

# GeneCIS: A Benchmark for General Conditional Image Similarity

Sagar Vaze<sup>1,2\*</sup>    Nicolas Carion<sup>1</sup>    Ishan Misra<sup>1</sup>  
<sup>1</sup> FAIR, Meta AI    <sup>2</sup> VGG, University of Oxford

Project Page: [sgvaze.github.io/genecis](https://sgvaze.github.io/genecis)

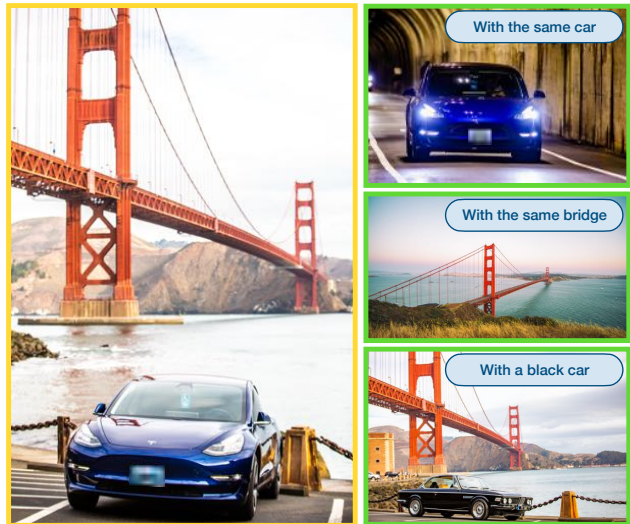
## Abstract

We argue that there are many notions of ‘similarity’ and that models, like humans, should be able to adapt to these dynamically. This contrasts with most representation learning methods, supervised or self-supervised, which learn a fixed embedding function and hence implicitly assume a single notion of similarity. For instance, models trained on ImageNet are biased towards object categories, while a user might prefer the model to focus on colors, textures or specific elements in the scene. In this paper, we propose the GeneCIS (‘genesis’) benchmark, which measures models’ ability to adapt to a range of similarity conditions. Extending prior work, our benchmark is designed for zero-shot evaluation only, and hence considers an open-set of similarity conditions. We find that baselines from powerful CLIP models struggle on GeneCIS and that performance on the benchmark is only weakly correlated with ImageNet accuracy, suggesting that simply scaling existing methods is not fruitful. We further propose a simple, scalable solution based on automatically mining information from existing image-caption datasets. We find our method offers a substantial boost over the baselines on GeneCIS, and further improves zero-shot performance on related image retrieval benchmarks. In fact, though evaluated zero-shot, our model surpasses state-of-the-art supervised models on MIT-States.

*We, the architects of the machine, must decide a-priori what constitutes its ‘world’; what things are to be taken as ‘similar’ or ‘equal’* — Karl Popper, 1963

## 1. Introduction

Humans understand many notions of similarity and choose specific ones depending on the task at hand [21, 58]. Consider the task of finding ‘similar’ images illustrated in Figure 1. Which of the rightmost images should be considered ‘most similar’ to the reference? Given different conditions, each image could be a valid answer. For instance, we may be interested in a specific object in the scene, focusing on either the ‘car’ or ‘bridge’. One could even indicate



**Figure 1.** Given different conditions (shown as blue text), different images on the right can be considered most ‘similar’ to the reference image on the left. We present a general way to train and evaluate models which can adapt to different notions of similarity.

a ‘negative’ similarity condition, specifying a *change* in the image to identify the bottom image as most similar.

Learning such similarity functions is a central goal in discriminative deep learning [11–13, 34, 63, 68, 75]. Discriminative models, either supervised [30, 75] or self-supervised [9, 10], learn embedding functions such that ‘similar’ images are closer in feature space than ‘dissimilar’ images. However, since there are infinitely many notions of image similarity, how do we allow our models to choose?

Almost all current approaches assume a single notion of similarity, either by explicitly training on a specific concept [68, 75] or through an implicit assumption in the underlying data distribution [9, 12]. Meanwhile, prior works tackling the conditional problem have focused on constrained domains such as fashion [69, 73] or birds [46], with a restricted set of similarity conditions. This is because developing and evaluating models that can adapt to generic notions of similarity is extremely challenging. Specifically, curating data to train and evaluate such models is difficult, as collecting annotations for all concepts of similarity is impossible.

\*Work done during an internship at Meta AI Research.

In this work we study the problem of general conditional image similarity, training on an open-set of similarity conditions, and evaluating on diverse similarity notions in a ‘zero-shot’ manner. We first design a benchmark comprising of *four evaluation datasets* for conditional image similarity, setting up conditional retrieval tasks. We define these tasks under a unified framework which spans practical use cases, and propose the benchmark as a sparse but broad coverage of the conditional similarity space. We propose these datasets for *zero-shot evaluation only*, and suggest that models which can perform well without fine-tuning can flexibly adapt to general notions of similarity, as desired. We name this benchmark GeneCIS (‘*genesis*’) for **General Conditional Image Similarity**. On GeneCIS, we find that baselines built from powerful CLIP backbones struggle and, moreover, that performance on it is only weakly correlated with the backbones’ ImageNet accuracy [17]. This is in contrast to popular vision tasks such as segmentation [39] and detection [45], underlining the benchmark’s utility.

We also propose a solution to training general conditional similarity models, based on parsing large-scale caption datasets [64, 66]. Rather than requiring exhaustive similarity annotations, we find that we can automatically mine this information from already abundant image-caption data. We show that training in this way offers substantial gains over the baselines, approaching (and in some cases surpassing) carefully designed specific solutions for each of the GeneCIS tasks. In addition, we demonstrate that our method scales with increasing amounts of caption data, suggesting promising directions for future work. Finally, on related benchmarks from the ‘Composed Image Retrieval’ (CIR) field [44, 74], we find our method provides gains over zero-shot baselines. In fact, our model outperforms state-of-the-art on the MIT-States benchmark [28], despite being evaluated zero-shot and never seeing the training data.

**Contributions.** (i) We present a framework for considering conditional image similarity, an important but understudied problem; (ii) We propose the GeneCIS benchmark to test models’ abilities to dynamically adapt to different notions of similarity; (iii) We show that current vision-language models like CLIP struggle on GeneCIS, and that performance on it is only weakly correlated with ImageNet accuracy; (iv) We design a scalable solution to the conditional similarity problem based on automatically parsing large-scale image-caption data; (v) We show our models provide substantial gains over zero-shot CLIP baselines; (vi) We validate our models on related CIR benchmarks, surpassing state-of-the-art on MIT-States despite zero-shot evaluation.

## 2. Related Work

Our thesis that the similarity between two images should be conditional is generally relevant to the *representation*

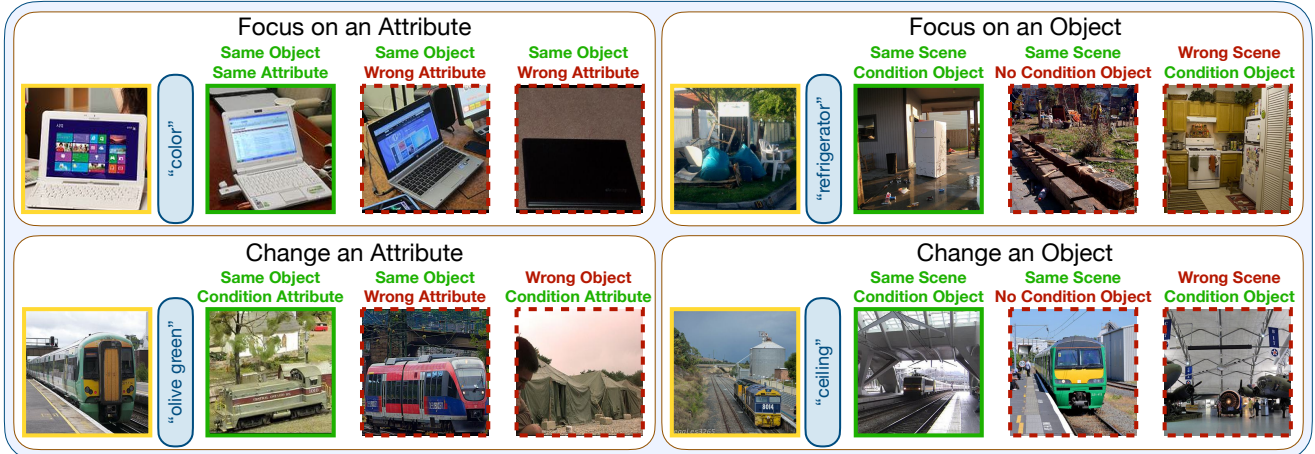
*learning* literature, which aims to learn embedding functions based on a single (often implicit) notion of similarity.

For instance, *deep metric learning* [30, 34, 63] aims to learn visual representations such that images from the same category are projected nearby in feature space. This idea is used in practical domains such as *image retrieval* [7, 59, 61], *face verification* [11, 67, 68] and *vehicle re-identification* [25, 31, 42]. The key limitation here is that networks are trained to encode a single notion of similarity, namely category-level similarity. While some work considered notions of similarity at different visual granularities [4, 15, 70], we posit that there exist concepts of similarity (e.g. shape and color) which are orthogonal to categories.

Meanwhile, *contrastive learning* [9, 10, 12, 13] defines notions of similarity by specifying a set of transformations to which the representation should be invariant (e.g. color jitter or random cropping), encouraging augmentations of the same instance to be embedded together. Similarly, *vision-language* contrastive training [29, 60] learns joint embedding spaces, where images’ representations are aligned with their paired captions. Though the precise notions of similarity are difficult to define in this case, we note that the embeddings are fundamentally unconditional, with a single deterministic embedding of a given image.

Finally, we highlight three relevant sub-fields in the literature: *conditional similarity networks* (CSNs); *compositional learning* (CL); and *composed image retrieval* (CIR). CSNs are networks with multiple subspaces for different notions of similarity [73]. Though their motivation is highly related to our work, CSNs are trained in a supervised manner with pre-defined similarity conditions [41, 46, 73], and/or are evaluated in constrained domains such as fashion [32, 69]. In contrast, we aim to train on an open-set of similarity conditions and evaluate zero-shot on natural images. Meanwhile, our work is related to CL research in that we seek to compose information from images and conditions to establish similarities. However, again, CL models are often assessed on their ability to recognize unseen combinations of a finite set of visual primitives [47, 54, 56]. Lastly, the most similar setup to GeneCIS is proposed in the recent CIR [74]. It tackles the problem of composing an image and text prompt to retrieve relevant images from a gallery [1, 3, 16]. This is typically posed in the context of fashion [23, 76], with the text prompt acting as an image edit instruction (e.g. ‘the same dress but in white’ [1]). As such, CIR tackles a subset of the conditional similarity problem, by presenting models with a ‘negative’ similarity condition.

**Key similarities and differences with prior work:** In this work, we leverage CIR [44] and MIT-States [28] (natural image CIR datasets) for additional evaluations, and further leverage the ‘Combiner’ architecture [3] to compose text conditions and image features. Broadly speaking, our work differs from CSNs, CL and CIR in that we do not



**Figure 2. The GeneCIS benchmark** contains four evaluation tasks for conditional similarity, where the goal is to retrieve the most similar image from a gallery (right, green squares), given a reference (left, yellow squares), and condition (blue ovals). Each task explores one combination of ‘focus’/‘change’ an ‘attribute’/‘object’. All galleries contain ‘distractors’ (dashed, dark-red squares) which are *implicitly* similar to the reference or condition. Thus, given a reference and explicit condition, GeneCIS evaluates models’ ability to select the *most conditionally similar* gallery image. Note: We show three gallery images for clarity, though all GeneCIS galleries have 10-15 images.

train on a finite, closed-set of similarity conditions or visual primitives. Instead, we train models on open-world image-caption data, and demonstrate a flexible understanding of conditional similarity through zero-shot evaluation on a range of similarity conditions in natural images.

### 3. Conditional Similarity

We now describe our setup for the conditional similarity problem and its associated challenges – both with benchmarking models and acquiring data to train them. In § 4 we introduce the GeneCIS benchmark which measures important aspects of the problem. In § 5, we present a scalable solution to automatically acquire training data from widely available image-caption datasets.

**Problem Definition:** We define the problem of conditional similarity as learning a similarity function between two images given an *explicit* condition:  $f(I^T; I^R, c)$  yields the scalar similarity between a target image,  $I^T$ , and a reference image,  $I^R$ , given some external condition,  $c$ . We use the scalar  $f(\cdot)$  to find the most conditionally similar image from a target set, *i.e.*, to solve a retrieval task. In this work we consider the condition to be a user-specified text prompt, although other types of condition are possible. We highlight that standard image similarity, framed as  $f(I^T, I^R)$ , *implicitly* assumes a similarity condition, often incorporated into the model or dataset (see § 2). We refer to the case where images are similar under an unspecified condition as the images being *implicitly similar*.

#### 3.1. Challenges in training and evaluation

**Challenges in evaluation:** The key difficulty in evaluating conditional similarity is that there are infinitely many possible conditions: from ‘images with the same top-left pixel

value are similar’ to ‘the same image but upside down is similar’. Thus, it is impossible to evaluate models’ ability to adapt to *every* similarity condition. Instead, in § 4, we introduce the GeneCIS benchmark which consists of a subset of such conditions, and covers a broad range of practical use cases. We suggest that models which produce *zero-shot* gains across GeneCIS, without finetuning, are more capable of flexibly adapting to different notions of similarity.

**Challenges in acquiring training data:** Since the space and diversity of similarity conditions is huge, acquiring human annotations to train for *every* type of conditional similarity is not feasible. For instance, to train a function which is sensitive to object category given some conditions (*e.g.*, ‘car’ or ‘bridge’ objects in Figure 1), and ‘color’ given others (*e.g.* ‘blue’ or ‘black’ car in Figure 1), we need training data containing both features. Prior work addresses this by dramatically restricting the space of conditions and training on human annotations for pre-defined notions of similarity [46, 73]. In § 5, we describe an automatic method which leverages existing large-scale image-text datasets to learn an open-set of similarity conditions. The resulting model can be evaluated in a zero-shot manner across different types of conditional similarity task.

### 4. The GeneCIS Benchmark

GeneCIS considers two important dimensions of the conditional similarity problem. Firstly, a user may be interested in an *object* in the scene (‘with the same car’) or an *attribute* of a given object (‘the same color as the car’). Secondly, the condition could either *focus* on a particular aspect of the image (‘the same color as the car’) or specify the ‘negative’ space of a similarity condition, by defining a *change* in the image (‘this car but in black’).

We propose **four evaluation tasks in GeneCIS**, that covers the combination of the above dimensions and hence a diverse range of conditional similarities. For each of the tasks, we construct retrieval problems with: a reference image,  $I^R$ ; a text condition,  $c$ ; and a retrieval gallery of  $M$  target images,  $\{I_i^T\}_{i=1}^M$ , of which only one is ‘correct’ or ‘positive’. The task is to identify which of the target images is most similar to the reference, given the condition. The retrieval tasks, illustrated in Figure 2 with more examples in Appendix G.1, are:

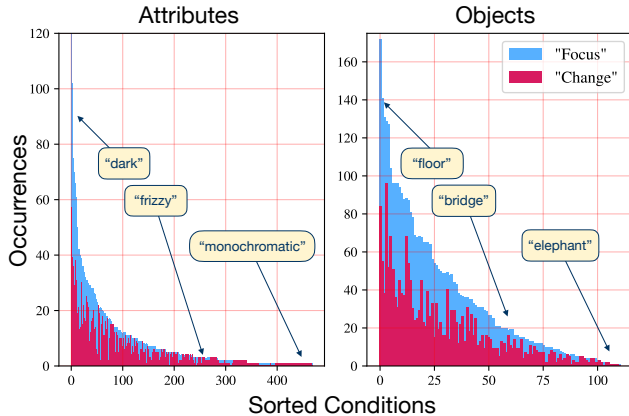
- **Focus on an Attribute:** This task evaluates a model’s ability to focus on a specific attribute type (e.g. ‘color’ or ‘material’). For instance, in Figure 2, we see a white laptop and the condition ‘color’, with the task being to select the laptop with the same color from the gallery.
- **Change an Attribute:** This task contains ‘negative’ similarity conditions, considering target images with a specific attribute changed to be most similar. In Figure 2, the aim is to retrieve the same object (‘train’) but with the color changed from ‘green’ to ‘olive green’.
- **Focus on an Object:** This task considers reference images with many objects, and we refer to the set of objects together as a proxy for the image ‘scene’. The condition selects a single object from the reference as the most important (e.g. ‘refrigerator’ in Figure 2) and the ‘positive’ target contains the condition object as well as the same ‘scene’ (e.g. also contains ‘sky’, ‘chair’ etc. in Figure 2).
- **Change an Object:** This task considers ‘negative’ similarity through conditions which specify an object to be added to a scene. For instance, in Figure 2, ‘ceiling’ is specified, with the aim being to retrieve the same scene (a train station) but with a ceiling also present.

The tasks in GeneCIS are designed to be diverse and challenging for a single model while remaining well-posed. In Figure 2, given only the reference image,  $I^R$ , and text condition,  $c$ , a human can readily identify which of the target images is most ‘similar’. We wish to benchmark vision models’ competency at the same task.

For the benchmark to be challenging, we would want the model to need both the image content and the text condition to solve the problem. Thus, we include different forms of ‘distractor’ images in the galleries. For instance, for tasks with objects in the condition, we include distractors which have a similar ‘scene’ to the reference but do not contain the condition object. Such distractors are likely to affect models which are over-reliant on information from the reference image, without considering the condition. Similarly, we include distractors which contain the object specified in the condition, but not the reference scene, confusing models which solely rely on the condition. Meanwhile, for the attribute-based tasks, we include distractors which contain the reference object category, but not the correct attribute,

**Table 1. Statistics** of the four tasks in the GeneCIS benchmark.

Name	Base Dataset	# Templates	# Gallery Images
Focus on an Attribute	VAW [56]	2000	10
Change an Attribute	VAW [56]	2112	15
Focus on an Object	COCO [36]	1960	15
Change an Object	COCO [36]	1960	15

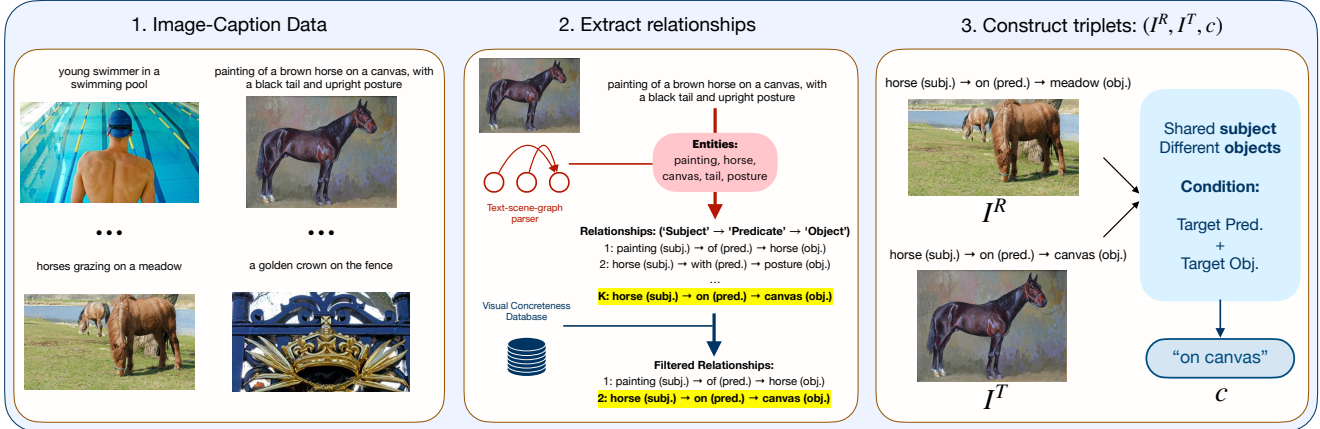


**Figure 3. Distribution of conditions** for attribute- and object-based conditions. For ‘Focus on an Attribute’, we show the distribution of the common attribute between the reference and positive target image (the condition itself is an attribute type, e.g. ‘color’).

and vice-versa. As such, many targets are *implicitly similar* to the reference (similar given some condition), but the positive image is the most similar *given the provided condition*.

**Benchmark Details:** We construct all tasks by repurposing existing public datasets. For the two tasks which ‘focus on’ and ‘change’ *attributes*, we leverage the VAW dataset [56], which inherits from Visual Genome [37]. From VAW, we extract crops for individual objects and, for each object, use annotations for: object category; positively labelled attributes (which the object definitely possesses); and negatively labelled attributes (which the object definitely does not possess). For the two tasks which ‘focus on’ or ‘change’ *objects*, we use COCO Panoptic Segmentation data [36, 40] containing dense category annotations for every pixel in the image. We give full details of the template construction process for each task in Appendix A.1.

We show statistics of the evaluations in Table 1, including the number of retrieval templates and number of gallery images. We note that we carefully construct the benchmarks such that there is only one ‘positive’ image among the targets, with gallery sizes of between 10 and 15 images. This is different to many ‘text-to-image’ or ‘image-to-text’ retrieval benchmarks [40, 57], which contain galleries with thousands of targets. Though larger galleries increase the tasks’ difficulty, the galleries inevitably contain some valid targets which are treated as negative. We further show the distribution of objects and attributes specified in the conditions in Figure 3, noting that our space of conditions spans a long-tail of over 400 attributes and 100 objects.



**Figure 4. Method overview.** Our method for training general conditional similarity functions extracts information from large-scale image-caption datasets (left). We extract ‘Subject’ → ‘Predicate’ → ‘Object’ relationships from the caption data (middle), before using them to construct training triplets where a *reference* and *target* image are related by a *condition* (right).

**Noise and human verification:** Though, in principle, our benchmark should be error free, manual inspection of the templates shows that noise is introduced through underlying inconsistencies in Visual Genome [37], VAW [56] and COCO [36]. We are currently in the process of collecting manual annotations and human verification of the templates, and present the current version as ‘GeneCIS v0’.

## 5. Method

In § 5.1, we briefly describe preliminaries for our approach to learning general conditional similarity functions. This includes the model architecture and optimization objective which we inherit from prior work [3]. In § 5.2, we describe our main methodological contribution: an automatic and scalable way of mining conditional similarity training data from widely available image-caption datasets.

### 5.1. Preliminaries

**Training data.** To learn a conditional similarity function  $f(\cdot)$ , we train with triplets  $(I^R, I^T, c)$ , where  $I^R$  and  $I^T$  are termed reference and target images, and  $c$  is the condition defining a relationship between them.

**Model Architecture** We parametrize the conditional similarity function  $f(\cdot)$  with deep networks, first encoding features for  $(I^R, I^T, c)$  as  $(\mathbf{x}^R, \mathbf{x}^T, \mathbf{e}) \in \mathbb{R}^D$ . We learn separate encoders,  $\Phi(I)$  and  $\Psi(c)$ , for the images and text condition. Next, we train a ‘Combiner’ network [3], which composes the reference image features with the condition text features as  $g(\mathbf{x}^R, \mathbf{e}) \in \mathbb{R}^D$ . Finally, we consider the scalar conditional similarity to be the dot product between the combined feature, and target image feature, as:  $f(I^T; I^R, c) = g(\mathbf{x}^R, \mathbf{e}) \cdot \mathbf{x}^T$ . Details of the Combiner architecture can be found in Appendix D and [3].

We initialize our image and text backbones,  $\Phi(\cdot)$  and  $\Psi(\cdot)$ , with CLIP [60]. CLIP models are pre-trained on

400M image-text pairs containing a range of visual concepts. Furthermore, the visual and text embeddings from CLIP are aligned, making it easier to learn the composition between reference image and conditioning text features.

**Optimisation Objective** Given a batch of triplets,  $B = \{(I_i^R, I_i^T, c_i)\}_{i=1}^{|B|}$ , we get features as  $\{(\mathbf{x}_i^R, \mathbf{x}_i^T, \mathbf{e}_i)\}_{i=1}^{|B|}$ . Then, given a temperature  $\tau$ , we optimise  $(\Phi, \Psi, g)$  with a contrastive loss [50], as:

$$\mathcal{L} = -\frac{1}{|B|} \sum_{i \in B} \log \frac{\exp(g(\mathbf{x}_i^R, \mathbf{e}_i) \cdot \mathbf{x}_i^T / \tau)}{\sum_{j \in B} \exp(g(\mathbf{x}_i^R, \mathbf{e}_i) \cdot \mathbf{x}_j^T / \tau)} \quad (1)$$

### 5.2. Scalable training for conditional similarity

To train for general conditional similarity, we wish to curate triplets for training,  $\mathcal{D}_{train} = \{(I_i^R, I_i^T, c_i)\}_{i=1}^N$ , with diverse conditions and concepts of similarity. However, as the space of conditions increases, the burden for exhaustively annotating such a dataset increases exponentially. Instead, our method (illustrated in Figure 4) automatically mines training triplets from existing data sources:

**Image-caption Data:** We begin with large-scale image-caption data scraped from the internet, containing images paired with descriptive captions [51, 66]. We hope that the captions contain information about the objects and attributes in the image, which we can utilize for the conditional similarity task. We also hope that such a method can scale with increasing data in the same way that conventional representation learning algorithms do.

**Extract relationships:** We use an off-the-shelf text-to-scene-graph parser [65, 77] to identify ‘Subject’ → ‘Predicate’ → ‘Object’ relationships within the caption [55]. For instance, from the central image in Figure 4, we extract the highlighted relationship ‘Horse’ → ‘on’ → ‘Canvas’. Note that one caption may contain many such relationships.

We find that many of the entities (‘Subjects’ or ‘Objects’) extracted by the parser are not visually grounded in

the image, *e.g.*, pronouns (‘I’, ‘you’) or time-based nouns (‘today’, ‘yesterday’). To address this, we introduce an additional filtering step, where every entity is scored for ‘visual concreteness’ based on a pre-existing database [8]. The database contains human ratings between 1 and 5 for how visually apparent a noun is. For each extracted relationship, we average its ‘Subject’ and ‘Object’ concreteness scores, discarding relationships if their value is below a threshold.

**Construct triplets:** We first randomly select a relationship, taking the image it comes from as the ‘reference’,  $I^R$ . Having identified the *subject* of the relationship (*e.g.* ‘Horse’ in the rightmost column of Figure 4) we identify all other relationships in the dataset containing the same subject. From this restricted pool of relationships, we randomly sample a ‘target’ relationship and image,  $I^T$ , with the same subject but a different *object* (*e.g.* a horse on a ‘canvas’ instead of in a ‘meadow’ in Figure 4). Finally, we define the *condition* of the triplet,  $c$ , as the concatenated ‘Predicate’ and ‘Object’ from the target relationship (‘on canvas’ in Figure 4).

**Discussion:** We note that our mined triplets exhibit a bias towards the ‘Change an Object’ GeneCIS task. However, the triplets often involve abstract relationships between reference and target images (*e.g.* ‘Horse on canvas’ in Figure 4). As such, solving the training task requires the model to use the condition to extract and modify diverse forms of information from the reference, which is the central requirement of the broader conditional similarity problem.

## 6. Main Experiments

We evaluate baselines, task-specific solutions, and our method on the proposed GeneCIS benchmark. § 6.1 describes the baselines as well as specific solutions which we design for each of the GeneCIS tasks. § 6.3 shows results on GeneCIS and, in § 6.4, we evaluate on related benchmarks from the Composed Image Retrieval (CIR) literature.

### 6.1. Baselines and Specific Solutions for GeneCIS

**CLIP-Only Baselines:** We provide three simple CLIP-only [60] baselines for GeneCIS. Our **Image Only** baseline embeds all images with the CLIP image encoder and retrieves the closest gallery image to the reference. The **Text Only** baseline embeds the text condition with the CLIP text encoder, and the gallery images with the image encoder, and finds the closest gallery image to the text embedding. Finally, our **Image + Text** baseline averages the reference image with the condition text feature, before using the combined vector to find the closest gallery image.

**CIRR Combiner baseline:** CIRR is a natural image dataset [44] containing 28K curated retrieval templates. All templates contain a human-specified text condition defining the relationship between the reference and ‘positive’ target image. Unlike our automatic and scalable triplet mining

method, CIRR is manually constructed with a lengthy annotation process. We include a baseline from [3], which trains a Combiner model with a CLIP backbone on CIRR. For fair comparison with our method, we fine-tune both the image and text backbones on CIRR before evaluating the model zero-shot on GeneCIS, terming it **Combiner (CIRR)**.

**Specific Solutions:** We also design specific solutions for each of the proposed tasks in GeneCIS. These solutions take into account the construction mechanisms of each task and represent sensible approaches to tackling the tasks independently. We design all solutions to respect the zero-shot nature of the evaluations and hence they are all based on ‘open-vocabulary’ models; we use CLIP for the attribute-based tasks and Detic [81] for the object-based ones. For the attribute-based tasks, we use CLIP to predict attributes or categories in the reference image, before using text embeddings of these predictions to search the gallery. For the object-based tasks, we use Detic to detect the object categories present in all images, treating the detected categories as bag-of-word descriptors of the target images. We give full details of the specific solutions in Appendix B.

### 6.2. Implementation Details

We train our strongest model on 1.6M triplets mined from Conceptual Captions 3 Million (CC3M) [66] which contains 3M image-caption pairs. Each triplet has a visual concreteness of at least 4.8 averaged over the ‘Subject’ and ‘Object’ entities in both the reference and target image. We train the contrastive loss with temperature  $\tau = 0.01$  and batch size of 256, training for 28K gradient steps. We use early stopping based on the Recall@1 on the CIRR validation set and, for fair comparison with [3], initialize the image and text backbones with the ResNet50×4 CLIP model. Further details are in Appendix E.

### 6.3. Analysis on GeneCIS

We report results for all methods on the GeneCIS benchmark in Table 2. Our evaluation metric is Recall@ $K$ : the frequency with which the model ranks the ‘correct’ gallery image in its top- $K$  predictions. We report results at  $K = \{1, 2, 3\}$  to evaluate under different constraints, and to account for any noise in the benchmark. We also report the Average R@1 over all tasks to measure the overall performance across different forms of conditional similarity.

**Takeaways:** From the *baselines* we find that both the ‘Image Only’ and ‘Text Only’ models perform poorly as expected, since they only rely on either the reference image content or the text condition. The ‘Image + Text’ and ‘Combiner (CIRR)’ models perform better, validating our claim that both the reference and text condition are required to solve the task. Phrased differently, this suggests the benchmark evaluates conditional similarity, as implicit similarity functions (*e.g.* the ‘Image Only’ baseline) perform poorly

**Table 2. Evaluation on GeneCIS.** We evaluate baselines and our method. We also evaluate specific solutions for each task (shown gray, these are not general conditional similarity functions and hence cannot be evaluated on all tasks). Both across ten random seeds, and with ten cross-validation splits, we find a standard deviation of  $\approx 0.2\%$  in our model’s R@1 on each task, as well as on average over all tasks.

	Focus Attribute			Change Attribute			Focus Object			Change Object			Average R@1
	R@1	R@2	R@3	R@1	R@2	R@3	R@1	R@2	R@3	R@1	R@2	R@3	
Specific Solution (Focus Attribute)	20.8	32.6	41.1	-	-	-	-	-	-	-	-	-	-
Specific Solution (Change Attribute)	-	-	-	15.2	25.8	35.6	-	-	-	-	-	-	-
Specific Solution (Object)	-	-	-	-	-	-	18.7	30.3	37.4	18.1	28.7	34.5	-
Image Only	17.7	30.9	<b>41.9</b>	11.9	20.8	28.8	9.3	18.2	26.2	7.2	16.7	24.9	11.5
Text Only	10.2	20.5	29.6	9.5	17.6	26.4	6.5	16.8	22.4	6.2	13.9	21.4	8.1
Image + Text	15.6	26.3	37.1	12.6	22.9	32.0	10.8	21.0	31.2	11.3	21.5	30.3	12.6
Combiner (CIRR)	15.1	27.7	39.8	12.1	22.8	31.8	13.5	25.4	<b>36.7</b>	15.4	28.0	39.6	14.0
Combiner (CC3M, Ours)	<b>19.0</b>	<b>31.0</b>	41.5	<b>16.6</b>	<b>27.5</b>	<b>36.5</b>	<b>14.7</b>	<b>25.9</b>	36.1	<b>16.8</b>	<b>29.1</b>	<b>39.7</b>	<b>16.8</b>

**Table 3. Results on MIT-States [28].** *Zero-shot evaluation* of our model outperforms SoTA supervised methods on this dataset.

	Zero-shot	Recall @ 1	Recall @ 5	Recall @ 10
TIRG [74]	$\times$	12.2	31.9	43.1
ComposeAE [1]	$\times$	13.9	35.3	47.9
LBF [27]	$\times$	14.7	35.3	46.6
HCL [79]	$\times$	15.2	36.0	46.7
MAN [19]	$\times$	15.6	36.7	47.7
Image Only	$\checkmark$	3.7	14.1	22.9
Text Only	$\checkmark$	9.5	22.5	31.4
Image + Text	$\checkmark$	13.3	31.7	42.6
Combiner (CC3M, Ours)	$\checkmark$	<b>15.8</b>	<b>37.5</b>	<b>49.4</b>

on average. We further find that *our method*, using automatically mined data, substantially outperforms all baselines on average across the tasks, as well as at Recall@1 on all tasks individually. Notably, it outperforms the model trained on manually collected data from CIRR.

As expected, most per-task *specific solutions* perform better than our general method. However, the broad zero-shot nature of GeneCIS makes all tasks independently challenging and the specific solutions do not work for all of them. Broadly speaking, we found that CLIP [60] struggles to predict object attributes, and that Detic [81] struggles on the ‘stuff’ categories in COCO Panoptic [36].

Finally, *caveats* can be found in ‘Image Only’ results on ‘Focus Attribute’, where the baseline performs slightly better than our method at higher recalls. This is because there are some similarity conditions (e.g. ‘color’) for which standard image embeddings are well suited. We also find that ‘Combiner (CIRR)’ performs better on tasks with object conditions, as the multi-object image distribution of CIRR is more closely aligned with these tasks, than with the single-object images in the attribute-based tasks. We note that good performance on all tasks collectively indicates strong general conditional similarity models.

## 6.4. Comparisons to Prior Work

GeneCIS uses natural images with general conditions, rather than being specialized to domains such as bird species [46], faces [80] or fashion compatibility [23, 24, 71, 76]. As such, to find comparable existing benchmarks, we turn to the *Composed Image Retrieval* (CIR) literature. The CIR task is to retrieve images which best match a com-

**Table 4. Results on CIRR [44].** Our model substantially outperforms the comparable zero-shot baselines.

	Zero-shot	Recall @ 1	Recall @ 5	Recall @ 10
ARTEMIS [16]	$\times$	17.0	46.1	61.3
CIRPLANT [44]	$\times$	19.6	52.6	68.4
Combiner (CIRR, [3])	$\times$	38.5	70.0	81.9
Combiner (CIRR, improved)	$\times$	40.9	73.4	84.8
Image Only	$\checkmark$	7.5	23.9	34.7
Text Only	$\checkmark$	20.7	43.9	56.1
Image + Text	$\checkmark$	21.8	50.9	63.7
Combiner (CC3M, Ours)	$\checkmark$	<b>27.3</b>	<b>57.0</b>	<b>71.1</b>

posed reference image and editing text condition. This task aligns with the ‘Change’ dimension of GeneCIS. We evaluate on both the MIT-States benchmark [28] as well as on CIRR [44], with the former precisely reflecting the ‘Change Attribute’ GeneCIS task.

**Metrics:** On both benchmarks, we evaluate our model *zero-shot* on the test-sets and compare with prior work trained on the datasets. These datasets are partially labeled and evaluate using a global retrieval setting, *i.e.*, the entire test-set is used as a gallery for each query. Thus, we follow prior work and report Recall@K at multiple  $K = \{1, 5, 10\}$  to fully capture the model’s performance.<sup>1</sup>

**Results:** We show results on MIT-States in Table 3. Prior work on this benchmark trains models on the dataset from scratch and thus is not zero-shot. Nonetheless, *zero-shot evaluation* of our model surpasses state-of-the-art on this task. However, we note that prior methods use smaller models compared to our pre-trained CLIP backbone.

We report on CIRR in Table 4, evaluating through the official test server and again comparing to methods that train for this setting. We report results for the Combiner method from the paper [3] as well as our improved implementation (see § 6.1), which are both trained on CIRR. Our improved implementation is a strong upper bound, surpassing previous fully supervised models. On zero-shot evaluation, our method surpasses the comparable baselines by a significant margin across all the recall metrics. Compared to supervised methods, our model outperforms [16] and [44] zero-shot, though we note [16] trains from scratch. Finally, our

<sup>1</sup>CIRR also has an evaluation on curated galleries, akin to GeneCIS. We do not report on this as we found that the ‘Text Only’ baseline performed comparably with SoTA models on this task, achieving over 60% Recall@1.

**Table 5. Ablations** of key design choices of our full model with results reported on our GeneCIS benchmark.

	Average Recall @ 1
<b>Full Model</b>	<b>16.8</b>
No filtering for visual concreteness	15.0
Freezing CLIP image backbone	14.7
Freezing CLIP text backbone	15.8
Freezing entire backbone	15.1
Training on SBU [51] instead of CC3M [66] caption data	16.5

model reduces the gap between the baselines and specialist Combiner models trained on CIRR.

## 7. Analysis

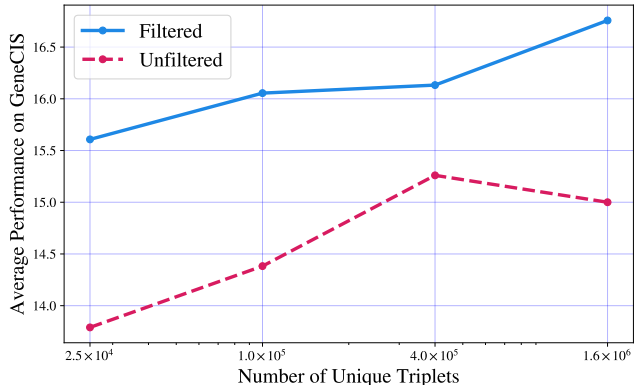
**Ablations:** Table 5 shows the effect of our design choices on the performance on GeneCIS. We find that filtering out relationships which are not visually concrete, and finetuning the entire backbone, both strongly affect the performance. We verify the robustness of our triplet mining procedure by training with SBU Captions [51], a smaller but different source of image-caption data. We find that though the larger CC3M [66] produces slightly better results, different image-caption datasets are also suitable.

**Comparing pretrained backbones:** In Figure 6, we study the effect of changing the CLIP initialization. We train Combiner models with ResNet [26] and ViT [18] backbones on CC3M, showing their performance as well as the ‘Image + Text’ baseline from § 6.1.<sup>2</sup>

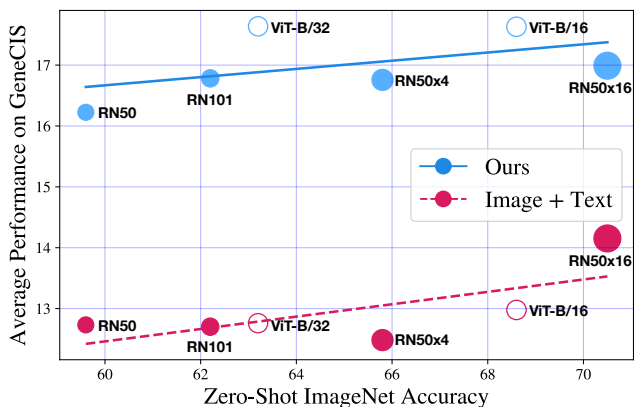
We plot the performance on GeneCIS against the CLIP backbone’s zero-shot ImageNet accuracy [17]. We observe that the performance on GeneCIS is **weakly correlated with the ImageNet performance** of the backbone: a Top-1 gain of 10% on ImageNet leads to only 1% improvement on GeneCIS. This suggests that improvements on ImageNet do not directly transfer to GeneCIS and that GeneCIS measures a different yet important capability of vision models. In addition, our method offers a substantial boost over the ‘Image + Text’ baseline, and a greater boost than scaling the underlying CLIP model. Both of these results are in stark contrast to trends on popular vision tasks such as segmentation [39] and detection [45], where gains on ImageNet directly transfer to large gains on the downstream task, and often more significantly so than gains from the underlying method.

**Scaling the number of triplets:** In Figure 5, we investigate the effect of scaling the conditional similarity *training data*. We successively decrease the number of mined triplets by factors of four (from the 1.6M used to train our strongest models) both with and without concreteness filtering. We find results improve with increasing numbers of triplets and that while our models are trained on a dataset of 3M image-caption pairs [66], open-source caption datasets exist with up to five billion images [64]. We emphasize the utility of

<sup>2</sup>For fair comparison with [3], we report with a ResNet50×4 backbone in Table 2, and report on our strongest ViT-B/16 model in Appendix C.



**Figure 5. Scaling the number of mined triplets** used for training our model improves the performance. This suggests that our automatic mining strategy is a promising and scalable approach to learning general similarity functions.



**Figure 6. Impact of different CLIP backbones** on the performance of our model and the ‘Image + Text’ baseline. We show the Average Recall@1 on GeneCIS against the backbones’ zero-shot ImageNet accuracy, showing the two have a weak correlation.

this finding, suggesting it is possible to train stronger conditional similarity models by further scaling the training data.

## 8. Conclusion

In this paper we have proposed the GeneCIS benchmark for General Conditional Image Similarity, an important but understudied problem in computer vision. The benchmark extends prior work and evaluates an open-set of similarity conditions, by being designed for zero-shot testing only. Furthermore, we propose a way forward for scalably training conditional similarity models, which mines information from widely available image-caption datasets.

Our method not only boosts performance over all baselines on GeneCIS, but also provides substantial zero-shot gains on related image retrieval tasks. Moreover, we find that unlike for many popular vision tasks, the performance of our models on GeneCIS is roughly decorrelated from scaling the backbone network’s ImageNet accuracy, motivating further study of the conditional similarity problem.



## References

- [1] Muhammad Umer Anwaar, Egor Labintsev, and Martin Kleinsteuber. Compositional learning of image-text query for image retrieval. In *WACV*, 2021. 2, 7
- [2] Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *CVPR*, 2022. 15
- [3] Alberto Baldrati, Marco Bertini, Tiberio Uricchio, and Alberto Del Bimbo. Conditioned and composed image retrieval combining and partially fine-tuning clip-based features. In *CVPRW*, 2022. 2, 5, 6, 7, 8, 14
- [4] Maxim Berman, Hervé Jégou, Andrea Vedaldi, Iasonas Kokkinos, and Matthijs Douze. Multigrain: a unified image embedding for classes and instances. *arXiv*, 2019. 2
- [5] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions. *ECCV*, 2022. 15
- [6] Andrew Brown, Cheng-Yang Fu, Omkar Parkhi, Tamara L Berg, and Andrea Vedaldi. End-to-end visual editing with a generatively pre-trained artist. *ECCV*, 2022. 15
- [7] Andrew Brown, Weidi Xie, Vicky Kalogeiton, and Andrew Zisserman. Smooth-ap: Smoothing the path towards large-scale image retrieval. In *ECCV*, 2020. 2
- [8] Marc Brysbaert, Amy Beth Warriner, and Victor Kuperman. Concreteness ratings for 40 thousand generally known english word lemmas. In *Behavior research methods*, 2014. 6
- [9] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *NeurIPS*, 2020. 1, 2
- [10] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021. 1, 2
- [11] Jun-Cheng Chen, Vishal M. Patel, and Rama Chellappa. Unconstrained face verification using deep cnn features. In *WACV*, 2016. 1, 2
- [12] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 1, 2
- [13] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv*, 2020. 1, 2
- [14] Katherine Crowson, Stella Biderman, Daniel Kornis, Dashiell Stander, Eric Hallahan, Louis Castriaco, and Edward Raff. Vqgan-clip: Open domain image generation and editing with natural language guidance. In *ECCV*. Springer, 2022. 15
- [15] Yin Cui, Zeqi Gu, Dhruv Mahajan, Laurens Van Der Maaten, Serge Belongie, and Ser-Nam Lim. Measuring dataset granularity. *arXiv*, 2019. 2
- [16] Ginger Delmas, Rafael S Rezende, Gabriela Csorka, and Diane Larlus. Artemis: Attention-based retrieval with text-explicit matching and implicit similarity. In *ICLR*, 2022. 2, 7
- [17] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 2, 8
- [18] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2020. 8, 14
- [19] Zhixiao Fu, Xinyuan Chen, Jianfeng Dong, and Shouling Ji. Multi-order adversarial representation learning for composed query image retrieval. In *ICASSP*, 2021. 7
- [20] Rinon Gal, Or Patashnik, Haggai Maron, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. Stylegan-nada: Clip-guided domain adaptation of image generators. *ACM Transactions on Graphics (TOG)*, 2022. 15
- [21] Robert L. Goldstone and Ji Yun Son. *155 Similarity*. Oxford University Press, 2012. 1
- [22] Agrim Gupta, Piotr Dollar, and Ross Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *CVPR*, 2019. 15
- [23] Xintong Han, Zuxuan Wu, Phoenix X. Huang, Xiao Zhang, Menglong Zhu, Yuan Li, Yang Zhao, and Larry S. Davis. Automatic spatially-aware fashion concept discovery. In *ICCV*, 2017. 2, 7
- [24] Xintong Han, Zuxuan Wu, Yu-Gang Jiang, and Larry S Davis. Learning fashion compatibility with bidirectional lstms. In *ACM*, 2017. 7
- [25] Bing He, Jia Li, Yifan Zhao, and Yonghong Tian. Part-regularized near-duplicate vehicle re-identification. In *CVPR*, June 2019. 2
- [26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 8
- [27] Mehrdad Hosseinzadeh and Yang Wang. Composed query image retrieval using locally bounded features. In *CVPR*, 2020. 7
- [28] Phillip Isola, Joseph J. Lim, and Edward H. Adelson. Discovering states and transformations in image collections. In *CVPR*, 2015. 2, 7
- [29] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, 2021. 2
- [30] Mahmut Kaya and Hasan Şakir Bilge. Deep metric learning: A survey. *Symmetry*, 11, 2019. 1, 2
- [31] Sultan Daud Khan and Habib Ullah. A survey of advances in vision-based vehicle re-identification. *Computer Vision and Image Understanding*, 2019. 2
- [32] Donghyun Kim, Kuniaki Saito, Samarth Mishra, Stan Sclaroff, Kate Saenko, and Bryan A. Plummer. Self-supervised visual attribute learning for fashion compatibility. *ICCV Workshops*, 2021. 2
- [33] Gwanghyun Kim, Taesung Kwon, and Jong Chul Ye. Diffusionclip: Text-guided diffusion models for robust image manipulation. In *CVPR*, 2022. 15
- [34] Sungyeon Kim, Dongwon Kim, Minsu Cho, and Suha Kwak. Proxy anchor loss for deep metric learning. In *CVPR*, 2020. 1, 2

- [35] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 14
- [36] Alex Kirillov, Tsung-Yi Lin, Holger Caesar, Ross Girshick, and Piotr Dollár. Microsoft coco: Panoptic segmentation challenge, 2017. 4, 5, 7, 12, 13, 14
- [37] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, Michael Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations, 2016. 4, 5, 13
- [38] Gihyun Kwon and Jong Chul Ye. Clipstyler: Image style transfer with a single text condition. In *CVPR*, 2022. 15
- [39] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and René Ranftl. Language-driven semantic segmentation. *ICLR*, 2022. 2, 8
- [40] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context. In *ECCV*, 2014. 4, 12, 13
- [41] Yen-Liang Lin, Son Tran, and Larry S. Davis. Fashion outfit complementary item retrieval. In *CVPR*, June 2020. 2
- [42] Xinchen Liu, Wu Liu, Huadong Ma, and Huiyuan Fu. Large-scale vehicle re-identification in urban surveillance videos. In *ICME*, 2016. 2
- [43] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 15
- [44] Zheyuan Liu, Cristian Rodriguez, Damien Teney, and Stephen Gould. Image retrieval on real-life images with pre-trained vision-and-language models. In *ICCV*, 2021. 2, 6, 7, 14
- [45] Matthias Minderer, Alexey Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, et al. Simple open-vocabulary object detection with vision transformers. *ECCV*, 2022. 2, 8
- [46] Samarth Mishra, Zhongping Zhang, Yuan Shen, Ranjitha Kumar, Venkatesh Saligrama, and Bryan A. Plummer. Effectively leveraging attributes for visual similarity. In *ICCV*, 2021. 1, 2, 3, 7
- [47] Ishan Misra, Abhinav Gupta, and Martial Hebert. From red wine to red tomato: Composition with context. In *CVPR*, 2017. 2
- [48] Andrei Neculai, Yanbei Chen, and Zeynep Akata. Probabilistic compositional embeddings for multimodal image retrieval. In *CVPR*, 2022. 15
- [49] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv*, 2021. 15
- [50] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv*, 2018. 5
- [51] Vicente Ordonez, Girish Kulkarni, and Tamara Berg. Im2text: Describing images using 1 million captioned photographs. In *NeurIPS*, 2011. 5, 8
- [52] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *NeurIPS*, 2019. 14
- [53] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *ICCV*, 2021. 15
- [54] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *AAAI*, 2018. 2
- [55] Julia Peyre, Ivan Laptev, Cordelia Schmid, and Josef Sivic. Detecting unseen visual relations using analogies. In *ICCV*, 2019. 5
- [56] Khoi Pham, Kushal Kaffle, Zhe Lin, Zhihong Ding, Scott Cohen, Quan Tran, and Abhinav Shrivastava. Learning to predict visual attributes in the wild. In *CVPR*, 2021. 2, 4, 5, 12, 13, 14
- [57] Bryan A. Plummer, Liwei Wang, Christopher M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. *IJCV*, 2017. 4
- [58] Karl Popper. *Conjectures and Refutations: The Growth of Scientific Knowledge*. Routledge, 1963. 1
- [59] Filip Radenović, Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondřej Chum. Revisiting oxford and paris: Large-scale image retrieval benchmarking. In *CVPR*, June 2018. 2
- [60] 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 *ICML*, 2021. 2, 5, 6, 7, 14, 15
- [61] Jerome Revaud, Jon Almazan, Rafael S. Rezende, and Cesar Roberto de Souza. Learning with average precision: Training image retrieval with a listwise loss. In *ICCV*, October 2019. 2
- [62] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 15
- [63] Karsten Roth, Timo Milbich, Samarth Sinha, Prateek Gupta, Bjorn Ommer, and Joseph Paul Cohen. Revisiting training strategies and generalization performance in deep metric learning. In *ICML*, 2020. 1, 2
- [64] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade W Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa R Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. Laion-5b: An open large-scale dataset for training next generation image-text models. In *NeurIPS Datasets and Benchmarks*, 2022. 2, 8
- [65] Sebastian Schuster, Ranjay Krishna, Angel Chang, Li Fei-Fei, and Christopher D Manning. Generating semantically

- precise scene graphs from textual descriptions for improved image retrieval. In *Proceedings of the fourth workshop on vision and language*, 2015. 5
- [66] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*, 2018. 2, 5, 6, 8, 15, 22
- [67] Yi Sun, Yuheng Chen, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation by joint identification-verification. In *NeurIPS*, 2014. 2
- [68] Yaniv Taigman, Ming Yang, Marc’ Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *CVPR*, June 2014. 1, 2
- [69] Reuben Tan, Mariya I. Vasileva, Kate Saenko, and Bryan A. Plummer. Learning similarity conditions without explicit supervision. In *ICCV*, 2019. 1, 2
- [70] Hugo Touvron, Alexandre Sablayrolles, Matthijs Douze, Matthieu Cord, and Hervé Jégou. Graft: Learning fine-grained image representations with coarse labels. In *ICCV*, 2021. 2
- [71] Mariya I Vasileva, Bryan A Plummer, Krishna Dusad, Shreya Rajpal, Ranjitha Kumar, and David Forsyth. Learning type-aware embeddings for fashion compatibility. In *ECCV*, 2018. 7
- [72] Vijay Vasudevan, Benjamin Caine, Raphael Gontijo-Lopes, Sara Fridovich-Keil, and Rebecca Roelofs. When does dough become a bagel? analyzing the remaining mistakes on imagenet. In *NeurIPS*, 2022. 13
- [73] Andreas Veit, Serge Belongie, and Theofanis Karaletsos. Conditional similarity networks. In *CVPR*, 2017. 1, 2, 3
- [74] Nam Vo, Lu Jiang, Chen Sun, Kevin Murphy, Li-Jia Li, Li Fei-Fei, and James Hays. Composing text and image for image retrieval-an empirical odyssey. In *CVPR*, 2019. 2, 7
- [75] Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu. Learning fine-grained image similarity with deep ranking. In *CVPR*, June 2014. 1
- [76] Hui Wu, Yupeng Gao, Xiaoxiao Guo, Ziad Al-Halah, Steven Rennie, Kristen Grauman, and Rogerio Feris. The fashion iq dataset: Retrieving images by combining side information and relative natural language feedback. *CVPR*, 2021. 2, 7
- [77] Hao Wu, Jiayuan Mao, Yufeng Zhang, Yuning Jiang, Lei Li, Weiwei Sun, and Wei-Ying Ma. Unified visual-semantic embeddings: Bridging vision and language with structured meaning representations. In *CVPR*, 2019. 5
- [78] Tete Xiao, Xiaolong Wang, Alexei A Efros, and Trevor Darrell. What should not be contrastive in contrastive learning. *ICLR*, 2021. 15
- [79] Yahui Xu, Yi Bin, Guoqing Wang, and Yang Yang. Hierarchical composition learning for composed query image retrieval. In *ACM Multimedia Asia*, 2021. 7
- [80] Yujie Zhong, Relja Arandjelović, and Andrew Zisserman. Faces in places: Compound query retrieval. In *BMVC*, 2016. 7
- [81] Xingyi Zhou, Rohit Girdhar, Armand Joulin, Philipp Krähenbühl, and Ishan Misra. Detecting twenty-thousand classes using image-level supervision. In *ECCV*, 2022. 6, 7, 14

# Appendix

## Table of Contents

<b>A Further GeneCIS Details</b>	<b>12</b>
A.1 Task construction . . . . .	12
A.2 Implementation Details . . . . .	13
A.3 Dataset noise . . . . .	13
A.4 Discussion on symmetry . . . . .	13
<b>B Specific Solutions</b>	<b>14</b>
<b>C Results with ViT-B/16 on GeneCIS</b>	<b>14</b>
<b>D Combiner Architecture</b>	<b>14</b>
<b>E Further Implementation Details</b>	<b>14</b>
<b>F. Extended Related Work</b>	<b>15</b>
<b>G Qualitative Examples</b>	<b>15</b>
G.1 GeneCIS examples . . . . .	15
G.2 Model Predictions . . . . .	15
G.3 Mined Triplets from CC3M . . . . .	15
<b>H Attributions</b>	<b>15</b>
<b>I. Acknowledgements</b>	<b>15</b>

We provide additional details and discussion of components of the main paper. We particularly highlight Appendix **A** for details on GeneCIS construction, and Appendix **G** for qualitative examples.

### A. Further GeneCIS Details

Here, we provide details on the construction process of each GeneCIS task, making reference to the examples from Figure 2 for clarity.

#### A.1. Task construction

**Focus on an Attribute:** VAW [56] contains bounding box annotations for various objects, as well as a list of *positively labelled attributes* and *negatively labelled attributes* for each object. Note that, as discussed in § 3.1, it is impossible to exhaustively label an object for all possible ‘positive’ attributes. It is, however, possible to determine a set of ‘negative’ attributes. For instance, one *cannot* exhaustively label a ‘thick’ tree trunk as {‘wide’, ‘fat’, ‘large’...etc.}, but one *can* determine that it is not ‘thin’.

For this task, we construct templates by first sampling a reference object (e.g ‘laptop’) and identifying all *positive*

*attributes* of the object (e.g. ‘white’, ‘plastic’) and their corresponding *attribute type* (e.g. ‘color’, ‘material’). Given an attribute type, we select a ‘correct’ target image to have the same object category and attribute within the attribute type as the reference (a ‘laptop’ with the same ‘color’). Distractors are then mined to have the same object category but to be explicitly *negatively labelled* for the reference attribute (e.g. laptops which are negatively labelled for ‘white’). The condition in this case is the attribute type (‘color’).

**Change an Attribute:** We first select an anchor *attribute type* (e.g ‘color’), before choosing a reference image and a ‘correct’ target image which share the same object category, but have different attributes within the attribute type. In Figure 2, the reference and ‘correct’ target are both have the same object category (‘train’) but have different ‘colors’. The attribute of the ‘correct’ target is given as the condition (‘olive green’), and a model must understand the category of the reference image, as well as the attribute specified in the condition, to solve the problem.

We include two forms of ‘distractors’ in the gallery. The first form includes images with the conditioning attribute (‘olive green’), but with a different object category (e.g. ‘tent’). These images behave as distractors for models which retrieve based only on the the condition (we include 9 such images). We also include 5 images with the reference object category but without the conditioning attribute (e.g ‘trains’ which are ‘red’), behaving as distractors for models which only use the reference image content.

**Focus on an Object:** For tasks where the condition contains an object, we take images of cluttered scenes from the multi-object COCO dataset [40]. We use COCO Panoptic Segmentation [36] data which contains dense category labels for every pixel in the image.

We first select a reference image and identify all of its constituent object categories, ensuring at least 10 categories are present. Next, we construct a set of all images in the dataset with at least 6 objects in common with the reference – but do not contain *all* reference categories – and rank them based on the extent of their category intersection ( $\mathcal{I}_{Close}$ ). We also construct a set of images with very *few* intersecting objects as  $\mathcal{I}_{Far}$ . We consider the set of object category IDs in an image as a ‘bag-of-words’ descriptor for the image scene, with images in  $\mathcal{I}_{Close}$  containing a ‘similar scene’ to the reference, and  $\mathcal{I}_{Far}$  representing a ‘different scene’.

We randomly select the ‘correct’ target image from  $\mathcal{I}_{Close}$ , and the *conditioning object* is selected as one of this image’s intersecting objects with the reference (e.g. ‘refrigerator’ in Figure 2). The first form of distractors is mined by taking images in  $\mathcal{I}_{Close}$  which *do not* have the conditioning object. These examples confuse models which only use the reference image (there are 9 of these). Another type of distractor is constructed by taking images from  $\mathcal{I}_{Far}$  which *do* have the conditioning object, confusing models which only

consider the text condition (there are 5 of these). In this way, only the target image has both a *similar scene* and also the *conditioning object*, and is thus *conditionally* the most similar image in the gallery. In Figure 2, only the target image contains a ‘refrigerator’ and is ‘outside’. We note that solutions which only match ‘bag-of-objects’ descriptors fail here (e.g. those which simply detect all objects in the images): the ‘correct’ gallery image is randomly selected from  $\mathcal{I}_{Close}$  and does not necessarily contain the highest object category overlap.

**Change an Object:** This task is constructed in a similar form to ‘Focus on an Object’, in that we first select a reference image and construct  $\mathcal{I}_{Close}$  and  $\mathcal{I}_{Far}$ . Differently, in this case, we first select the ‘correct’ gallery image from  $\mathcal{I}_{Close}$  as the most similar image which does not have perfect object overlap. Next, the conditioning object is selected randomly from the objects which *do* appear in the ‘correct’ gallery image, but *not* in the reference (‘ceiling’ in the example in Figure 2). Distractors are constructed from both  $\mathcal{I}_{Far}$  and  $\mathcal{I}_{Close}$ , such that they do, and do not, contain the conditioning object respectively. There are 5 distractors from  $\mathcal{I}_{Far}$  and 9 distractors from  $\mathcal{I}_{Close}$ .

## A.2. Implementation Details

**Attribute-based Tasks:** A taxonomy of *attribute types* is provided in VAW [56], containing diverse attribute types from ‘letter color’ to ‘texture’. We manually clean and refine the taxonomy for our purposes, for instance reassigning many attributes which were assigned to the ‘other’ attribute type. The resulting taxonomy contains 45 attribute types with 663 constituent attributes. We build the tasks such that they are roughly balanced with respect to attribute type, noting that for some attributes it was not possible to construct a suitable retrieval template. For the ‘Focus on an Attribute’ task, we manually filter attribute types which do not form clear and visually grounded attribute categories. Specifically, we filter: ‘*opinion*’; ‘*other after*’; ‘*other physical quality*’; ‘*state*’; and ‘*type*’.

Finally, when cropping an object with a bounding box, we dilate the box by a factor of 0.7 in height and width, before padding the resultant image to square with zeroes. This allows some context to identify the object (we often found it difficult to categorize the image without this), and also maintains the aspect ratio of the underlying object. We chose the dilation factor which maximized the discrepancy between the ‘Image + Text’ and ‘Image Only’ Recall@1.

**Object-based Tasks:** The object-based datasets are derived from the validation set of COCO Panoptic [36], containing 57K images with 133 categories. The categories include ‘thing’ classes like ‘zebra’ and ‘bench’, as well as ‘stuff’ categories like ‘sand’ and ‘roof’. We only consider an object category to be present in an image if it occupies more than 1% of the image pixels. After a conditioning object is

selected, the COCO category name is given as a condition, and we strip miscellaneous identifiers such as ‘-stuff’ and ‘-other’ from the category names.

## A.3. Dataset noise

Our tasks are built upon manual annotations in the VAW [56], COCO [40] and Visual Genome [37] datasets. These are widely used datasets in the vision community and, as such, our tasks should be error free in principle. However, we find some templates provide ill-posed problems through noise and ambiguities in the underlying annotations, as well as through bounding box dilation for attribute-based tasks.

Noise in the datasets is easy to understand, constituting instances where an object category or attribute is obviously mislabelled. However, the ambiguities are more subtle, and are artefacts of the underlying taxonomies of the datasets. For example, in some COCO images, ‘ceiling lights’ are labelled as ‘ceiling’ instead of ‘light’. This is not necessarily wrong, but reflects the fact that labels are defined in a one-hot manner and these pixels could refer to either object category. This is particularly difficult for attribute-based annotations, as interpretations of attributes are highly subjective (e.g. the definitions of ‘wide’ and ‘narrow’ are open to interpretation). We highlight that such label ambiguity, though underexplored, is present in almost all computer vision datasets, including in ImageNet [72].

We address this in a number of ways. Firstly, we ran a version of our method with 10 random seeds as well as on 10 cross-validation splits, finding the standard deviation in Recall@1 on each task to be around 0.2%. Although this does not quantify noise in the dataset, it gives an indication of what can be considered ‘signal’ on the tasks. Secondly, we find that the ‘Image + Text’ baseline outperforms the ‘Image Only’ and ‘Text Only’ baselines on most tasks, suggesting that the tasks measure conditional similarity. We discuss the exceptional case of ‘Focus on an Attribute’ in Appendix C. Thirdly, we evaluate at Recall@{1, 2, 3}, to account for any templates in which a ‘distractor’ image (*i.e.* ‘incorrect’ target) in the gallery actually constitutes a valid solution to the problem. Finally, we are in the process of manually filtering and verifying the templates, presenting the current version as ‘GeneCIS v0’.

## A.4. Discussion on symmetry

We highlight that ‘similarity’, as discussed in this paper, does not describe a *symmetric* mathematical property. In GeneCIS, while the reference image is considered ‘similar’ to the correct target image given the condition, the reverse may not be true. For instance, in the ‘Change an Attribute’ example in Figure 2, the ‘green train’ in the reference is conditionally similar to the ‘olive green train’ target image, given the condition ‘olive green’. However, this target image is *not* similar to the reference image given the same con-

dition. In general, we find that ‘Focus’ tasks *are* symmetric given the conditions, but ‘Change’ tasks *are not*.

## B. Specific Solutions

We design specific solutions for each of the proposed tasks in GeneCIS. These solutions take into account the specific construction mechanisms of each task and represent sensible approaches to tackling each task independently. We design all solutions to respect the ‘zero-shot’ nature of the evaluations and hence they are all based on ‘open-world’ models; we use CLIP [60] for the attribute-based tasks and Detic [81] for the object-based ones. All descriptions here refer to Figure 2 for clarity.

**Focus on an Attribute:** Given the attribute type in the condition (e.g. ‘color’), we first task CLIP with predicting the attribute of the reference image. Specifically, we use the taxonomy of attributes provided in VAW to construct a zero-shot classifier between attributes within that attribute type (e.g. {‘red’, ‘blue’, ‘white’} within ‘color’). Given the predicted attribute (e.g. ‘white’), we use its text embedding to find the nearest neighbour from the image embeddings of the gallery set.

**Change an Attribute:** We first use CLIP to predict the category of the the reference image, by constructing a zero-shot classifier from the categories in VAW. We then compute the text embedding of the concatenated predicted object name and conditioning attribute (e.g. ‘olive green train’) and find the nearest neighbour in the gallery.

**Focus on an Object and Change an Object:** We use the same specific solution for both of these settings. We first use Detic [81] to detect all object categories in the reference and gallery images, passing it the 2017 COCO Panoptic categories to construct the classifier. Next, we filter out any gallery images which *do not* contain the conditioning object category. Finally, using the detected object categories in a given image, we construct ‘bag-of-words’ descriptors of the reference image and the remaining gallery images. Specifically, these descriptors are binary vectors for each image (with elements for every COCO Panoptic category) and are set to ‘1’ if a given category is detected in the image. We use these descriptors to find the most conditionally similar image to the reference from the (filtered) gallery.

**Discussion:** Note that all of the solutions described here are specialized in two senses. Firstly, they are designed with the specific task construction method in mind, and hence are not applicable to all tasks as we desire for a general conditional similarity model. Secondly, all specific solutions leverage the underlying taxonomy of the *datasets* (VAW [56] and COCO Panoptic [36]) used in the benchmark.

## C. Results with ViT-B/16 on GeneCIS

In Table 2, we report results on GeneCIS with a ResNet50×4 backbone for fair comparison with [3]. However, in Figure 6, we demonstrate that our model performs best when intialized with a ViT-B/16 CLIP backbone [18, 60]. We include results for this model in Table 6, along with the CLIP-only baselines described in § 6.1.

We first note that, with the ViT-B/16 backbone, our model outperforms all CLIP-only baselines, on all tasks and at all recalls. Particularly, with the ResNet50×4 backbone in Table 2, the ‘Image Only’ baseline outperformed ours on ‘Focus Attribute’ at higher recalls, which is no longer the case here. We further note that the ‘Focus Attribute’ task gives anomalous results when comparing the ‘Image Only’ baseline with ‘Image + Text’. Specifically, this is the only task for which the ‘Image + Text’ model does not outperform the other baselines. On this task, the information given by the condition is an attribute *type*, rather than the attribute itself (e.g. ‘color’ rather than ‘white’ in Figure 2). As such, the condition information likely only confuses existing vision models, and reduces the performance over the ‘Image Only’ baseline.

## D. Combiner Architecture

The Combiner architecture takes in reference image and condition text features, composing them into a single vector as:  $g(\mathbf{x}^R, \mathbf{e})$  where  $g, \mathbf{x}^R, \mathbf{e} \in \mathbb{R}^D$ .

The architecture consists of four functions ( $h_i$ , built from MLPs) which process features in parallel as:

$$g(\mathbf{x}^R, \mathbf{e}) = \lambda h_1(\mathbf{x}^R) + (1 - \lambda)h_2(\mathbf{e}) + h_3(\mathbf{x}^R, \mathbf{e}) \quad (2)$$

where  $\lambda = h_4(\mathbf{x}^R, \mathbf{e})$  is a dynamic weighting of the features. We refer to [3] for full details.

## E. Further Implementation Details

**Our method:** All models trained on CC3M were trained for 28K gradient steps. We train our strongest models with an initial learning rate of  $1 \times 10^{-6}$  and a cosine decay schedule, training both the CLIP backbone and the Combiner head with the same learning rate. We evaluate our model at each epoch, selecting the checkpoint with the best Recall@1 on the CIRR validation set [44]. Note that this single model is then taken and evaluated *zero-shot* on all benchmarks reported in this paper. Our optimizer is Adam [35] and we implement our models using PyTorch [52]. To fine-tune both the backbones and Combiner head with a batch size of 256, we train our models on 16 A100 GPUs, with a training time of approximately 12 hours. Finally, all features are normalized before the contrastive loss is computed.

**Specific Solutions:** For the attribute-based specific solutions, we use the same ResNet50×4 backbone as for our

**Table 6. Evaluation on GeneCIS with a ViT-B/16 backbone** where we evaluate our method and CLIP-only baselines. We find our method performs best with a ViT-B/16 backbone and that, with this architecture, our model outperforms the baselines at all recalls on all GeneCIS tasks.

	Focus Attribute			Change Attribute			Focus Object			Change Object			Average R@1
	R@1	R@ 2	R@3	R@1	R@ 2	R@3	R@1	R@ 2	R@3	R@1	R@ 2	R@3	
Image Only	18.1	30.1	40.6	11.5	21.9	30.9	9.4	17.0	25.4	7.6	17.1	25.5	11.7
Text Only	10.3	20.9	30.4	10.2	18.2	26.1	7.4	14.0	23.0	8.1	16.4	24.7	9.0
Image + Text	17.1	29.5	40.5	13.1	22.2	31.9	11.5	20.1	29.2	9.8	20.0	28.9	12.9
Combiner (CC3M, Ours)	<b>19.7</b>	<b>31.7</b>	<b>42.1</b>	<b>16.2</b>	<b>27.3</b>	<b>37.5</b>	<b>16.6</b>	<b>27.7</b>	<b>37.2</b>	<b>18.0</b>	<b>32.2</b>	<b>41.6</b>	<b>17.6</b>

method in Table 2. Furthermore, when embedding an object name, we ensemble over the 80 standard CLIP prompts from [60]. For the object-based baselines, we use a Detic model with the strongest Swin-B backbone [43], trained for open-vocabulary detection on the ‘base’ classes of the LVIS dataset [22]. For the first detection stage, we select a confidence threshold which optimizes downstream Recall@1 on the ‘Focus Object’ evaluation (a confidence of 0.2).

## F. Extended Related Work

We briefly describe work from the *text-based image-editing* domain and discuss how it relates to our work. Generative image-editing [6, 49, 53] is a popular task in which, given a reference image and some condition, a sensible edit of the image is *generated*. Recently, with the advent of widely available large-scale generative models [62], there has been substantial work which considers the prompt as a text-condition, and uses CLIP text embeddings to guide the generation process [2, 5, 14, 20, 33, 38]. As such, the inputs and outputs of image-editing models are similar to those considered in this work. However, we highlight our work focuses on the *representation* and *retrieval* of existing images, rather than the synthesis of new ones.

We also note recent work which considers a task similar to ‘Change an Object’ in GeneCIS, in the context of compositional learning [48]. Similar to work detailed in § 2, [48] trains on a finite set of categories and considers only one of the four areas of the conditional similarity space which we propose here. Finally, we highlight [78], which considers contrastively training a single backbone with multiple heads, each of which is invariant to different data augmentations. This work shares similar motivations to ours – that there are many notions of image similarity – though they train with a pre-determined and fixed set of data augmentations (and hence fixed concepts of similarity).

## G. Qualitative Examples

### G.1. GeneCIS examples

We provide example templates from the GeneCIS benchmark tasks in Figures 7 to 10. Differently to Figure 2, we show the entire curated retrieval templates of 10-15 target images, as well as the reference image (leftmost, yellow) and condition text (blue oval). As discussed in § 4, all

gallery images are *implicitly similar* to the reference image or condition. The ‘positive’ target image is the *most similar given the condition*.

### G.2. Model Predictions

We show qualitative results of our model and CLIP-only baselines in Figure 11. We show instances where our model fails in Figure 12 along with the ‘correct’ target image and the prediction of the Image-Only CLIP baseline.

### G.3. Mined Triplets from CC3M

We show examples of training triplets which we *automatically* mine from CC3M [66] in Figure 13.

## H. Attributions

Figure 1: Thomas Hawk, CC BY-NC 2.0 (<https://creativecommons.org/licenses/by-nc/2.0/>), via Flickr.

Figure 4, image of horse on canvas: Simon Kozhin, CC BY-SA 3.0 (<https://creativecommons.org/licenses/by-sa/3.0/>), via Wikimedia Commons.

## I. Acknowledgements

We would like to thank Weidi Xie, Liliame Momeni, Mannat Singh and Kalyan Vasudev Alwala for valuable discussions on this work. Sagar is supported by a Facebook AI Research Scholarship.



Figure 7. Focus on an Attribute example templates.





Figure 8. Change an Attribute example templates.



Figure 9. Focus on an Object example templates.



Figure 10. Change an Object example templates.



Figure 11. Qualitative results from our method showing instances where our model predicts correctly.

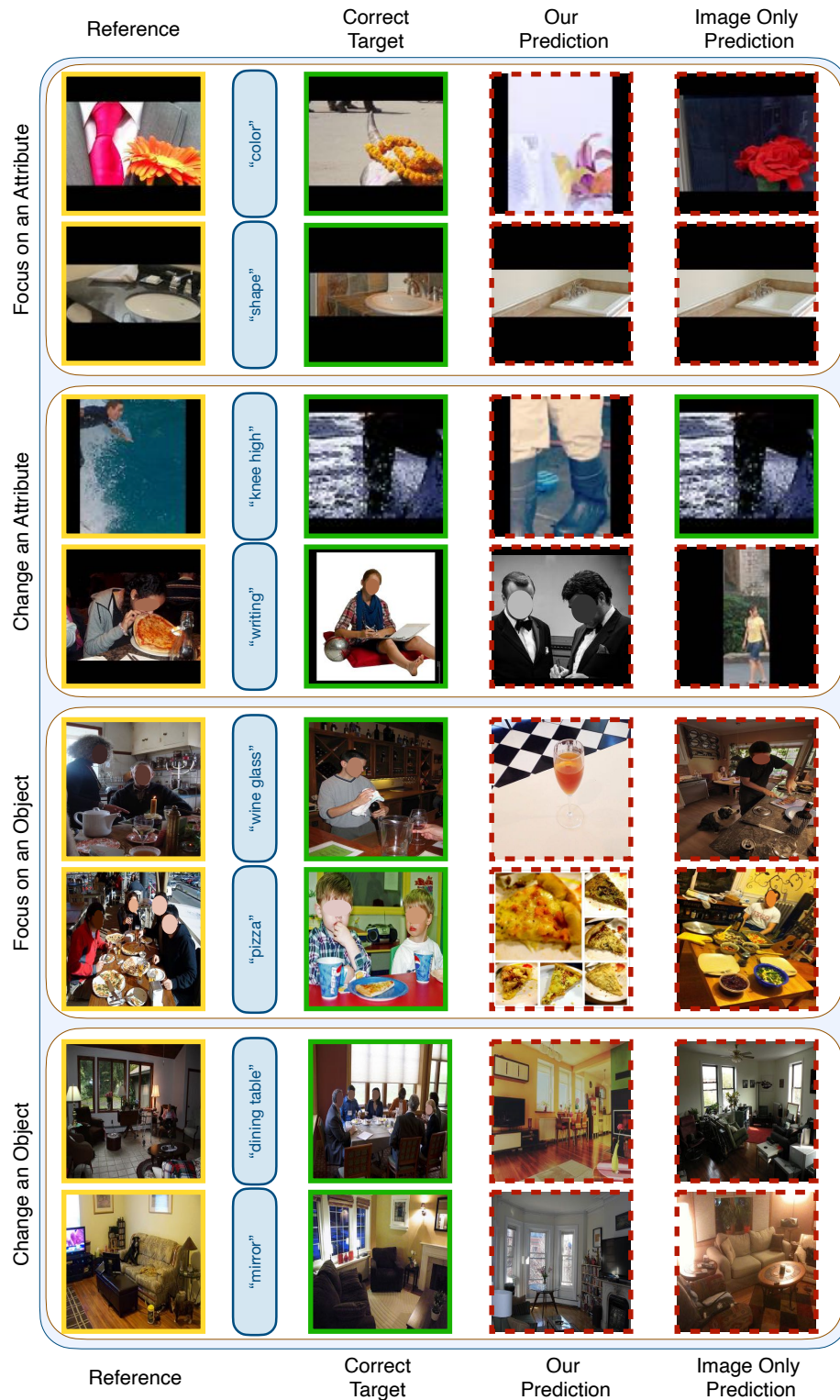
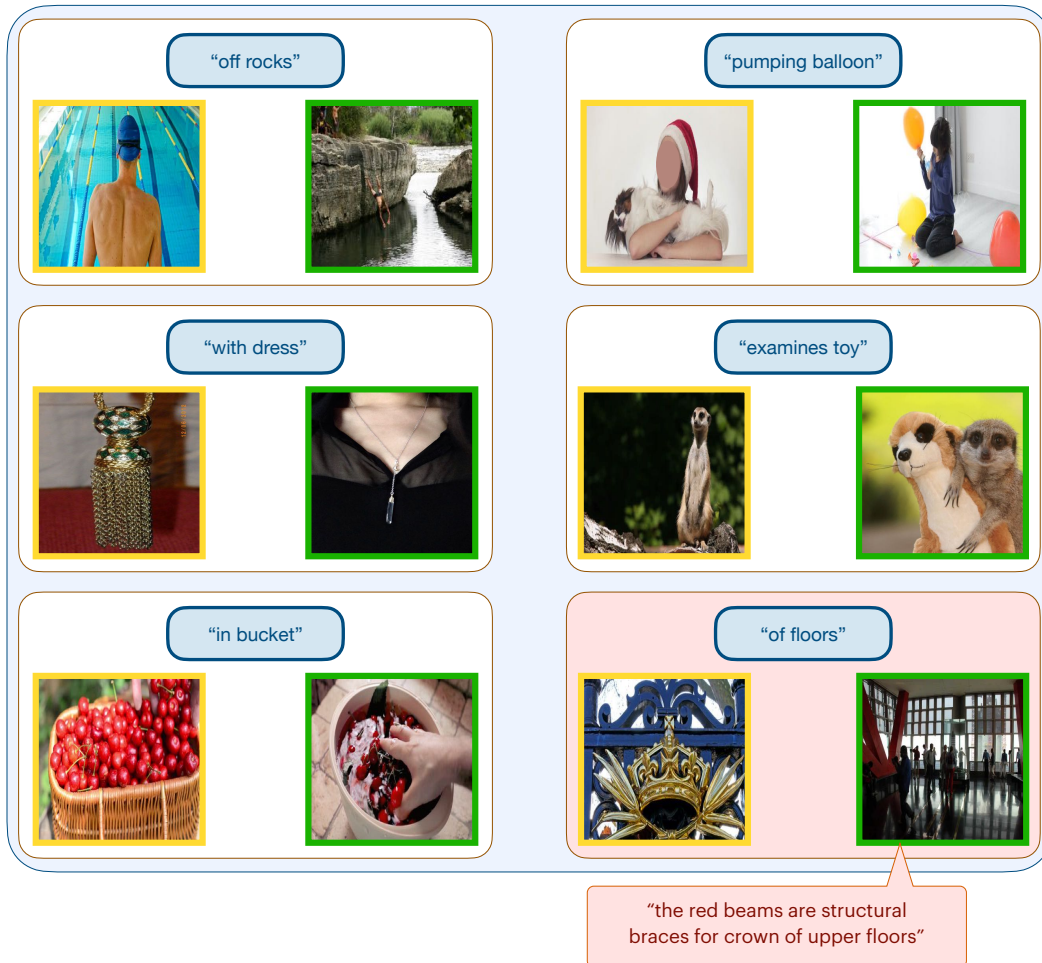


Figure 12. Qualitative results from our method showing instances where our model *fails*.



**Figure 13.** Examples of mined triplets from CC3M [66] as described in § 5.2. In each triplet, the text condition (blue oval) links the reference image (left) to the target image (right). We show an instance where a noisy triplet is produced in the bottom right. The caption (shown in a speech bubble) incurs a misleading parsed relationship of ‘Crown’ → ‘of’ → ‘floors’.