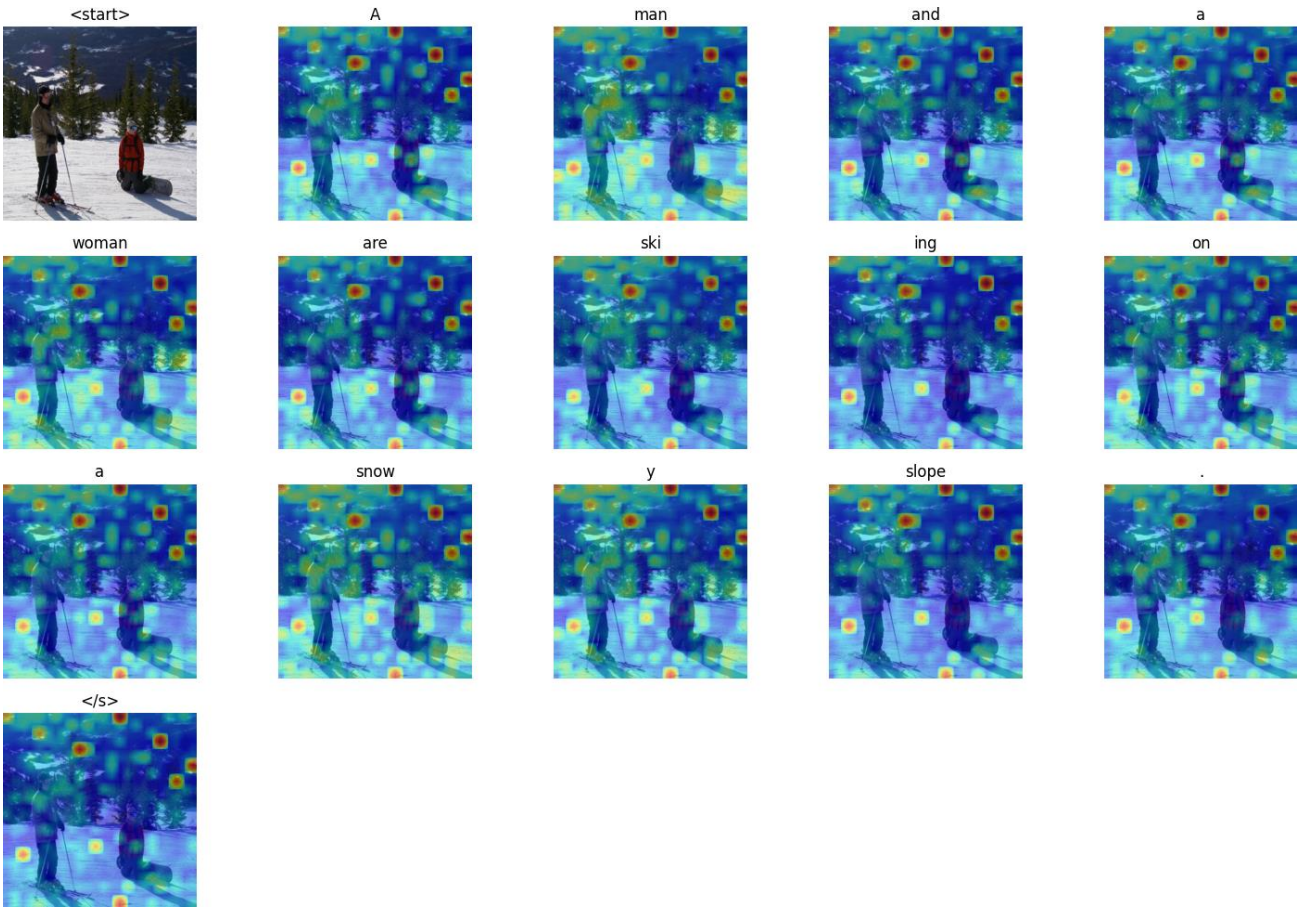
- Bike:


<start>
A
woman
r
iding
a
b
icy
cle
in
the
rain
,
holding
an
um
bre
lla
.
</s>

- Girl:

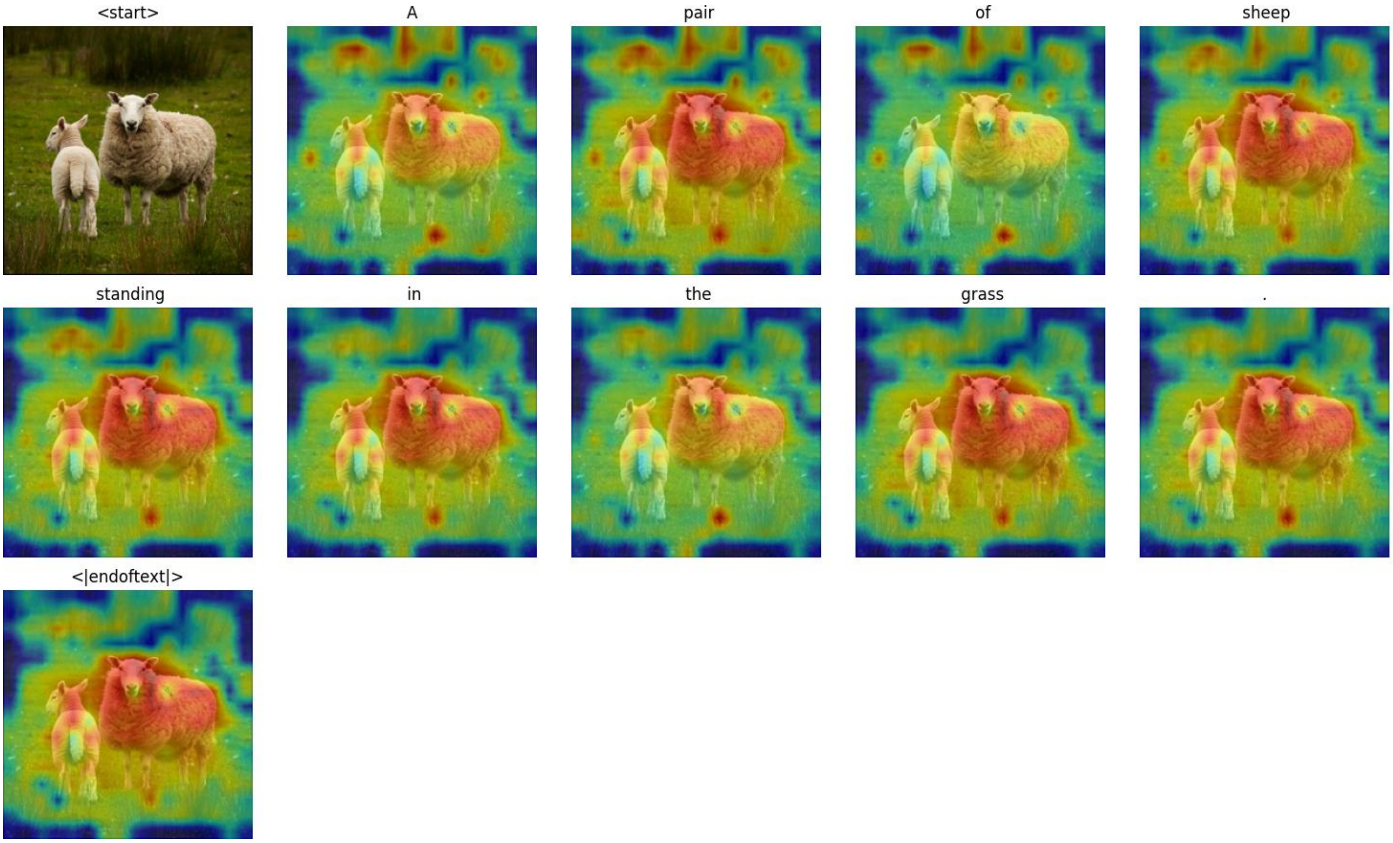

<start>
A
young
girl
is
holding
a
slice
of
pe
pper
oni
p
izza
.
</s>

- Ski:

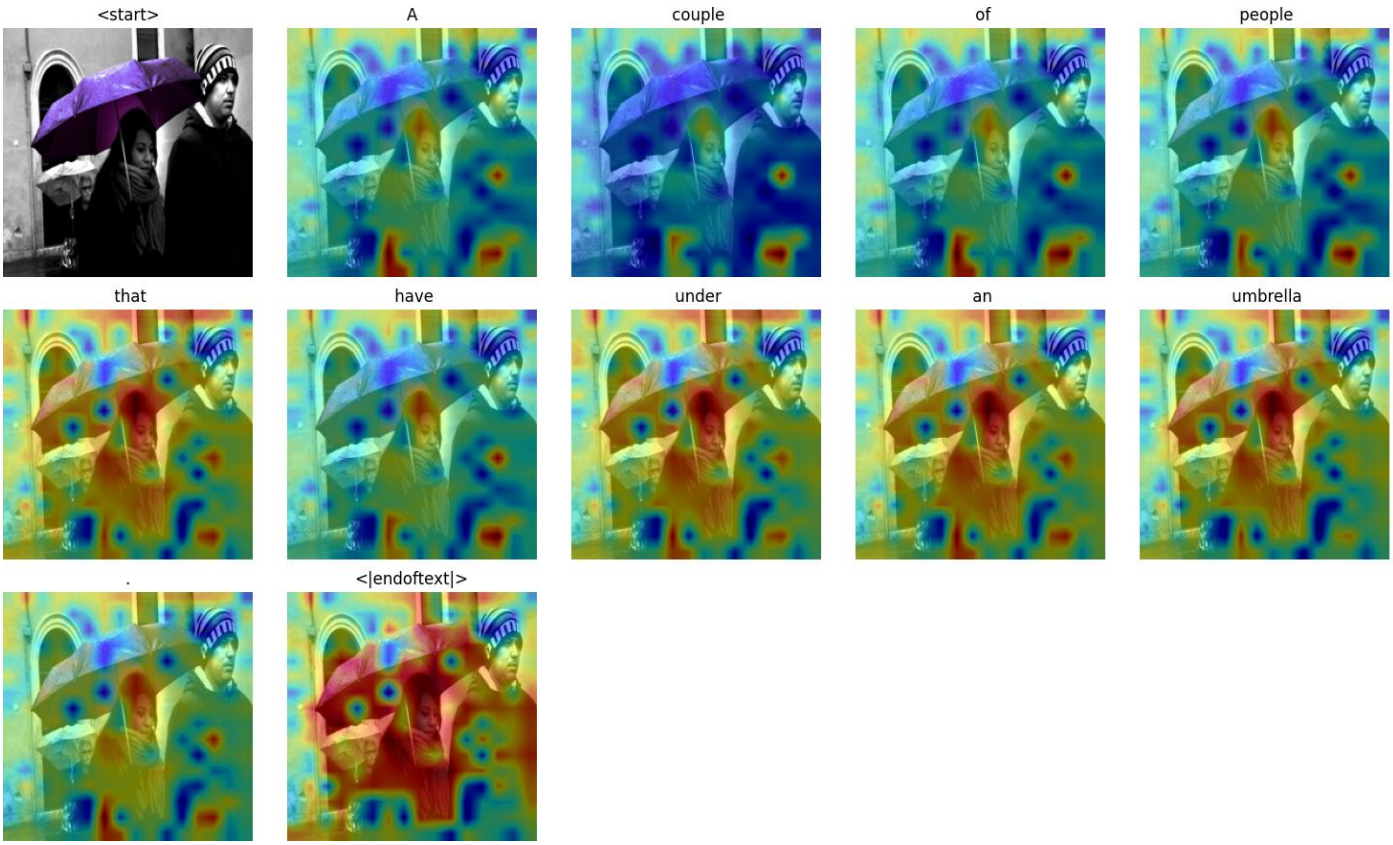

<start>
A
man
and
a
woman
are
ski
ing
on
a
snow
y
slope
.
</s>

Problem 2:

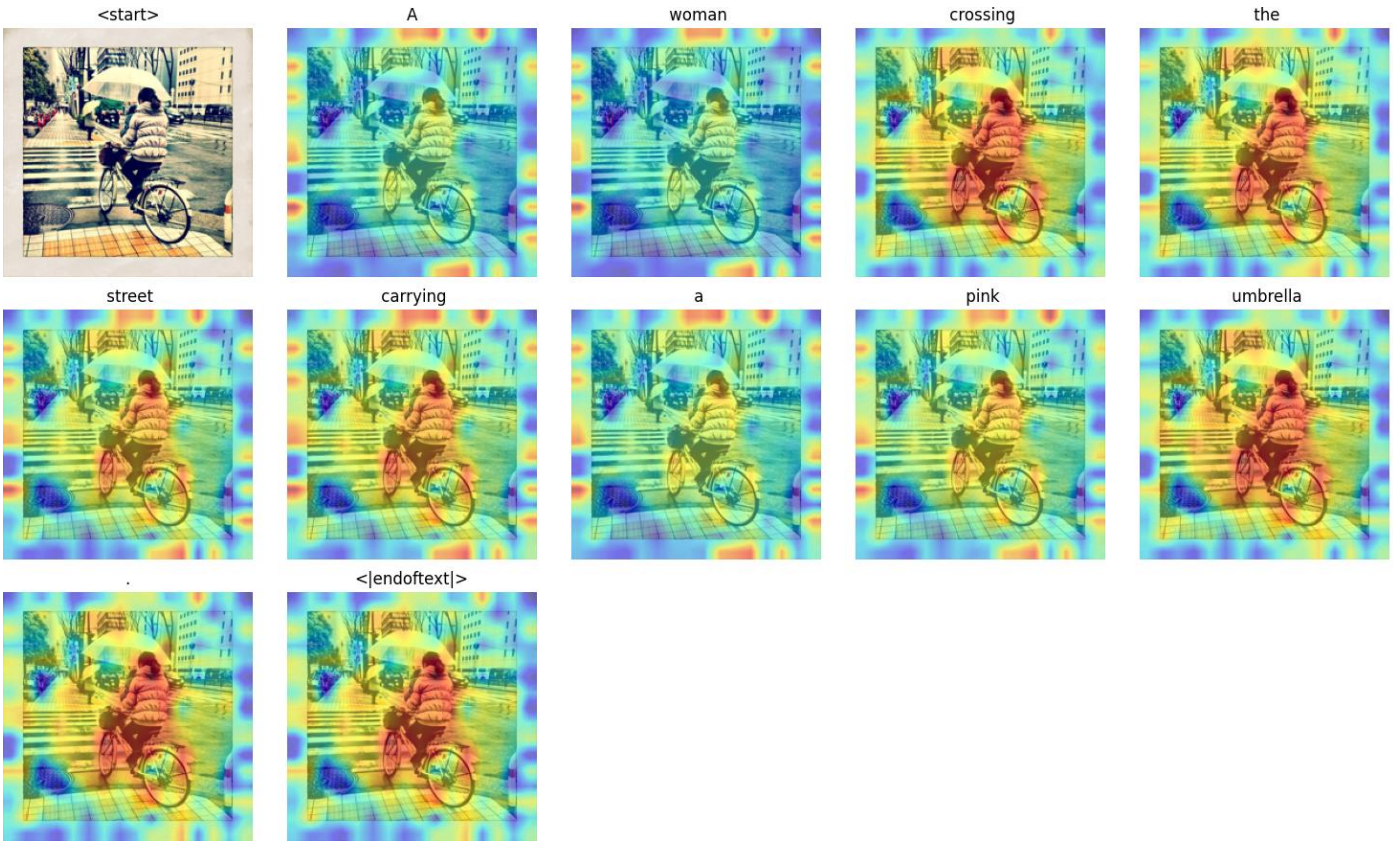
- Sheep:



- Umbrella:



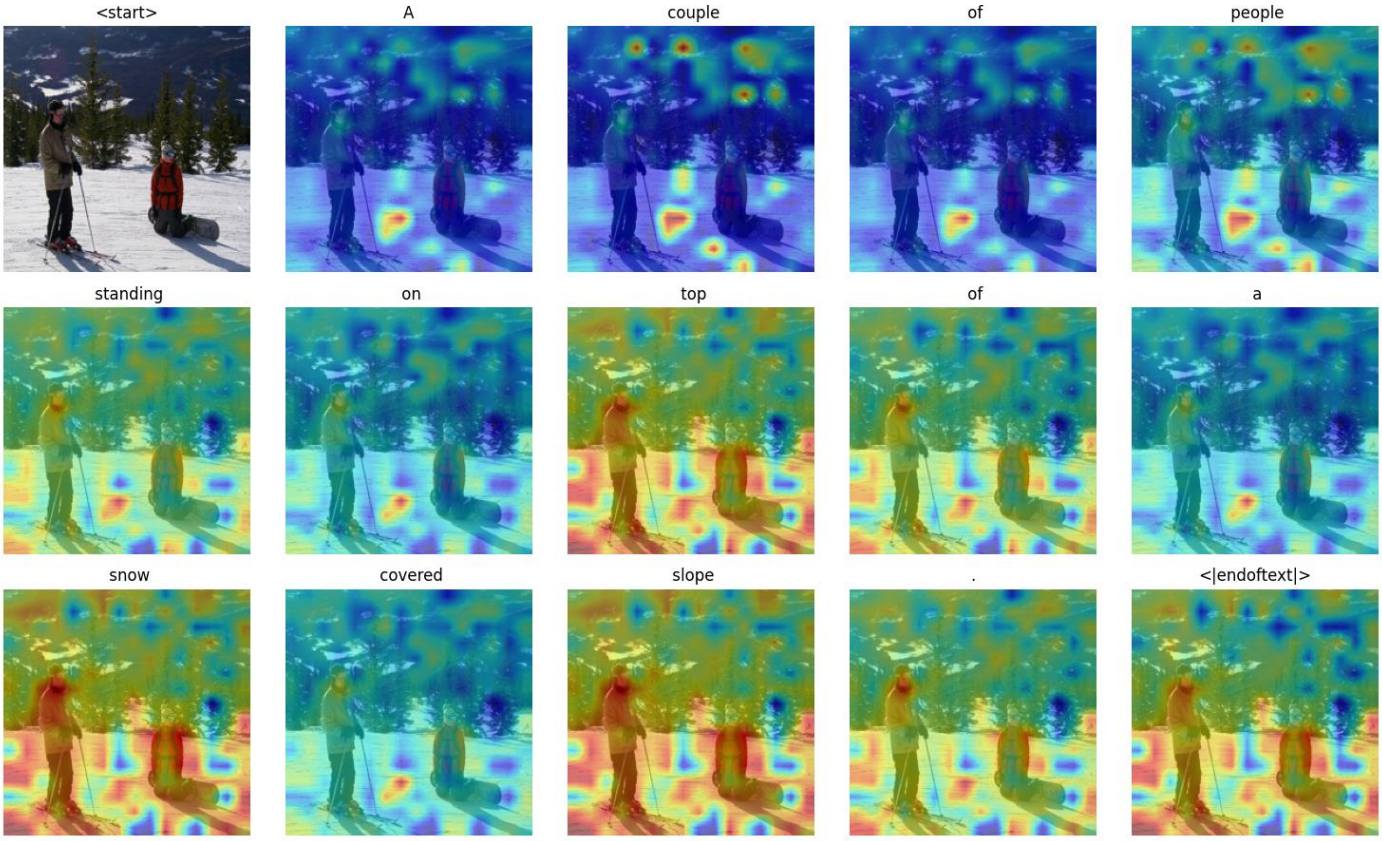
- Bike:


<start>
A
woman
crossing
the
street
carrying
a
pink
umbrella
.
<|endoftext|>

- Girl:


<start>
A
little
girl
that
is
holding
a
slice
of
pizza
.
<|endoftext|>

- Ski:


<start>
A
couple
of
people
standing
on
top
of
a
snow
covered
slope
.
<|endoftext|>

2. According to CLIPScore, you need to:

   i. visualize top-1 and last-1 image-caption pairs

   ii. report its corresponding CLIPScore

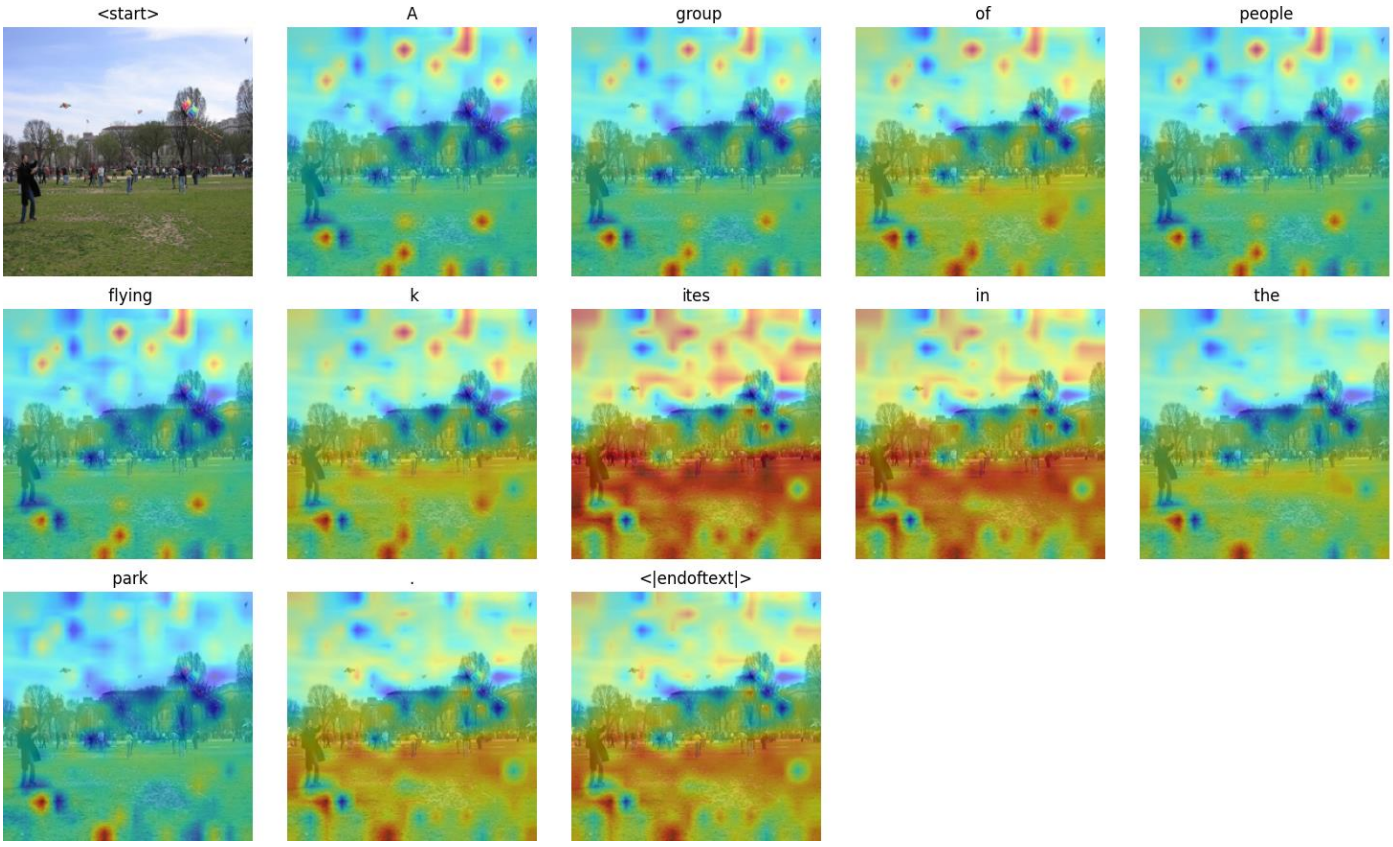
in the validation dataset of problem 2.

Top1:

Image: 000000001086.jpg

Caption: A group of people flying k ites in the park .

CLIPScore: 1.0382

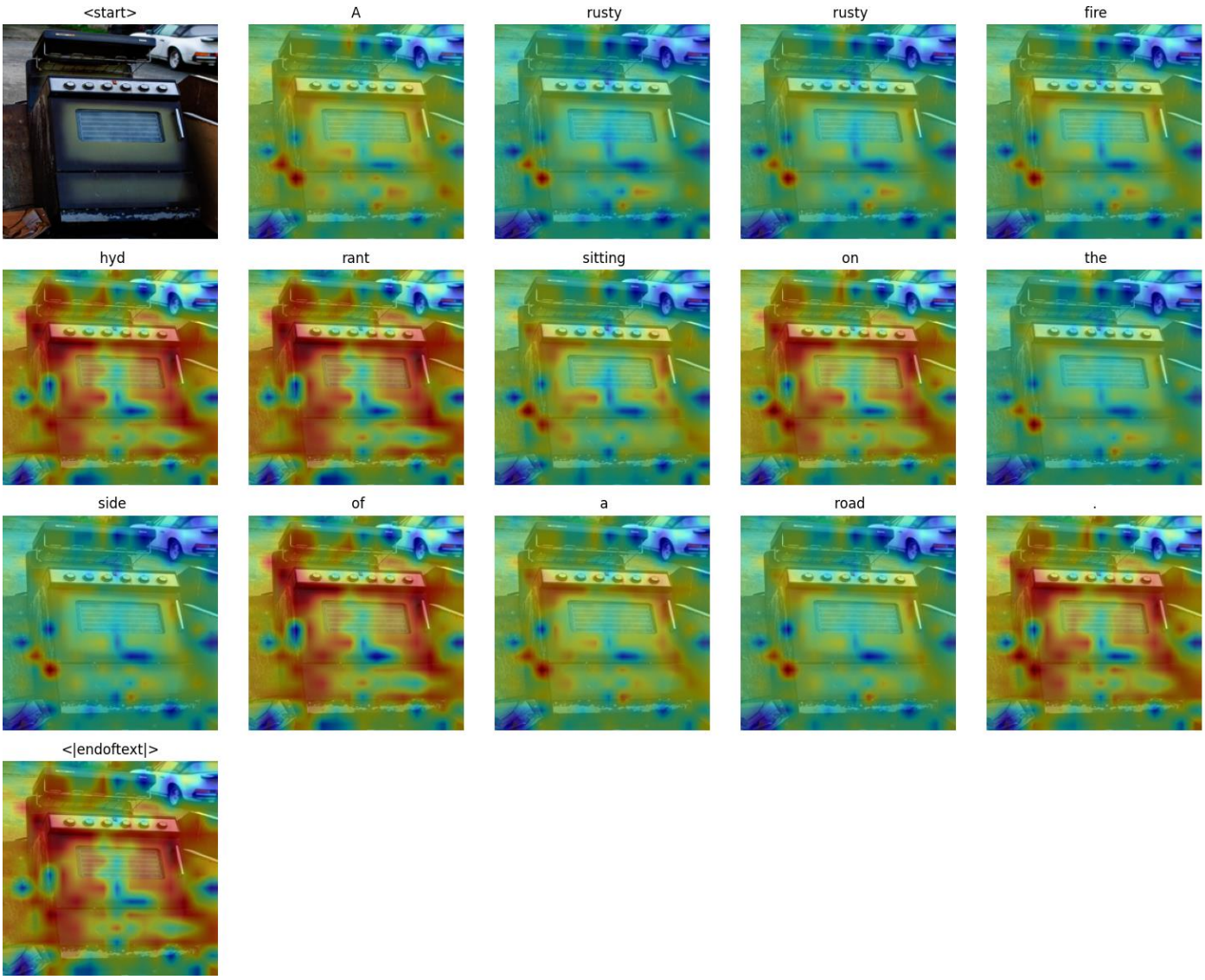


Last1:

Image: 000000001219.jpg

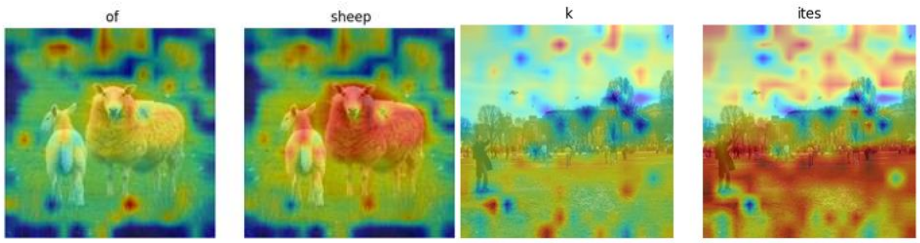
Caption: A rusty rusty fire hyd rant sitting on the side of a road .

CLIPScore: 0.3952

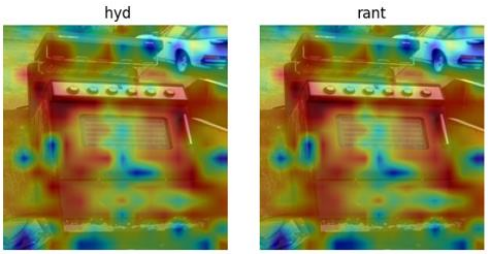


3. Analyze the predicted captions and the attention maps for each word according to the previous question. Is the caption reasonable? Does the attended region reflect the corresponding word in the caption?

大致上還合理，大部分 attention 都有 attend 到正確的地方, ex:



但也有 attend 到正確地方，但分類錯誤的情況，例如在 last-1 image-caption pair 的確有 attend 到相對應的物體，但分類錯誤，不是 hydrant 而應該是 oven 之類的 word。



Reference:

1. Chatgpt

https://chatgpt.com/

2. Hugging Face llava-hf/llava-1.5-7b-hf

https://huggingface.co/llava-hf/llava-1.5-7b-hf

3. How to generate text: using different decoding methods for language generation with Transformers

https://huggingface.co/blog/how-to-generate

4. Visual Instruction Tuning

arXiv:2304.08485

5. LoRA: Low-Rank Adaptation of Large Language Models

https://pypi.org/project/loralib/

6. Hugging Face timm/vit_large_patch14_clip_224.laion2b_ft_in12k_in1k

https://huggingface.co/timm/vit_large_patch14_clip_224.laion2b_ft_in12k_in1k