A Limitation, future work, and societal impact

A.1 Limitation and future work

There are several limitations to this work that future research can further explore. First, we focus our scope on compositionality benchmarks formulated as image-to-text retrieval task. While this is currently the most prevailing evaluation framework, future research can characterize compositionality evaluation as text-to-image retrieval problem, as in the initial efforts considered by [32, 39]. More importantly, we hope our work can guide future efforts in creating and ensuring faithful compositionality benchmarks in text-to-image form. Second, in this work, we identify *two* human interpretable dataset biases, the nonsensical and non-fluent biases, which may not cover all dataset artifacts that could possibly be exploited by a model. Future work may utilize more sophisticated techniques to remove spurious dataset artifacts beyond human comprehension [20]. Finally, we focus our evaluations on contrastively learned vision-language models [30]. Future work should include and characterize the compositionality of modern generative vision-language models [1, 5, 21].

A.2 Societal impact

As vision-language models such as CLIP [30] are becoming the foundation models for many downstream applications [34, 31], it is imperative to understand the limitations of these models to avoid misuses and undesirable outcomes [6, 2]. Compositionality benchmarks probe a model's understanding of finer-grained concepts, and hence allow us to identify blind spots [42, 45, 26] of seemingly powerful models deemed by standard classification and retrieval benchmarks [9, 23]. Our work further alleviates common artifacts in existing compositionality benchmarks that result in overestimation of a model's capability. We hope our proposed benchmark SUGARCREPE leads to more faithful assessment of a vision-language model's compositionality, and can hence guide more accurate usages of the models. Nevertheless, we note that strong performances on SUGARCREPE do not imply perfect models. We envision SUGARCREPE being one of the many benchmarks used to comprehensively understand the abilities of vision-language models from various aspects.

B Implementation details

B.1 Hardware information

All experiments are run on a machine with an Intel(R) Xeon(R) CPU E5-2678 v3 with a 512G memory and two 48G NVIDIA RTX A6000 GPUs.

B.2 Dataset sources

We obtain all existing datasets from their original sources released by the authors. We refer readers to these sources for the dataset licenses. To the best of our knowledge, the data we use does not contain personally identifiable information or offensive content.

- CREPE [26]: We obtain CREPE dataset from its official repository [4].
- ARO [42]: We obtain ARO dataset from its official repository [5].
- VL-CheckList [45]: We obtain VL-CheckList dataset from its official repository [6].
- COCO [23]: We obtain COCO from its official project website [7].

[4] https://github.com/RAIVNLab/CREPE
[5] https://github.com/mertyg/vision-language-models-are-bows
[6] https://github.com/om-ai-lab/VL-CheckList
[7] https://cocodataset.org/

B.3 Software configuration

Models. We detail the sources of the pretrained models we use in the paper, and the hyper-parameters used in training our own models.

- Vera model [24]: We obtain pretrained Vera model released by its author [8].
- Grammar model [27]: We obtain the Grammar model released by the authors [9].
- All pretrained CLIP models: We obtain all pretrained CLIP models' weights from Open-CLIP [10].
- NEGCLIP [42]: We obtain weights for pretrained NEGCLIP released by the authors [11].
- Models trained from scratch: We train RN50 based on OpenCLIP codebase and set hype-rparameters as following: number of warmup steps is 1000, batch size is 256, learning rate is 1e-4, weight decay is 0.1, number of epochs is 30. We augment the original CLIP loss with hard negative captions following NEGCLIP [42].

Evaluations. We base our evaluation framework on OpenCLIP [16]. We follow all default hyper-parameters used for evaluating models.

C Vision-language compositionality benchmarks

We provide an overview of existing vision-language compositionality benchmarks below, with Table 7 summarizing the dataset comparisons.

C.1 Image-to-text formulation

A majority of current benchmarks formulate the evaluation task as image-to-text retrieval problem. These benchmarks generate hard negative texts procedurally through rule-based templates, where each benchmark considers different types of hard negatives.

VL-Checklist [45]. VL-CheckList aims at evaluating vision-language models' understanding of different objects, attributes, and relationships. It contains REPLACE hard negatives generated by replacing atomic parts of the positive texts with other foils. VL-CheckList further breaks the hard negatives down into more granular categories based on the type of the replaced atomic part, *i.e.*, object, attribute, or relationship.

ARO [42]. ARO focuses on models' understanding of different relationships, attributes, and order information. It considers SWAP and SHUFFLE hard negatives. SWAP hard negatives are generated by swapping two words in the positive texts; on the other hand, SHUFFLE hard negatives are generated by shuffling words in the positive texts. ARO further divides SWAP hard negatives into attribute or relationship type.

CREPE [26]. CREPE is a large-scale evaluation benchmark that includes three types of hard negatives: REPLACE, SWAP and NEGATE. REPLACE and SWAP hard negatives are generated as in VL-CheckList and ARO. In addition, NEGATE hard negatives are generated by adding negation keywords (*i.e.*, *not* or *no*) to the original positive texts. The hard negatives are not further divided into fine-grained types (object, attribute, or relations).

C.2 Text-to-image formulation

Complementary to image-to-text formulation, compositionality can as well be evaluated by probing a model to select an image that best matches a given text description, against other hard negative

[8] https://huggingface.co/liujch1998/vera
[9] https://huggingface.co/textattack/distilbert-base-uncased-CoLA
[10] https://github.com/mlfoundations/open_clip
[11] https://github.com/mertyg/vision-language-models-are-bows

Table 7: Summary on vision-language compositionality benchmarks. SUGARCREPE considers image-to-text formulation to enable larger scale evaluation set. In addition, SUGARCREPE considers a wide range of hard negative types. SHUFFLE and NEGATE are omitted as they introduce inevitable biases discussed in Sec. 4.2.

			Hard Negative Text Type				
Benchmark	**Task Formulation**	**Scale**	**SHUFFLE**	**REPLACE**	**SWAP**	**NEGATE**	**ADD**
VL-CheckList [45]	Image-to-Text	> 1000		✓			
ARO [42]	Image-to-Text	> 1000	✓		✓		
CREPE [26]	Image-to-Text	> 1000		✓	✓	✓	
Winoground [39]	Image-to-Text / Text-to-Image	400			✓		
Cola [32]	Text-to-Image	210			N/A		
SUGARCREPE	Image-to-Text	> 1000		✓	✓		✓

images as distractors. Unlike hard negative texts, hard negative images are more difficult to obtain and thus current text-to-image compositionality benchmarks are smaller at scale.

Winoground [39]. Winoground is a small dataset manually curated by human annotators. Each example in the dataset contains two images and two matching captions, where both captions contain identical words that appear in different orders. Note that Winoground can be used for either image-to-text or text-to-image retrieval. While the original intention for Winoground is to evaluate vision-language compositionality, recent work [10] has pointed out that solving the tasks in Winoground requires not just compositional vision-language understanding, but additionally a suite of other abilities such as commonsense reasoning, or distinguishing visually difficult images.

Cola [32]. Cola tests a vision-language model's ability to select an image that correctly matches a given caption, against another distractor image with the same objects and attributes but in the wrong composition. The image pairs are mined from existing datasets. As a result, the final evaluation set is relatively small in size (210 examples in total).

We deem text-to-image evaluation as important as image-to-text evaluation. Future work can explore approaches to generate or mine compositional hard negative images at scale, as preliminarily explored in [32, 42].

D SUGARCREPE

D.1 Taxonomy

Figure 6 shows the taxonomy of SUGARCREPE. We first categorize the hard negatives based on their forms: REPLACE, SWAP, and ADD. We then further divide each type of hard negatives into finer-grained sub-categories based on the type (object, attribute, or relation) of the atomic concept altered. SUGARCREPE covers a total of 7 fine-graind hard negative types.

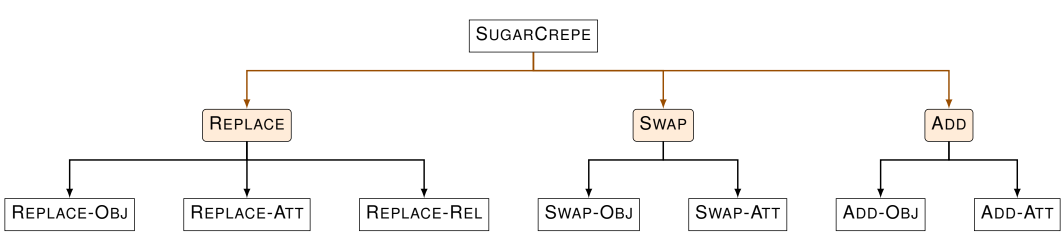

Figure 6: Taxonomy of hard negatives considered in SUGARCREPE.

```
Given an input sentence describing a scene, your task
is to:
1. Locate the noun words in the sentence.
2. Randomly pick one noun word.
3. Replace the selected noun word with a new noun word
to make a new sentence.

The new sentence must meet the following three
requirements:
1. The new sentence must be describing a scene that is
as different as possible from the original scene.
2. The new sentence must be fluent and grammatically
correct.
3. The new sentence must make logical sense.

Here are some examples:

Original sentence: A man is in a kitchen making pizzas.
Nouns: ["man", "kitchen", "pizzas"]
Selected noun: man
New noun: woman
New sentence: A woman is in a kitchen making pizzas.

Original sentence: a woman seated on wall and birds
besides her
Nouns: ['woman', 'wall', 'birds']
Selected noun: wall
New noun: bench
New sentence: A woman seated on a bench and birds
besides her.
```

(a) REPLACE-OBJ.

```
Given an input sentence describing a scene, your task
is to:
1. Locate the adjective words describing objects in the
sentence. If there are no adjective words, return an
empty list.
2. Randomly pick one adjective word.
3. Replace the selected adjective word with a new
adjective word to make a new sentence.

The new sentence must meet the following three
requirements:
1. The new sentence must be describing a scene that is
as different as possible from the original scene.
2. The new sentence must be fluent and grammatically
correct.
3. The new sentence must make logical sense.

Here are some examples:

Original sentence: a blue bike parked on a side walk.
Adjectives: ["blue"]
Selected adjective: blue
New adjective: red
New sentence: a red bike parked on a side walk.

Original sentence: The kitchen is clean and ready for
us to see.
Adjectives: ["clean", "ready"]
Selected adjective: clean
New adjective: dirty
New sentence: The kitchen is dirty and ready for us to
see.
```

(b) REPLACE-ATT.

```
Given an input sentence describing a scene, your task
is to:
1. Find any action or spatial relationships between two
objects in the sentence. If there are no such
relationships, return an empty list.
2. Randomly pick one relationship.
3. Replace the selected relationship with a new
relationship to make a new sentence.

The new sentence must meet the following three
requirements:
1. The new sentence must be describing a scene that is
as different as possible from the original scene.
2. The new sentence must be fluent and grammatically
correct.
3. The new sentence must make logical sense.

Here are some examples:

Original sentence: The dining table near the kitchen
has a bowl of fruit on it.
Relationships: ["near", "on"]
Selected relationship: near
New relationship: far from
New sentence: The dining table far from the kitchen has
a bowl of fruit on it.

Original sentence: A couple of buckets in a white room.
Relationships: ['in']
Selected relationship: in
New relationship: outside
New sentence: A couple of buckets outside a white room.
```

(c) REPLACE-REL.

Figure 7: Example prompt templates (black) and outputs (green) from ChatGPT for REPLACE hard negatives.

D.2 Hard negative generation procedure and templates

To generate hard negatives in SUGARCREPE, we come up with three different prompt templates for the three hard negative types considered: REPLACE, SWAP, and ADD. Each template consists of task instruction for generating the corresponding type of hard negatives and several (7 or more) few-shot demonstrations. We describe the general generation procedure and example prompt templates below and refer readers to our dataset repository for the full prompts used [12].

Generating REPLACE hard negatives. To best leverage ChatGPT's capabilities, we devise a three-step workflow to generate REPLACE hard negatives: (1) We prompt ChatGPT in locating the desired atomic concepts (*e.g.*, objects) in the sentence; (2) We prompt ChatGPT to generate a new concept to replace a randomly selected old concept; (3) We let ChatGPT compose a new sentence by replacing the old concept with the new one. For steps (1) and (3), we prompt ChatGPT with a temperature of 0.0 to get stable outputs. For step (2), however, we diversify the outputs by prompting ChatGPT with a higher temperature of 1.5. With this design, we are able to generate diverse REPLACE hard negatives. Figure 7 shows the example templates and outputs for REPLACE hard negatives.

[12] https://github.com/RAIVNLab/sugar-crepe

```
Given an input sentence describing a scene, your task
is to first locate two swappable noun phrases in the
sentence, and then swap them to make a new sentence.
The new sentence must meet the following three
requirements:
1. The new sentence must be describing a different
scene from the input sentence.
2. The new sentence must be fluent and grammatically
correct.
3. The new sentence must make logical sense.

To complete the task, you should:
1. Answer the question of whether generating such a new
sentence is possible using Yes or No.
2. Output the swappable noun phrases.
3. Swap them to make a new sentence.

Here are some examples:

Input: A cat resting on a laptop next to a person.
Is it possible to swap noun phrases in the input
sentence to generate a new sentence that is different
from the input sentence and makes logical sense? Yes
Swappable noun phrases: laptop, person
Output: A cat resting on a person next to a laptop.

Input: A plate of donuts with a person in the
background.
Is it possible to swap noun phrases in the input
sentence to generate a new sentence that is different
from the input sentence and makes logical sense? Yes
Swappable noun phrases: a plate of donuts, a person
Output: A person with a plate of donuts in the
background.
```

(a) SWAP-OBJ.

```
Given an input sentence describing a scene, your task
is to first locate two swappable adjectives in the
sentence describing different objects, and then swap
them to make a new sentence.
The new sentence must meet the following three
requirements:
1. The new sentence must be describing a different
scene from the input sentence.
2. The new sentence must be fluent and grammatically
correct.
3. The new sentence must make logical sense.

To complete the task, you should:
1. Answer the question of whether generating such a new
sentence is possible using Yes or No.
2. Output the swappable adjectives.
3. Swap them to make a new sentence.

Here are some examples:

Input: A girl in a pink shirt holding a blue umbrella.
Is it possible to swap attributes in the input sentence
to generate a new sentence that is different from the
input sentence and makes logical sense? Yes
Swappable attributes: pink, blue
Output: A girl in a blue shirt holding a pink umbrella.

Input: A girl with a green shirt brushing her teeth
with a blue toothbrush.
Is it possible to swap attributes in the input sentence
to generate a new sentence that is different from the
input sentence and makes logical sense? Yes
Swappable attributes: green, blue
Output: A girl with a blue shirt brushing her teeth
with a green toothbrush.
```

(b) SWAP-ATT.

Figure 8: Example prompt templates (black) and outputs (green) from ChatGPT for SWAP hard negatives.

```
Given an input sentence describing a scene, your task
is:
1. Find the objects in the sentence.
2. Randomly pick one object.
3. Generate a new object that's not in the sentence.
4. Add the new object next to the selected object to
make a new sentence.

The new sentence must meet the following three
requirements:
1. The new sentence must describe a clearly new and
different scene.
2. The new sentence must be fluent and grammatically
correct.
3. The new sentence must make logical sense.

Here are some examples:

Original sentence: An elephant standing under the shade
of a tree.
Objects: ["elephant", "shade of a tree"]
Selected object: elephant
New object: squirrel
New sentence: An elephant and a squirrel standing under
the shade of a tree.

Original sentence: A bench at the beach next to the sea
Objects: ['bench', 'beach', 'sea']
Selected object: bench
New object: umbrella
New sentence: An umbrella and a bench at the beach next
to the sea.
```

(a) ADD-OBJ.

```
Given an input sentence describing a scene, your task
is:
1. Find the objects in the sentence.
2. Randomly pick one object.
3. Generate a new plausible but uncommon attribute for
this object that's not in the sentence.
4. Add the new attribute next to the selected object to
make a new sentence.

The new sentence must meet the following three
requirements:
1. The new sentence must describe a clearly new and
different scene.
2. The new sentence must be fluent and grammatically
correct.
3. The new sentence must make logical sense.

Here are some examples:

Original sentence: A large white airplane and a person
on a lot.
Objects: ["airplane", "person"]
Selected object: airplane
New attribute: blue
New sentence: A large white and blue airplane and a
person on a lot.

Original sentence: three people riding horses on a
beach
Objects: ['three people', 'horses', 'beach']
Selected object: three people
New attribute: elderly
New sentence: Three elderly people riding horses on a
beach.
```

(b) ADD-ATT.

Figure 9: Example prompt templates (black) and outputs (green) from ChatGPT for ADD hard negatives.

Generating SWAP hard negatives. To generate SWAP hard negatives, which do not require any new concepts, we simply prompt ChatGPT once with a temperature of 0.0. Unlike REPLACE, SWAP hard negatives are only possible when there are at least two atomic concepts of the same category, *i.e.*, either object or attribute. Thus, our prompt first queries ChatGPT whether it is possible to swap two atomic concepts in the input sentence to generate a new description. Only if the answer is yes, will ChatGPT then proceed to identify two swappable concepts and compose the corresponding new sentence by swapping the two concepts. Figure 8 shows the example templates and outputs for SWAP hard negatives.

Generating ADD hard negatives. Similar to the REPLACE, we also employ a three-step prompting procedure to generate ADD hard negatives. The only difference in the procedure is that we prompt

ChatGPT to add the generated new concept to the original caption, instead of using it to replace an old concept. Figure 9 shows the example templates and outputs for ADD hard negatives.

D.3 Adversarial refinement

We detail the adversarial refinement procedure below. Given a text model M, we denote its output score for the positive and negative caption of i-th image as $M(p_i)$ and $M(n_i)$. If $M(p_i) > M(n_i)$, then the model could identify the correct caption for the i-th image without referring to it. For a test set to be unattackable given the text model M, the expectation of M's identifying the correct caption should be as close to random guess as possible; in particular, we hope that $E_i[M(p_i) > M(n_i)] = 0.5$. To achieve this for both the grammar model M_1 and plausibility model M_2, we first calculate the score difference $g_i^{(1)} = M_1(p_i) - M_1(n_i)$ and $g_i^{(2)} = M_2(p_i) - M_2(n_i)$, where the range of both $g^{(1)}$ and $g^{(2)}$ is $[-1, 1]$. Then we split the 2D space of the joint range of $g^{(1)}$ and $g^{(2)}$ into 100×100 equal grids, and for each pair of symmetric grids, *e.g.*, $\{(g^{(1)}, g^{(2)}) | g^{(1)} \in (0.02, 0.04], g^{(2)} \in (-0.04, 0.06]\}$ and $\{(g^{(1)}, g^{(2)}) | g^{(1)} \in (-0.02, -0.04], g^{(2)} \in (0.04, -0.06]\}$, we preserve the same number of data for both grids, therefore we ensure that for the resultant set, $E_i[M_1(p_i) > M_1(n_i)] = 0.5$ and $E_i[M_2(p_i) > M_2(n_i)] = 0.5$.

D.4 Dataset information

We host SUGARCREPE on Github [13]. The data card [29] for SUGARCREPE, containing detailed dataset documentation, is available at the dataset repository [14]. We provide a summary below.

Dataset documentation. SUGARCREPE is a benchmark for faithful vision-language compositionality evaluation. Given an image, a model is required to select the positive text that correctly describes the image, against another hard negative text distractor that differs from the positive text only by small compositional changes. Each example consists of three fields:

- `filename`: The id to an image
- `caption`: Positive text correctly describing the image
- `negative_caption`: Hard negative text incorrectly describing the image

Maintenance plan. We are committed to maintain the dataset to address any technical issues. We actively monitor issues in the repository.

Licensing. We license our work using MIT License [15]. All the source data we use is publicly released by prior work [23].

Author statement. We the authors will bear all responsibility in case of violation of rights.

E Detailed evaluation results

E.1 Full evaluation results on existing benchmarks

We provide the full evaluation results over 17 pretrained CLIP models as well as 2 text-only models, Vera [24] and the Grammar model [27], on existing compositionality benchmarks in Table 8. We see that the text-only models, arguably without any vision-language compositionality, outperform most of the pretrained CLIP models, achieving state-of-the-art performances on many benchmark tasks. This implies that current benchmarks fail to faithfully reflect a model's vision-language compositionality.

[13] `https://github.com/RAIVNLab/sugar-crepe`
[14] `https://github.com/RAIVNLab/sugar-crepe/blob/main/data_card.pdf`
[15] `https://github.com/RAIVNLab/sugar-crepe/blob/main/LICENSE`

Table 8: Blind models (*i.e.*, Vera and Grammar model) outperform all 17 existing pretrained CLIP models on nearly all existing benchmark tasks. This implies that current benchmarks fail to faithfully measure a model's vision-language compositionality.

		CREPE			ARO				VL-Checklist		
Source	**Model**	**Atomic**	**Swap**	**Negate**	**VG-Relation**	**VG-Attribution**	**COCO-Order**	**Flickr30K-Order**	**Object**	**Attribute**	**Relation**
Text-only model	Vera [24]	43.70	70.80	66.15	61.71	82.59	59.81	63.52	82.48	73.99	85.72
	Grammar [27]	18.15	50.88	9.77	59.55	58.38	74.33	76.26	57.95	52.35	68.50
OpenAI [30]	RN50	26.47	28.32	31.25	53.87	63.37	44.89	52.46	86.85	68.30	75.95
	RN101	27.63	32.74	12.50	52.43	62.93	29.86	39.34	86.44	67.93	71.75
	RN50x4	26.24	28.32	9.51	51.59	62.27	29.39	34.56	87.23	68.74	73.81
	ViT-B-32	22.31	26.55	28.78	51.12	61.33	37.14	47.18	87.00	68.80	77.04
	RN50x16	26.36	29.65	9.38	52.13	62.71	29.95	34.26	86.95	69.34	76.83
	RN50x64	26.82	30.09	23.57	51.00	62.56	40.54	46.74	87.71	68.61	74.97
	ViT-L-14	26.36	25.66	24.74	53.34	61.50	36.11	45.08	87.86	68.27	75.89
LAION [36]	ViT-H-14	23.70	25.22	16.54	50.33	62.93	25.79	30.96	85.39	68.46	71.13
	ViT-g-14	23.70	24.78	20.70	51.60	61.20	25.59	30.10	86.07	69.43	71.03
	ViT-bigG-14	23.58	24.78	17.97	51.61	61.89	25.24	30.22	84.66	67.80	66.48
	roberta-ViT-B-32	22.66	21.24	20.31	47.46	62.00	24.77	30.76	85.71	68.82	65.90
	xlm-roberta-base-ViT-B-32	21.16	20.80	12.76	47.93	59.73	23.85	30.32	86.06	70.41	63.01
	xlm-roberta-large-ViT-H-14	24.16	23.89	20.05	46.14	57.84	26.05	31.00	87.89	70.25	63.89
DataComp [12]	`small`:ViT-B-32	13.64	27.88	14.84	50.83	50.17	13.35	14.02	68.72	58.80	57.00
	`medium`:ViT-B-32	16.42	20.35	11.33	50.45	54.04	16.44	16.26	78.43	63.53	62.94
	`large`:ViT-B-16	18.15	17.26	17.06	48.82	53.21	21.49	26.44	84.73	65.72	64.81
	`x-large`:ViT-L-14	21.62	22.57	16.28	48.54	60.03	23.19	29.52	86.66	67.01	67.93

E.2 SUGARCREPE human evaluation

To compare the quality of the hard negatives generated in SUGARCREPE to those in current benchmarks (*i.e.*, ARO+CREPE), we randomly sample 100 examples for each of the hard negative types: REPLACE, SWAP, and NEGATE / ADD. Each example is organized to consist of (1) the original positive text, (2) its hard negative in ARO+CREPE, and (3) its hard negative in SUGARCREPE. For each example, a human user rates whether the hard negative in ARO+CREPE or that in SUGARCREPE is better (or tie) in terms of commonsense and grammatical correctness, respectively. Note that we compare NEGATE in ARO+CREPE to ADD in SUGARCREPE, as both hard negatives are intended to probe a model's understanding of the *existence or not* of an atomic concept. Table 9 shows that hard negatives in SUGARCREPE are much more sensical and fluent than that in ARO+CREPE across all three different types. For instance, SUGARCREPE has 68% more sensical and 46% more fluent hard negatives than ARO+CREPE on SWAP.

Table 9: Human evaluation results on the comparisons between hard negatives in ARO+CREPE and SUGARCREPE. We report the counts (out of 100 sampled examples) that the human user considers better or tie, w.r.t. both commonsense and grammatical correctness.

		Human counts of better examples		
Hard-negative Type	Evaluation	ARO+CREPE	SUGARCREPE	Tie
REPLACE	Commonsense	11	29	60
	Grammar	4	33	63
SWAP	Commonsense	4	68	28
	Grammar	4	46	50
NEGATE / ADD	Commonsense	1	26	73
	Grammar	1	35	64

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022.

[2] Federico Bianchi, Pratyusha Kalluri, Esin Durmus, Faisal Ladhak, Myra Cheng, Debora Nozza, Tatsunori Hashimoto, Dan Jurafsky, James Zou, and Aylin Caliskan. Easily accessible text-to-image generation amplifies demographic stereotypes at large scale. *arXiv preprint arXiv:2211.03759*, 2022.

[3] Léon Bottou. From machine learning to machine reasoning. *Machine learning*, 94(2):133–149, 2014.

[4] Paola Cascante-Bonilla, Khaled Shehada, James Seale Smith, Sivan Doveh, Donghyun Kim, Rameswar Panda, Gül Varol, Aude Oliva, Vicente Ordonez, Rogerio Feris, and Leonid Karlinsky. Going beyond nouns with vision & language models using synthetic data, 2023.

[5] Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, et al. Pali: A jointly-scaled multilingual language-image model. *arXiv preprint arXiv:2209.06794*, 2022.

[6] Jaemin Cho, Abhay Zala, and Mohit Bansal. Dall-eval: Probing the reasoning skills and social biases of text-to-image generative transformers. *arXiv preprint arXiv:2202.04053*, 2022.

[7] Noam Chomsky and Morris Halle. Some controversial questions in phonological theory. *Journal of linguistics*, 1(2):97–138, 1965.

[8] MJ Cresswell. Logics and languages. 1973.

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, K. Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[10] Anuj Diwan, Layne Berry, Eunsol Choi, David Harwath, and Kyle Mahowald. Why is winoground hard? investigating failures in visuolinguistic compositionality. *arXiv preprint arXiv:2211.00768*, 2022.

[11] Sivan Doveh, Assaf Arbelle, Sivan Harary, Eli Schwartz, Roei Herzig, Raja Giryes, Rogerio Feris, Rameswar Panda, Shimon Ullman, and Leonid Karlinsky. Teaching structured vision & language concepts to vision & language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2657–2668, 2023.

[12] Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, et al. Datacomp: In search of the next generation of multimodal datasets. *arXiv preprint arXiv:2304.14108*, 2023.

[13] Madeleine Grunde-McLaughlin, Ranjay Krishna, and Maneesh Agrawala. Agqa: A benchmark for compositional spatio-temporal reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[14] Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[15] Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. Compositionality decomposed: How do neural networks generalise? *Journal of Artificial Intelligence Research*, 67:757–795, 2020.

[16] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, July 2021.

[17] Theo MV Janssen and Barbara H Partee. Compositionality. In *Handbook of logic and language*, pages 417–473. Elsevier, 1997.

[18] Jingwei Ji, Ranjay Krishna, Li Fei-Fei, and Juan Carlos Niebles. Action genome: Actions as compositions of spatio-temporal scene graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10236–10247, 2020.

[19] 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.

[20] Ronan Le Bras, Swabha Swayamdipta, Chandra Bhagavatula, Rowan Zellers, Matthew Peters, Ashish Sabharwal, and Yejin Choi. Adversarial filters of dataset biases. In *International Conference on Machine Learning*, pages 1078–1088. PMLR, 2020.

[21] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, 2023.

[22] Junnan Li, Dongxu Li, Caiming Xiong, and Steven C. H. Hoi. BLIP: bootstrapping language-image pre-training for unified vision-language understanding and generation. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 12888–12900. PMLR, 2022.

[23] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[24] Jiacheng Liu, Wenya Wang, Dianzhuo Wang, Noah A Smith, Yejin Choi, and Hannaneh Hajishirzi. Vera: A general-purpose plausibility estimation model for commonsense statements. *arXiv preprint arXiv:2305.03695*, 2023.

[25] Cewu Lu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. Visual relationship detection with language priors. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 852–869. Springer, 2016.

[26] Zixian Ma, Jerry Hong, Mustafa Omer Gul, Mona Gandhi, Irena Gao, and Ranjay Krishna. Crepe: Can vision-language foundation models reason compositionally? *arXiv preprint arXiv:2212.07796*, 2022.

[27] John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 119–126, 2020.

[28] OpenAI. Chatgpt. 2022.

[29] Mahima Pushkarna, Andrew Zaldivar, and Oddur Kjartansson. Data cards: Purposeful and transparent dataset documentation for responsible ai. In *2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 1776–1826, 2022.

[30] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 2021.

[31] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

[32] Arijit Ray, Filip Radenovic, Abhimanyu Dubey, Bryan A. Plummer, Ranjay Krishna, and Kate Saenko. Cola: How to adapt vision-language models to compose objects localized with attributes?, 2023.

[33] Yuval Reif and Roy Schwartz. Fighting bias with bias: Promoting model robustness by amplifying dataset biases. *arXiv preprint arXiv:2305.18917*, 2023.

[34] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

[35] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

[36] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade W Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

[37] Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. Flava: A foundational language and vision alignment model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15638–15650, 2022.

[38] Harman Singh, Pengchuan Zhang, Qifan Wang, Mengjiao Wang, Wenhan Xiong, Jingfei Du, and Yu Chen. Coarse-to-fine contrastive learning in image-text-graph space for improved vision-language compositionality. 2023.

[39] Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. Winoground: Probing vision and language models for visio-linguistic compositionality. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5238–5248, 2022.

[40] Junke Wang, Dongdong Chen, Zuxuan Wu, Chong Luo, Luowei Zhou, Yucheng Zhao, Yujia Xie, Ce Liu, Yu-Gang Jiang, and Lu Yuan. Omnivl: One foundation model for image-language and video-language tasks. *arXiv preprint arXiv:2209.07526*, 2022.

[41] Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, et al. Image as a foreign language: Beit pretraining for all vision and vision-language tasks. *arXiv preprint arXiv:2208.10442*, 2022.

[42] Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it? In *International Conference on Learning Representations*, 2023.

[43] Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. Swag: A large-scale adversarial dataset for grounded commonsense inference. *arXiv preprint arXiv:1808.05326*, 2018.

[44] Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. Lit: Zero-shot transfer with locked-image text tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18123–18133, 2022.

[45] 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.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments (e.g. for benchmarks)...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]