Supplementary Material for Teaching Structured Vision & Language Concepts to Vision & Language Models

Sivan Doveh[1,2], Assaf Arbelle[1], Sivan Harary[1], Eli Schwartz[1,3], Roei Herzig[1,3], Raja Giryes[3], Rogerio Feris[4], Rameswar Panda[4], Shimon Ullman*[2], Leonid Karlinsky*[4]

[1]IBM Research, [2]Weizmann Institute of Science, [3]Tel-Aviv University, [4]MIT-IBM Watson AI Lab

1. Rule-Based Negative Generation

In this section we describe in detail the process of the Rule-Based negative generation discussed in Section 3.1.1 of the main paper. The Rule-Based method is a simple yet effective method for generating negative examples of a specific SVLC type. We do this by first collecting a list of all words related to the desired SVLC type, this can be done by a simple internet search. For every sentence in our dataset we can compare the words in the sentence with the list. If the sentence contains a word from the list we randomly replace it with a different word from the same list.

Algorithm 1 A pseudo-code for generating Rule-Based negative examples

Let $\mathcal{L}$ be a list of words
Let $\mathcal{T}$ a dataset of sentences
for all $t \in \mathcal{T}$ **do**
    Let W be all words in t
    for all $w \in t$ **do**
        if $w \in \mathcal{L}$ **then**
            Sample w' from $\mathcal{L}$
            Replace w with w' in t
            break # we replace only a single word
        end if
    end for
end for

For example, when creating the rule-based negatives for **colors** we collected the list: teal, brown, green, black, silver, white, yellow, purple, gray, blue, orange, red, blond, concrete, cream, beige, tan, pink, maroon, olive, violet, charcoal, bronze, gold, navy, coral, burgundy, mauve, peach, rust, cyan, clay, ruby, and amber. Then applying the RB-negatives algorithm on a given sentence "A **blue** car on the road" could randomly change the color **blue** to **beige** to get: "A **beige** car on the road". The pseudo-code for RB-negatives is described in Algorithm 1. Similarly, we have RB-negatives generation set-up for several SVLCs, namely: colors, materials, states, sizes, and actions.

2. LLM-Based Negative generation

Algorithm 2 The pseudo-code for generating LLM-based negative examples

Let $\mathcal{T}$ be a dataset of sentences
for all $t \in \mathcal{T}$ **do**
    Parse t using spacy
    Randomly choose a part of the sentence (POSTAG)
    Randomly choose word $w \in t$ that has the chosen POSTAG
    Replace w with <MASK> token in t
    $W' \leftarrow$ unmask using DistilRoBERTa
    Remove w from W'
    Randomly select w' from W'
    Replace w with w' in t
end for

As opposed to the RB negative generation (SupSec 1), the LLM-based negative generation does not require a human definition of the set of valid negatives. This method, introduced in Section 3.1.2 of the main paper, can be broken into three steps using two components of language modeling. First, we extract the linguistic parts of the sentence such as nouns, adjectives, verbs, etc. For this step, we used the spacy [2] "en-core-web-sm" python package. Then we randomly select which part to change and randomly choose a word of that category. We replace the selected word using Distil-Roberta [1, 5, 10] mask-filling capabilities. We replace the selected word with the masking token <MASK> and input the new sentence to the Distil-Roberta model which outputs several candidates for valid words. We select one word from the list, filtering out the original word. For example, the sentence: "Two kids playing in the park" was parsed and the verb "playing" was randomly selected.

*Equal contribution

We mask out this word to get: "Two kids <MASK> in the park". The model then predicts several valid words: sitting, playing, eating, drawing, running. We randomly select one of these possibilities while excluding the original word "playing" to get the negative caption: "Two kids eating in the park". The pseudo-code for LLM-based negatives generation is described in Algorithm 2.

3. Text Analogies via LLM Prompting

This method, as discussed in Section 3.1.3 of the main paper, aims to generate a semantically similar sentence with different wording than the original sentence. When generating negative examples (Sections 1,2) the goal was to make **minimal** changes in the original sentence while significantly changing the meaning. However, in this case we require the exact opposite, i.e. major changes in the sentence while still keeping the same semantic meaning. We generate these sentences using BLOOM [6]. We prompt the model with the following caption: *"a woman standing left to a sitting cat is semantic similar to a cat standing right to a woman. a baby crying to the right of a box is semantic similar to a box placed to the left of a crying baby. a man sitting to the right of a dog is semantic similar to a dog sitting to the left of a man. a blue boat is semantic similar to a boat that is blue."* Followed by: ***CAPTION** is semantic similar to"*, where ***CAPTION*** is the sentence we want to find analogy for. The output of the BLOOM model is a continuation of the sentence in the spirit of the initial prompt.

4. Analysis and Ablation Study

4.1. Individual Dataset Analysis

As discussed in the Dataset section (Section 4.1) of the main paper, the VL-Checklist [11] evaluation dataset is comprised of four different sets, Visual Genome (VG) [3], VAW [7], SWIG [8], and HAKE [4]. In the main paper, we presented the results over a combination of all the datasets evaluated jointly. As promised in the main paper, a more detailed analysis partitioned according to the individual datasets comprising VL-Checklist is available in tables 1-10. Tables 1-5 show the results of the CC3M finetuning experiments, corresponding to Table 1a in the main paper. Tables 6-10 show the results of the models trained from scratch on CC3M, corresponding to Table 1c in the main paper. Note that not all aspects of the tests are available for all datasets, for example, VAW evaluation only includes the "Attribute" aspects, i.e. color, size, material, state, and action. For each dataset we include evaluations on all of its available aspects.

4.2. Error Analysis

In this section we qualitatively evaluate the success and errors of our method and that of the CLIP [9] baseline. Figures 1-2 show examples where our model succeeds while CLIP model fails.

Figure 3 show examples of our method failures. It is evident that most of these failure cases are ambiguous even for a human observer. For example, in the second row of figure 3 there is an image with a lot of suitcases, some are open and some are closed. Therefore, both the positive caption "closed luggage" and the negative caption "open luggage" are valid captions in this case.

5. Code

Our code and pretrained models are available at: https://github.com/SivanDoveh/TSVLC

	O-Large	O-Medium	O-Small	O-Center	O-Mid	O-Margin	Avg O
CLIP [9]	86.95	77.75	72.75	85.5	80.5	70.6	79.00
CLIP +LoRA	86.5	75.9	71.65	85.25	78.25	67.25	77.46
Ours RB+LLM Negs	91.7	**83.2**	**78.9**	**90.3**	**84.55**	**77.4**	**84.34**
Ours Combined	**90.5**	81.95	77.6	89.75	83.8	73.35	82.82

Table 1. Results of **fine-tuning** on CC3M evaluated on the **VG** dataset - **Objects**

	A-Color	A-Material	A-Size	A-State	A-Action	R-action	R-spatial	Avg A+R
CLIP [9]	68.9	65.4	72.1	69.3	72.37	62.4	54	66.35
CLIP +LoRA	72.3	64.8	69.4	63.6	69.71	55.7	41	62.36
Ours RB+LLM Negs	**82.7**	**84.9**	**78.1**	**71.6**	**75.13**	**70.00**	**78.4**	**77.26**
Ours Combined	79.9	78	76.8	68.7	74.18	61.9	63.2	71.81

Table 2. Results of **fine-tuning** on CC3M evaluated on the **VG** dataset - **Attributes, Relations**

	O-Large	O-Medium	O-Small	O-Center	O-Mid	O-Margin	R-action	Avg All
CLIP [9]	76.975	73.28	59.41	78.075	74.63	64.49	77.2	72.00
CLIP +LoRA	80.82	75.02	60.81	81.6	76.94	68.37	81.8	75.05
Ours RB+LLM Negs	82.77	77.97	67.34	83.05	79.92	**75.36**	**87.6**	79.14
Ours Combined	**83.5**	**80.05**	**71.70**	**84.02**	**81.17**	75.01	84.2	**79.95**

Table 3. Results of **fine-tuning** on CC3M evaluated on the **SWIG** dataset

	O-Large	O-Medium	O-Small	O-Center	O-Mid	O-Margin	R-action	Avg All
CLIP [9]	**97.9**	**93.3**	**90.00**	98.6	98.1	**89.7**	78.2	**92.25**
CLIP +LoRA	96.2	89.2	85.1	97.7	96.4	83.7	72.6	88.7
Ours RB+LLM Negs	**97.9**	89.8	88.4	**99.2**	97.7	86.1	**79.4**	91.21
Ours Combined	97.6	89.8	86.5	98.6	**98.5**	86.6	78	90.8

Table 4. Results of **fine-tuning** on CC3M evaluated on the **HAKE** dataset

	A-Color	A-Material	A-Size	A-State	A-Action	Avg All
CLIP [9]	71	73.3	68	53.3	62.7	65.66
CLIP +LoRA	74	71.4	66.9	51.6	59.1	64.6
Ours RB+LLM Negs	**75.7**	**83.5**	66.3	**56.6**	**64.6**	**69.34**
Ours Combined	75	76.7	**69.9**	55.9	**64.6**	68.42

Table 5. Results of **fine-tuning** on CC3M evaluated on the **VAW** dataset

	O-Large	O-Medium	O-Small	O-Center	O-Mid	O-Margin	Avg O
CLIP [9]	76.5	66.15	**64.35**	74.85	66.6	62.35	68.46
Ours RB+LLM Negs	**79.4**	**67.8**	62.15	**75.15**	**70.00**	**64.7**	**69.86**
Ours Combined	76.7	67.4	62.15	74.7	67.85	64.15	68.82

Table 6. Results of **training from scratch** on CC3M evaluated on the **VG** dataset - **Objects**

	A-Color	A-Material	A-Size	A-State	A-Action	R-action	R-spatial	Avg A+R
CLIP [9]	62	58.3	**68.4**	46.8	63.87	44.3	32.5	53.73
Ours RB+LLM Negs	72.4	**74.4**	57	**61.2**	**75.35**	**54.7**	**82.3**	**68.19**
Ours Combined	**74**	64.4	65.9	54.6	70.99	51.4	56.2	62.49

Table 7. Results of **training from scratch** on CC3M evaluated on the **VG** dataset - **Attributes, Relations**

	O-Large	O-Medium	O-Small	O-Center	O-Mid	O-Margin	R-action	Avg All
CLIP [9]	**68.15**	62.36	58.60	**68.15**	63.74	60.26	**65.9**	63.88
Ours RB+LLM Negs	66.02	63.60	**61.51**	65.92	64.10	63.94	61.4	63.78
Ours Combined	67.57	**64.94**	60.21	66.55	**65.52**	**67.49**	60.4	**64.67**

Table 8. Results of **training from scratch** on CC3M evaluated on the **SWIG** dataset

	O-Large	O-Medium	O-Small	O-Center	O-Mid	O-Margin	R-action	Avg All
CLIP [9]	87.7	78.8	72	90.4	85	75.1	63.5	78.92
Ours RB+LLM Negs	86.8	**85.8**	**79.6**	**91.9**	86	**79.2**	**74.8**	**83.44**
Ours Combined	**88.1**	76.5	72.7	90.5	**86.1**	73	68.3	79.31

Table 9. Results of **training from scratch** on CC3M evaluated on the **HAKE** dataset

	A-Color	A-Material	A-Size	A-State	A-Action	Avg All
CLIP [9]	55.4	57	61.3	48.3	57	55.8
Ours RB+LLM Negs	**75.1**	**68.6**	56.3	**61.1**	**60.2**	**64.26**
Ours Combined	61.3	63.7	**68.6**	53.8	55.5	60.58

Table 10. Results of **training from scratch** on CC3M evaluated on the **VAW** dataset

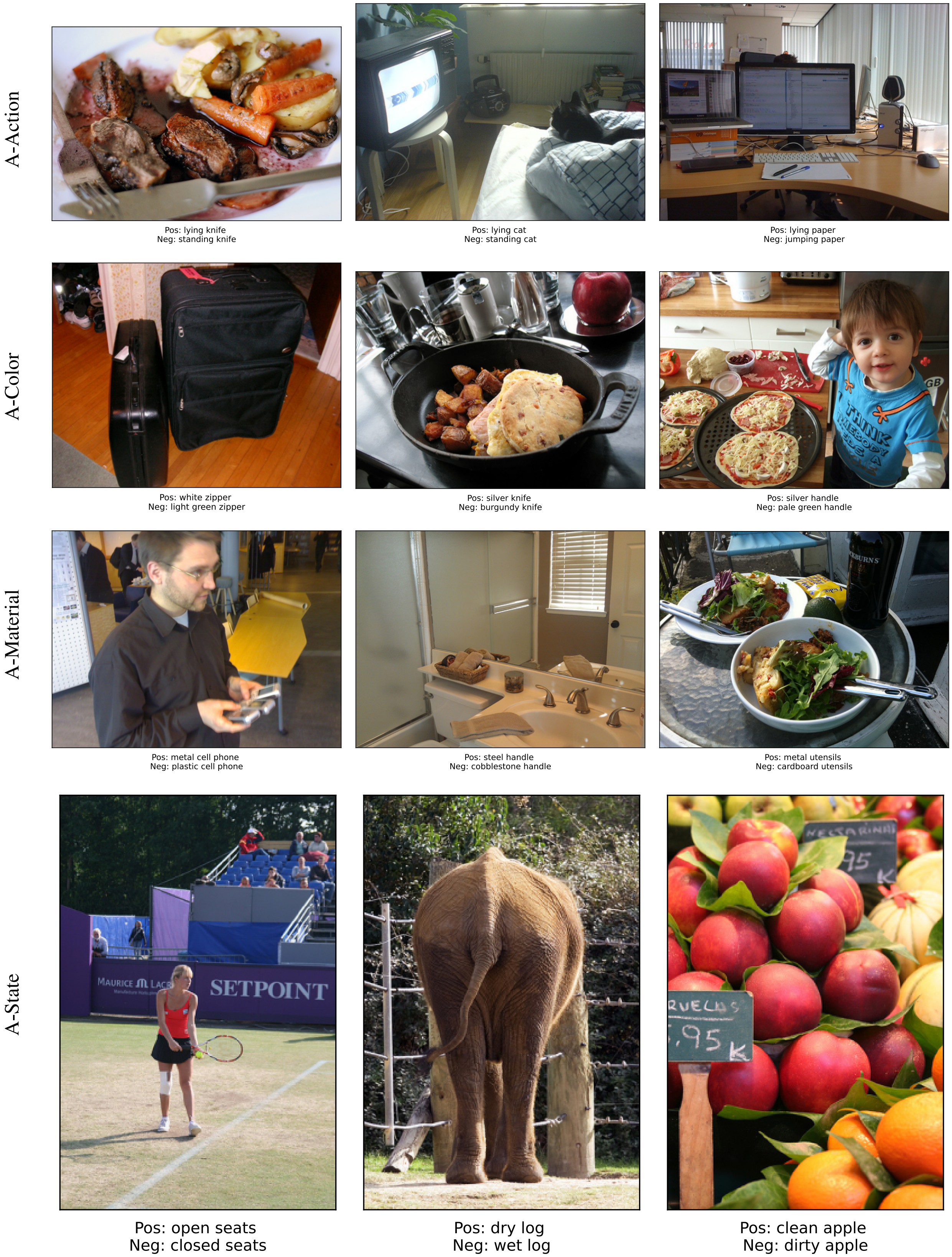

Figure 1. Examples where our model correctly chooses the positive caption, while the CLIP baseline fails and incorrectly chooses the negative caption. We show the respective positive and negative captions underneath each image.

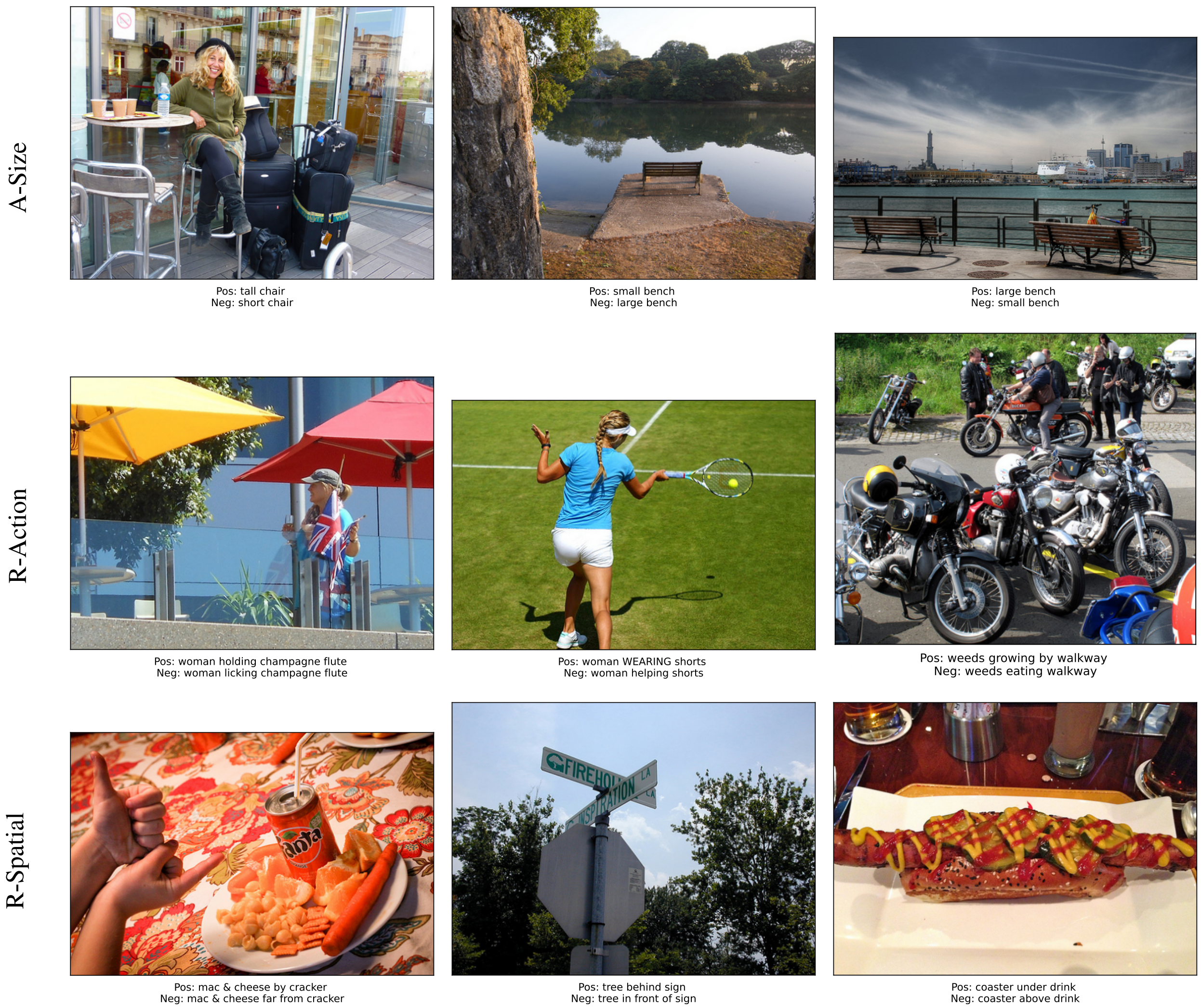

Figure 2. Examples where our model correctly chooses the positive caption, while the CLIP baseline fails and incorrectly chooses the negative caption. We show the respective positive and negative captions underneath each image.

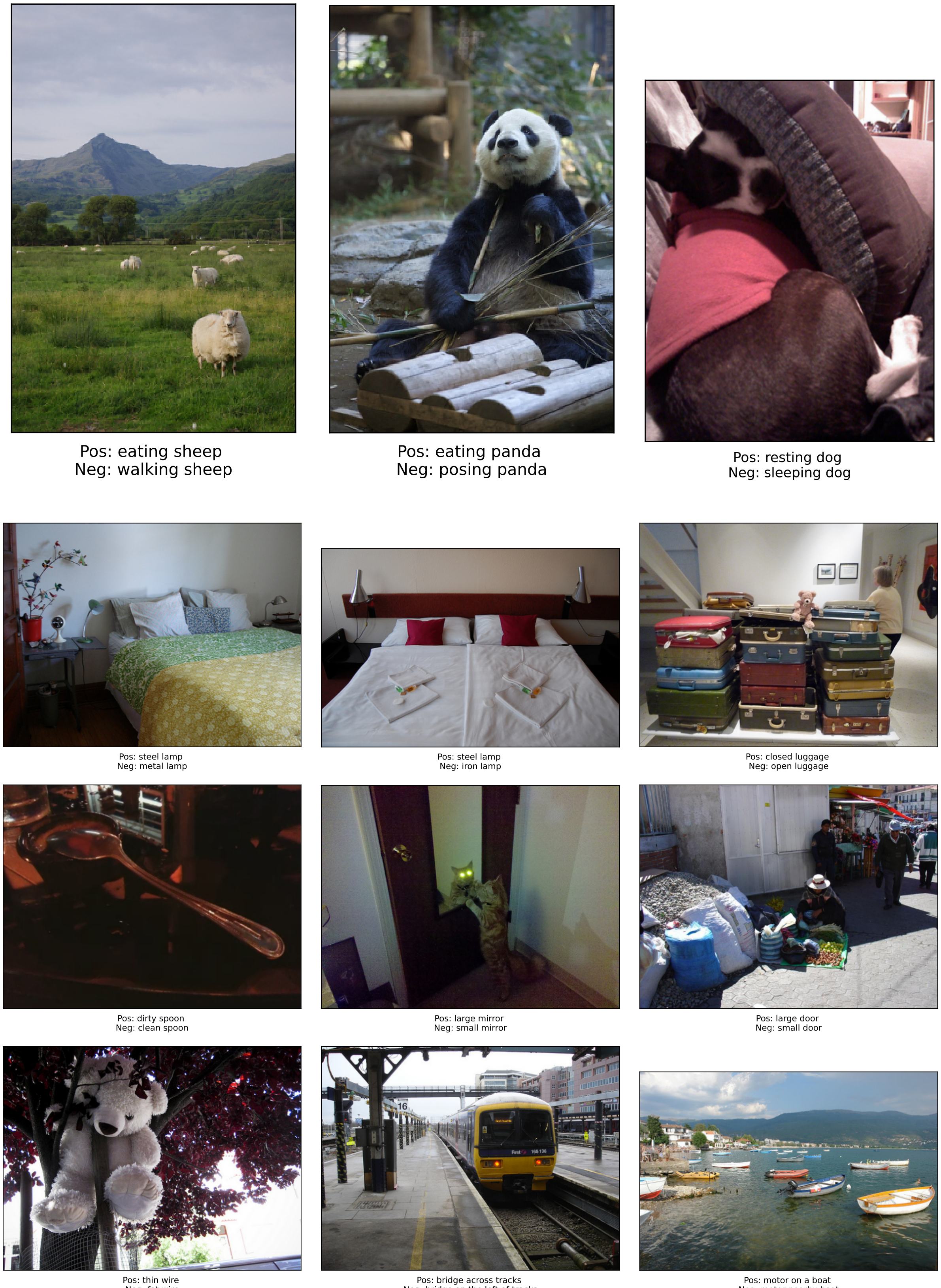

Figure 3. Failure cases of our method. In most cases, they are justifiable as both positive and negative captions match the respective images. Notably, these examples are also failing CLIP.

References

[1] DistillRoBERTa HuggingFace model card. https://huggingface.co/distilroberta-base. Accessed: 2022-11-01. 1

[2] Matthew Honnibal and Ines Montani. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear, 2017. 1

[3] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73, 2017. 2

[4] Yong-Lu Li, Liang Xu, Xinpeng Liu, Xijie Huang, Yue Xu, Mingyang Chen, Ze Ma, Shiyi Wang, Hao-Shu Fang, and Cewu Lu. Hake: Human activity knowledge engine. *arXiv preprint arXiv:1904.06539*, 2019. 2

[5] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019. 1

[6] Margaret Mitchell, Giada Pistilli, Yacine Jernite, Ezinwanne Ozoani, Marissa Gerchick, Nazneen Rajani, Sasha Luccioni, Irene Solaiman, Maraim Masoud, Somaieh Nikpoor, Muñoz Ferrandis Carlos, Stas Bekman, Christopher Akiki, Danish Contractor, David Lansky, Angelina McMillan-Major, Tristan Thrush, Suzana Ilić, Gérard Dupont, Shayne Longpre, Manan Dey, Stella Biderman, Douwe Kiela, Emi Baylor, Teven Le Scao, Aaron Gokaslan, Julien Launay, and Niklas Muennighoff. Bigscience, bigscience language open-science open-access multilingual (bloom) language model. *International*, May 2021-May 2022. 2

[7] Khoi Pham, Kushal Kafle, Zhe Lin, Zhihong Ding, Scott Cohen, Quan Tran, and Abhinav Shrivastava. Learning to predict visual attributes in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13018–13028, 2021. 2

[8] Sarah Pratt, Mark Yatskar, Luca Weihs, Ali Farhadi, and Aniruddha Kembhavi. Grounded situation recognition. In *European Conference on Computer Vision*, pages 314–332. Springer, 2020. 2

[9] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 2, 3, 4

[10] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108, 2019. 1

[11] Tiancheng Zhao, Tianqi Zhang, Mingwei Zhu, Haozhan Shen, Kyusong Lee, Xiaopeng Lu, and Jianwei Yin. Vl-checklist: Evaluating pre-trained vision-language models with objects, attributes and relations. *arXiv preprint arXiv:2207.00221*, 2022. 2