

```

<task>
  Rewrite the following image description:
</task>
**Input**:
{caption}
**output**:

```

Figure 6: A prompt for generating synthetic paraphrased captions.

A Experimental Details

A.1 Implementation Details

Batch Sampling for Training. As described in Sec. 4.1, all our experiments are conducted using the COCO [34] dataset. The COCO dataset is an image-caption dataset consisting of images, each associated with a set of captions. During training, we first sample a batch of B images from this dataset. For each $i \in \{1, 2, \dots, B\}$, let $\mathcal{T}_i = \{T_i^{(n)}\}_{n=1}^N$ denote the set of N captions associated with I_i (where I_i denotes the i -th image). From this set, we randomly select one caption $T_i \in \mathcal{T}_i$ to form a positive image-text pair (I_i, T_i) . In this way, we obtain a batch of B image-text pairs, denoted as $\{(I_i, T_i)\}_{i=1}^B$.

Implementation Details for Each Loss Component. Recall that our proposed *REconstruction and Alignment of text Descriptions* (READ) method comprises three components in the final loss (Eq. 8): the standard *contrastive* loss, the token-level *reconstruction* loss, and the sentence-level *alignment* loss. For the *contrastive* loss (Eq. 4), we incorporate $M = 3$ hard negative captions per image, generated via rule-based perturbations as proposed in NegCLIP [64]. For the *reconstruction* loss (Eq. 5), we use a frozen decoder extracted from the pre-trained encoder-decoder language model, T5-Large [45]. To obtain $\{y_i^{(k)}\}_{k=1}^K$, we randomly sample $K = 1$ element from the caption set \mathcal{T}_i associated with each image-text pair (I_i, T_i) . For the *alignment* loss (Eq. 7), we generate a paraphrased caption T'_i for each T_i via augmentation using large language models. Specifically, prior to training, we generate one paraphrased caption for every caption in each image’s original caption set using the gpt-4o-mini-2024-07-18 model [21]. This is done by applying a simple prompt as shown in Fig.6, with a temperature of 1.0 and all other parameters set to their default values [21]. From this augmentation, we obtain a synthetic caption set for each image. Given a batch of sampled image-text pairs $\{(I_i, T_i)\}_{i=1}^B$ during training, we randomly sample one caption from the union of the original and synthetic caption sets associated to I_i , and use it as T'_i . Finally, the weighting factors in Eq. 8 are set to $\alpha = 0.1$ and $\beta = 0.5$.

Training Details. We fine-tune all models using the Huggingface transformers [60] library². The AdamW optimizer is used with a learning rate of 1.0×10^{-5} , cosine annealing schedule, 50 warmup steps, and a weight decay of 0.1. Training is performed with bf16 mixed precision for computational efficiency. All experiments are conducted using a single A100 40GB GPU.

A.2 Evaluation Benchmarks

Benchmarks Details. **WhatsUp** [23] evaluates spatial reasoning by testing whether models can interpret relative object positions. **CREPE** [36] measures compositional reasoning at varying complexity levels using logical operations such as conjunction, negation, and attribute swapping. **VALSE** [39] assesses fine-grained linguistic understanding, including object existence, quantity, action semantics, and coreference resolution. **SugarCrepe** [18] focuses on relational reasoning through hard negative captions crafted with natural linguistic variation. **SugarCrepe++** [10] extends SugarCrepe by adding a paraphrased positive caption and introduces two tasks: (1) image-to-text (ITT), which tests whether all paraphrased positives for a given image are ranked above all negatives, and (2) text-to-text (TOT), which evaluates semantic consistency by checking whether each positive paraphrase pair is ranked above all negative pairs in the absence of visual context. Since our study

²<https://github.com/huggingface/transformers>

Benchmark	License	Image Source
CREPE-Productivity [36]	Unspecified	Visual Genome [26]
SugarCrepe [18]	MIT	COCO [34]
SugarCrepe++ [10]	MIT	COCO [34]
WhatsUp [23]	MIT	Custom-collected, COCO [34], GQA [20]
VALSE [39]	MIT	Visual7W [73], COCO [34], SWiG [42], FOIL-it [51]

Table 5: Benchmarks used in the evaluation, along with license and image source.

947 aims to improve the compositional reasoning capability of VLMs such as CLIP, we primarily adopt
948 the ITT metric as a major focus for evaluation, while including TOT as a supplementary measure.

949 **Benchmarks and Licensing.** We conduct our evaluation on five publicly available compositional
950 reasoning benchmarks. Table 5 summarizes their license information and image sources. All datasets
951 used for training and evaluation are either MIT-licensed or publicly released for research use.

	<div><div>✓ Positive</div><div>✗ Negative</div></div>		<div>w/o Reconst. Loss</div>	<div>w/ Reconst. Loss</div>		<div>w/o Reconst. Loss</div>	<div>w/ Reconst. Loss</div>	
WhatsUp		<div>✓ A laptop to the left of a armchair</div> <div>✗ A laptop <u>on</u> a armchair</div> <div>✗ A laptop <u>under</u> a armchair</div> <div>✗ A laptop to the <u>right</u> of a armchair</div>	#2	#1		<div>✓ A can in front of a headphones</div> <div>✗ A can <u>behind</u> a headphones</div> <div>✗ A can <u>to the left</u> of a headphones</div> <div>✗ A can <u>to the right</u> of a headphones</div>	#3 #2 #1 #4	#1 #2 #3 #4
		<div>✓ trash can on ground. there is a liner.</div> <div>✗ trash can <u>inside</u> ground. there is a liner.</div> <div>✗ trash can <u>across</u> ground. there is a liner.</div> <div>✗ trash can on <u>water</u>. there is a liner.</div> <div>✗ there is a liner.</div> <div>✗ trash can on ground. there is a <u>flare</u>.</div> <div>✗ trash can on <u>location</u>. there is a liner.</div>	#4 #5 #3 #6 #2 #1	#1 #2 #4 #6 #3 #5		<div>✓ flower in garden box; bench between box</div> <div>✗ <u>box</u> in garden <u>flower</u>:</div> <div>✗ bench between <u>flower</u>.</div> <div>✗ flower in garden <u>bench</u>:</div> <div>✗ box between <u>bench</u>.</div> <div>✗ flower in bench; <u>garden</u></div> <div>✗ <u>box</u> between box</div> <div>✗ <u>bench</u> in garden box;</div> <div>✗ <u>flower</u> between box</div> <div>✗ <u>garden box</u> in <u>flower</u>:</div> <div>✗ bench between box</div>	#3 #5 #4 #1 #1 #6 #2	#1 #5 #6 #3 #2 #6 #2
		<div>✓ A reporter interviews a policeman.</div> <div>✗ A <u>policeman</u> interviews a <u>reporter</u>.</div>	#2 #1	#1 #2		<div>✓ There are exactly 4 stars on flag.</div> <div>✗ There are exactly <u>2</u> stars on flag.</div>	#2 #1	#1 #2
	SugarCrepe		<div>✓ A cat sitting in front of a monitor that is displaying a picture of another cat.</div> <div>✗ A cat sitting in front of a monitor that is displaying <u>an animated</u> picture of another cat.</div>	#2 #1	#1 #2		<div>✓ This man is riding a skateboard behind a dog.</div> <div>✗ This <u>dog</u> is riding behind a <u>man</u> on a <u>skateboard</u>.</div>	#2 #1

Figure 7: Extended representative examples for Fig. 3, including additional examples from CREPE [36] and VALSE [39], as well as WhatsUp [23] and SugarCrepe [18]. These extended examples additionally include a broader range of benchmarks where applying the *reconstruction loss* proved effective.

SWAP

			w/o Alignment Loss	w/ Alignment Loss		w/o Alignment Loss	w/ Alignment Loss	
		<ul style="list-style-type: none">✓ A person in a green shirt stands by a child holding a piece of cake on a plate.✓ A person in a green shirt stands by a child holding a piece of cake on a plate.✗ A <u>child</u> in a green shirt stands by a <u>person</u> holding a piece of cake on a plate.	#2	#2		<ul style="list-style-type: none">✓ a black goat standing next to two white goats✓ A black goat is positioned next to two white goats.✗ Two <u>black</u> goats standing next to a <u>white</u> goat.	#1	#1
		<ul style="list-style-type: none">✓ Trays of pastries and sandwiches beside a bowl of soup.✓ The bowl of soup is placed beside trays of sandwiches and pastries.✗ A <u>bowl</u> of pastries and sandwiches beside <u>trays</u> of soup.	#3	#2		<ul style="list-style-type: none">✓ An open laptop and speakers on top of a desk.✓ Speakers and an open laptop are positioned on top of a desk.✗ <u>Speakers</u> on top of <u>an open laptop</u> on a desk.	#1	#1
		<ul style="list-style-type: none">✓ another horse standing and staring.✓ A horse is standing and staring while another two zebras are grazing.✗ Two zebras <u>running</u> while another horse standing and staring.	#1	#1		<ul style="list-style-type: none">✓ A kid wearing yellow is holding a pizza in a box.✓ A kid in a yellow outfit is holding a pizza in a box.✗ A kid wearing <u>green</u> is holding a pizza in a box.	#1	#1
		<ul style="list-style-type: none">✓ A person leaning up against a metal rail while holding a rainbow colored umbrella.✓ A person is positioned against a metal rail while holding a rainbow-colored umbrella.✗ <u>plastic</u> rail while holding a rainbow colored umbrella.	#1	#2		<ul style="list-style-type: none">✓ A person in grey shirt and hat sitting on a wooden bench.✓ A person wearing a grey shirt and a hat is sitting on a wooden bench.✗ A person in <u>pink</u> shirt and hat sitting on a wooden bench.	#3	#1

REPLACE

Figure 8: Extended representative examples for Fig. 4, including additional examples each category (SWAP and REPLACE) of SugarCrepe++. These extended examples further illustrate the effectiveness of applying the *alignment loss* across diverse cases.