# Scene Designer: a Unified Model for Scene Search and Synthesis from Sketch

Leo Sampaio Ferraz Ribeiro[1], Tu Bui[2], John Collomosse[2, 3], and Moacir Ponti[1]

[1]ICMC, Universidade de São Paulo – São Carlos/SP, Brazil
{leo.sampaio.ferraz.ribeiro,ponti}@usp.br
[2]CVSSP, University of Surrey – Guildford, Surrey, UK
{t.bui,j.collomosse}@surrey.ac.uk
[3]Adobe Research, Creative Intelligence Lab – San Jose, CA, USA
collomos@adobe.com

## Abstract

*Scene Designer is a novel method for searching and generating images using free-hand sketches of scene compositions; i.e. drawings that describe both the appearance and relative positions of objects. Our core contribution is a single unified model to learn both a cross-modal search embedding for matching sketched compositions to images, and an object embedding for layout synthesis. We show that a graph neural network (GNN) followed by Transformer under our novel contrastive learning setting is required to allow learning correlations between object type, appearance and arrangement, driving a mask generation module that synthesizes coherent scene layouts, whilst also delivering state of the art sketch based visual search of scenes.*

## 1. Introduction

Creativity is increasingly inspired by the wealth of visual content online. Visual search eases content discovery and re-use, whilst generative artwork is emerging as a novel genre, driven by models trained on thousands of images. Yet the *fusion of search and synthesis* is under-explored. Generative content oveoffers users control but rarely the quality and diversity of real images. By contrast, search offers quality but not customization. Recent works have explored both image search and generation guided by free-hand sketches; an intuitive way to communicate visual intent. Sketch therefore offers an opportunity to unify search and synthesis technologies within a creative workflow.

This paper presents Scene Designer; a unified model for searching and generating scenes using free-hand sketches of scene compositions; i.e. drawings that describe both the appearance and relative positions of multiple objects. Our model provides a cross-modal search embedding where similarity of visual compositions comprising sketches and images (or mixtures of both) can be measured, as well as an object-level embedding for the synthesis of scene layouts to generate images. Our technical contributions are:

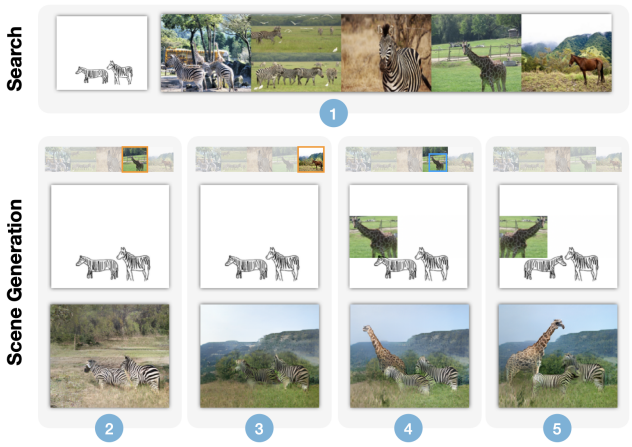



Figure 1. Scene Designer enables iterative design of image compositions through retrieval and synthesis. Row 1: User sketches initial concept, and matching scenes are returned (1). Row 2: In each column, the user uses the search results to help compose the final image; an orange square means the background was selected (2, 3) while the blue one means an object crop was chosen and added to the composition (4); on the final stage the user poses the objects as desired, and the final scene is synthesized (5).

**1. Compositional Sketch Search.** We propose a hybrid graph neural network (GNN) and Transformer architecture to learn a metric search embedding for comparing sketched and photographic scenes. Existing sketch based image retrieval (SBIR) methods predominantly match queries containing a single, dominant object invariant to its position within an image. Our novel contrastive training matches sketched compositions containing multiple objects.

**2. Sketch-to-Scene Synthesis.** We hallucinate entire scene layouts from either fully or partially sketched compositions, via a decoder that generates and combines object masks

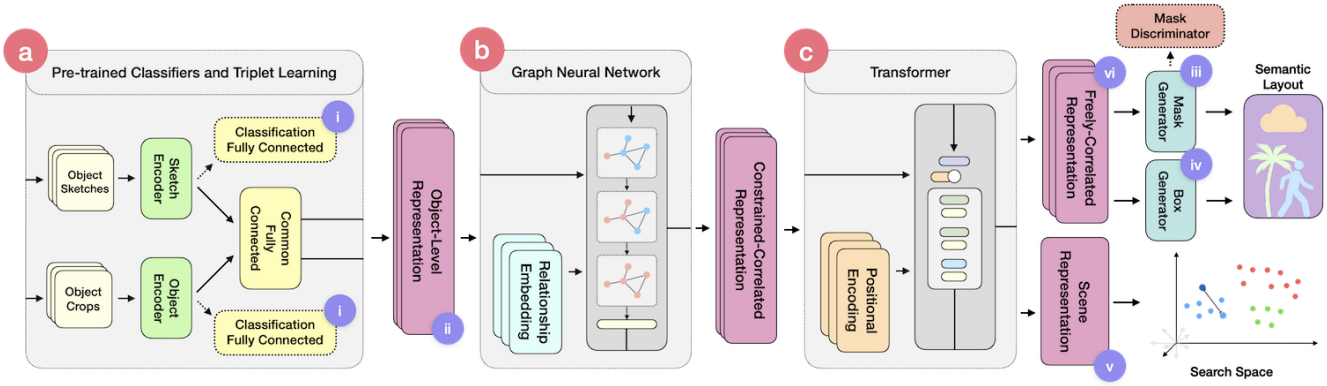



Figure 2. Proposed architecture of Scene Designer. Compositions (spatial arrangements of sketched or photoreal objects) are encoded via (a) to produce an *object-level representation*; *i.e*. a feature encoding for each object in the scene. A scene graph is constructed from these and encoded via GNN (b) to yield the *constrained-correlated representation*, and subsequently via Transformer to produce (c) the *freely-correlated representation* and *scene representation*. The roman numerals indicate where each of the losses (see Sec. 3.1, 3.4 and 3.5) are applied during training: (i) Cross-entropy classification $\mathcal{L}_{CCE}$, (ii) Triplet loss $\mathcal{L}_{tri}$, (iii) Mask generation losses $\mathcal{L}_{G_m}$, (iv) Bounding box generation losses $\mathcal{L}_{G_b}$, (v) Contrastive loss $\mathcal{L}_{cont}$, (vi) Final cross-entropy classification $\mathcal{L}_{CCE_f}$.

synthesized from learned correlations between object type, appearance and arrangement. The scene layouts drive texture synthesis to generate photographic content.

**3. Fusion of Synthesis and Cross-Domain Search.** Training a single mixed-task model yields improvements on the state of the art for both scene search and scene synthesis tasks. We also enable a novel creative workflow in which image compositions may be created iteratively; users sketch an initial composition (one or more objects), elements of the resulting images may be incorporated potentially with further sketched objects to drive either further searches or full image synthesis. Fig. 1 demonstrates how our fused model enables iterative design of generative compositions.

## 2. Related Work

**Sketch based image retrieval (SBIR)** has received extensive attention in the past decade. Dictionary learning methods (e.g. bag of visual words) have leveraged wavelet, edge-let, shape-let and sparse gradient features [1, 2] to match sketches to edge structures in images. With the advent of deep learning, CNNs (convnets) were rapidly adopted for cross-modal representation learning; exploring joint search embeddings for matching structure between sketches and images. Early approaches learned mappings between edge maps and sketches using contrastive [3], siamese [4] and triplet networks [5]. Fine-grained SBIR was explored by Yu *et al*. [6] and Sangkloy *et al*. [7] who used a three-branch CNN with triplet loss. Bui *et al*. learned a cross-domain embedding through triplet loss and partial weight sharing between sketch-image encoders [8]. Later studies followed a natural extension from single object sketches to scenes with multiple objects. Liu *et al*. [9] recently used a graph encoder with triplet loss to do compositional SBIR, but their results were dependent on known bounding boxes and category labels of photos and sketches. Our method also explores sketched compositions for search, learning a GNN and Transformer hybrid model for both search and synthesis.

**Conditional Image Generation** has been a rapidly developing field since the introduction of Conditional GANs [10]. Initial work exploring class label priors were soon followed by models with image based priors such as Pix2Pix [11] and more recently high definition synthesis models such as Pix2PixHD [12] and SPADE [13]. Those models learn from pairs of matching samples from each domain, and are capable of mapping semantic layouts to images.

**Scene Graphs** are a representation for images where individual objects are defined as nodes with the graph edges describe their relationships. They were first used for text-based image retrieval [14] but were more recently used by Johnson *et al*. [15] and Ashual *et al*. [16] as a initial representations to drive semantic layout generation and image synthesis based on those layouts. Our synthesis approach follows a similar principle, where our model learns to generate semantic layouts and a separate GAN (SPADE [13]) synthesizes the final image.

**Image-synthesis from Sketch** is a challenging problem due to the abstract and ambiguous nature of sketches. Recent work such as Sketchy-GAN [17] and Contextual-GAN [18] achieved promising results generating images from single-object sketches, but as was shown in [19], cannot accommodate sketched scenes with multiple objects. Gao *et al*. [19] were the first to implement image synthesis from sketch scenes and introduced the paired Sketchy-COCO dataset. Our approach differs, in that we learn a single mixed-task model for both search and synthesis — improving quantitatively and in user study synthesis results.

**Joint image search and synthesis** has been largely unexplored. Early work includes Sketch2Photo [20] that supports image search using text and sketch queries then compose an output image from objects of interest found in the

returned results. This approach uses statistical methods for image composition thus does not yield high quality output. In a more recent work, Sketchformer [21] also supports search and synthesis but only for sketch (single-domain). Similarly, Pang *et al*. [22] uses a sketch decoder mainly to assist their cross-domain learning thus have poor synthesized results. Our method learns both a cross-domain embedding for scene sketches and images and also enables image synthesis from composed objects in either domains.

## 3. Methodology

We developed a cross-modal representation learning framework in order to represent images and sketches with multiple objects in a common feature space; this representation is useful to both cross-modal retrieval and generation. Making use of scene graph representations, our framework looks at objects in a scene using a hierarchy based on object correlation. Fig. 2 describes the architecture and stages of the representation, which are each summarized below:

1. **Object-level representation (OLR):** Given an input composition, we encode individual objects (which might be sketched or photo-real) to a common representation. We refer to this embedding, in which each object is independently represented, as the OLR (subsec. 3.1).
2. **Scene Graph (SG)**: is formed using the OLR to encode objects, along with the discrete positional relationships of all object pairs.
3. **Constrained-correlated representation (CCR):** A graph neural network (GNN) encodes nodes in the Scene Graph (SG) into a continuous representation, encoding object appearances and their pairwise correlations (subsec. 3.2).
4. **Freely-correlated representation (FCR):** The sets of correlated vectors are fed into a Transformer module, where the attention layers allow for free correlation between all objects. The FCR encodes relationships between each object and all other objects in a weighted manner (subsec. 3.3). This representation is used for synthesis (subsec. 3.4).
5. **Scene representation (SR):** Together with the FCR, the Transformer module computes a separate single-vector latent space to which metric learning is applied to train a search embedding (subsec. 3.5).

We now describe in further detail how each stage of the representation is learned.

### 3.1. Object-level Representation (OLR)

We begin by independently encoding each object within the input composition to a common feature embedding, regardless of its input modality. Following contemporary single-object SBIR [8, 7, 6] we adopt a deep metric learning approach using a heterogeneous triplet architecture (*i.e*. no shared weights between the anchor (a) and positive/negative (p/n) branches). Our network comprises a MobileNet [23] backbone, terminated with two shared fully connected (fc) layers, the latter yielding the 128-D OLR embedding.

The MobileNet is pre-trained on ImageNet [24], and finetuned for cross-domain learning using a combined categorical cross-entropy (CCE) and triplet loss, with the latter defined as:

$$\begin{aligned}\mathcal{L}_{tri}\left(a,p,n\right) = \max\{0, m + |f_s(a), f_i(p)|_2 \\ - |f_s(a), f_i(n)|_2\}\end{aligned} \tag{1}$$

where $f_s(.)$ is the MobileNet that encodes sketches, $f_i(.)$ the one that encodes photo-real objects cropped from images, each followed by shared-domain MLP (2 fc layers) that encodes the MobileNet's outputs to the OLR, $m = 0.5$ is the margin and $|.|_2$ is the $L_2$ norm. Hereafter, we use shorthand $f(.)$ to refer to the encoder irrespective of its modality. The OLR is trained using rasterized sketches (a), and objects cropped from the COCO-stuff dataset with corresponding (p) and differing (n) object class; see subsec. 4.1 for dataset details. To aid convergence, the equal-weighted CCE loss $CCE(.)$ is applied to a further fc layer $f_e(.)$ appended to the network: vspace-0.4em

$$\mathcal{L}_{cce}(a,p,n) \quad = \quad \sum_{c\in\{a,p,n\}} CCE(f_e(c), \hat{c})) \tag{2}$$

where $\hat{a}$,$\hat{p}$, $\hat{n}$ are the one-hot vectors encoding the semantic class of each input.

### 3.2. Constrained-Correlated Representation (CCR)

A scene graph (SG) describes objects and their spatial relationships within the composition. Formally an SG is a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = \{o_1, \cdots, o_i, \cdots, o_\kappa\}$ is set of nodes representing the $\kappa$ objects, and $(o_i, r_{ij}, o_j) \in \mathcal{E}$ the set of edges encoding discrete relationships, $r_{ij} \in \mathcal{R}$ (we used "left of", "right of", "above", "below", "contains", "inside of").

We encode the scene graph via a graph neural network (GNN). We represent each node via the OLR as $f(o_i)$, encoding the pose and appearance of the object. As in [16], we adopt a learnable embedding $f_r(r_{ij})$ to map relationships $\mathcal{R}$ into dense vectors. The GNN $g_r(.)$ comprises a sequence of 6 layers $g_r^k(.)$. In each we follow the architecture of [15], and edges are processed by an fc layer $f_{fc1}^k$:

$$\begin{aligned}< v_i^0, r_{ij}^0, v_j^0 > = < f(o_i), f_r(r_{ij}), f(o_j) > \\ < \hat{v}_i^k, r_{ij}^{k+1}, \hat{v_j}^k > = f_{fc1}^k(< v_i^k, r_{ij}^k, v_j^k >)\end{aligned} \tag{3}$$

that updates the vector for each relationship to $r_{ij}^{k+1}$ and builds the set $\hat{v}_i^k \in V^k$ of object vectors. A function $(h)$ gathers vectors in $\mathcal{V}$ that describe the same object and averages them; finally they are processed by another fc layer to compute the updated $v_i^{k+1} = f_{fc2}^k(h(V^k))$ for each object. These layers are shared between images and sketches and trained end-to-end as part of our model to learn the *constrained-correlated representation* (CCR).

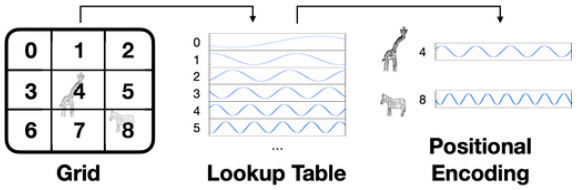



Figure 3. Grid-based Positional Encoding. We adapted the positional encoder from traditional Transformers by attributing one encoding to each block of the gridded scene.

### 3.3. Freely-Correlated Representation (FCR) and Scene Representation (SR)

We require a final representation that allows for objects to learn from their connection to all others in the scene, without being constrained to the pairwise encoding of the SG. We model this via attention layers, stacked similarly to a Transformer architecture [25]. This approach was inspired by the use of LSTMs for SG representation in [26] in order to accommodate a variable number of objects within a single representation. Transformers offer an attention mechanism to analyze each vector against all others, making them a good candidate for our model; They employ positional encoding so that the sequence order information is kept during processing. Our objects do not have a defined order, but we want their spacial positions to be taken into account. Following [16], we break the scene into a 5x5 grid and attribute a position $p \in 0, 1, ..., 24$ to each object based on where its center falls on the grid. This $p$ is used to compute its positional encoding (see Fig. 3). Note that Ashual *et al*. concatenated this $p$ to the object embeddings processed by the GNN; we took our approach because the vectors processed by our GNN are dynamic and because it aligns better with the Transformer architecture.

We specify our Transformer $t(.) = Z$ with 3 layers, and 16 attention heads on each. The input is the CCR $g_r(f(o_i))$ for each object in the image and one empty vector $\vec{0}$. In each layer $t^i(.)$ the model computes the attention weights that relate each vector to all the vectors in the sequence:

$$\begin{aligned} Z^0 &= s\left(F_q^0(Q+E)F_k^0(Q+E)^T\right)F_v^0(Q+E) \\ Z^{i+1} &= s\left(F_q^i(Z^i)F_k^i(Z^i)^T\right)F_v^i(Z^i), \end{aligned} \tag{4}$$

where multiplications are matrix-based, $s$ is the softmax function, $F_q^i$, $F_k^i$, $F_v^i$ are fc layers, $Q$ is a matrix made by stacking $\vec{0}$ and $g_r(f(o_i))$ CCR vectors and $E$ is our positional encodings $f_p(p)$, stacked in the same fashion as $Q$. In the output, the position that contained $\vec{0}$ has the SR while the other ones contain the FCR for each object.

### 3.4. Semantic Layout Generation

The learned FCR is used to synthesize a semantic scene layout (c.f. Fig. 7) that is used by our image generator. The layout is made from bounding boxes and masks for each object. We train two generators, one for masks and one for bounding boxes. The box generator is trained using Generalized Intersection-over-Union (GIOU) [27] and mean-squared error (MSE). The generator itself is a 2-layer MLP. The mask generator is trained within a GAN [28] framework, using a conditional setup [10] with LSGAN [29] losses plus MSE and Feature Matching to regularize the adversarial training. Additionally, another CCE classification loss $\mathcal{L}_{CCE_f}$ is applied to the FCR to encourage it to remain semantic.

Taking $b$ as ground truth bounding box, $\hat{x}_i$ as the FCR based on image input and $\hat{x}_s$ as the FCR based on sketch input, our generation losses are, for bounding boxes:

$$\mathcal{L}_{G_b}(x) = \lambda_1 \mathcal{L}_{GIOU}(b, G_b(\hat{x})) + \lambda_2 |b, G_b(\hat{x})|_2 \tag{5}$$

for each object on each image, applied to both $\hat{x}_i$ and $\hat{x}_s$, with $\lambda_1 = \lambda_2 = 10$. Using a similar notation, take our ground truth masks as $m_g$, and the object labels as $y \in C$, where $C$ is the set of object classes. The loss for the mask generator is:

$$\begin{aligned} \mathcal{L}_{G_m}(\hat{x}) = \lambda_3 |m_g, G_m(x)|_2 + \lambda_4 |D(G_m(\hat{x}), y), 1|_2 \\ + \lambda_5 \mathcal{L}_{FM}(\hat{D}(m_g, y), \hat{D}(G_m(x), y)) \end{aligned} \tag{6}$$

with $D$ as the discriminator and also applied to both $\hat{x}_i$ and $\hat{x}_s$. The weights are $\lambda_3 = 10$, $\lambda_4 = 0.25$, $\lambda_5 = 10$. The $\mathcal{L}_{FM}(.)$ is the feature matching loss, the L1 difference in the activations of $D$ ($\hat{D}(.)$). $D$ itself is trained with the adversarial loss in classic GAN fashion. Those losses backpropagate through all representation levels.

Finally, we turn the masks and boxes into a semantic layout. In this object-only layout, objects are placed ordered by their size, so that bigger objects are behind smaller ones (after [16]); then it may be combined with a layout for the background (*e.g.* selected from the top search results or coarsely drawn). Ultimately the final layout is passed through a SotA SPADE model [13] to synthesize an image.

### 3.5. Compositional Sketch-based Image Retrieval

For scene retrieval, we learn a single *Scene Representation* (SR) as a search embedding in which the similarity of images and the sketched input may be measured. This is an additional latent vector computed by the Transformer, that has metric properties encouraged through an adaptation of the contrastive loss.

During contrastive training, we sample half of our negatives randomly from other images of scenes in the training set. The other half of the negatives are synthesized by swapping objects of different class into the positions of the objects in the positive image (Fig. 4). By including such a class-swapped version of the image as a negative, the model is encouraged to discriminate change in object class to a greater degree than changes in object positions. Our contrastive loss, adapted to incorporate these synthetic negatives, is:

$$\mathcal{L}_{cont}(x_s, x_i, x_{sn}) = \frac{1}{2}(Y)(\mathcal{D}(x_s, x_i)) + \frac{1}{4}(1-Y)(1-\mathcal{D}(x_s, x_i)) - \frac{1}{4}\mathcal{D}(x_s, x_{sn}), \quad (7)$$

where $\mathcal{D}$ computes $|.|_2$ between all vectors on one set and all vectors on another set, $x_s$ and $x_i$ are respectively the sketch-based and image-based SR, $Y$ is a label indicating if a pair of vectors are from the same scene. Finally, $x_{sn}$ is the SR of the synthesized scenes with swapped objects.

### 3.6. Training Stages

Training is performed in three stages. First the OLR is trained independent of the rest of the model, using the dual triplet and categorical cross-entropy loss, for 100K iterations, on single-object sketches and object crops. In the second stage we use our novel soft-paired sketch and image scenes dataset (see Sec. 4.1) to train entire model end-to-end for a further 120K iterations. Finally, we finetune the model with a hard-paired dataset, training for a further 5K iterations. Training is via the ADAM [30] optimizer, with $\beta_1 = 0.5$ and $\epsilon = 1e-9$; learning rate (lr) is $lr = 10^{-4}$ for the first two stages and $lr = 10^{-5}$ for the finetuning stage. The mask discriminator follows the *Two Time-Scale Update Rule* (TTUR) [31] with $lr_D = 4lr$.

## 4. Experiments and Discussion

We evaluate the performance of Scene Designer for both compositional sketch search (SBIR) and synthesis, contrasting performance against contemporary baselines for both tasks. We explore the efficacy of our model for both search and synthesis from sketch, image and mixed domain compositions. For SBIR we compare against four baselines: a scene-level technique (SceneSketcher [9]) and three single-object techniques. Of these, two are fine-grained SBIR models (Sketchy [7], and Sketch-me-that-shoe [6]), and one is coarse-grained (Multi-stage Learning or MSL [8]). For scene synthesis we compare against the sketch driven method of Gao *et al*. [19] (proposed alongside their SketchyCOCO dataset) and the method of Ashual *et al*. [16] that accepts semantic scene graphs (spatial arrangements of keywords) as input.

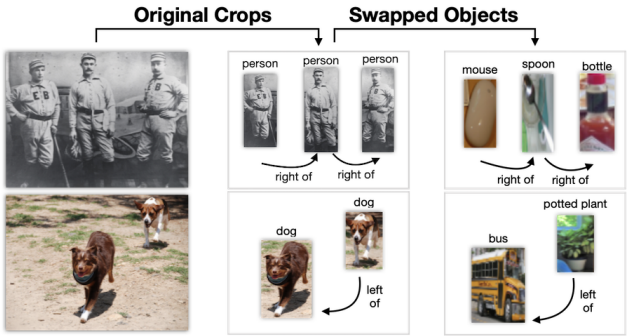



Figure 4. During contrastive training of the SR, negatives are synthesized by swapping objects in the scene for others of differing class. This helps the model prioritize class over structure. First column shows the original image, the second the individual cropped objects, and the third the swapped objects.

### 4.1. Datasets

We require paired data; sketched compositions and corresponding images of the scenes depicted by those sketches. **SketchyCOCO** is a recent dataset of 14K such pairs, proposed for sketch based scene synthesis [19]. SketchyCOCO samples scenes and associated annotations (bounding boxes, masks) from the COCO-stuff [32] images dataset, augmenting these with sketched depictions of those scenes. Approximately 4K of SketchyCOCO images contain object instances (the remainder are solely background material; 'stuff'). Only 14 classes (14c) of objects are annotated. To mitigate overfitting, we use the SketchyCOCO training partition (4,060 samples) only for fine-tuning our model. We include the test partition in our evaluation and split it per experiment requirements: retrieval eval. uses 233 samples as [9] needs scenes with 2 or more objects; generation tests needs to use the overlap between test sets used in each of the compared models: 137 samples. We also emulate [9] in constructing an 'Extended SketchyCOCO' search corpus by holding out 5K random images from the COCO-stuff training set and adding these to the SketchyCOCO test set as distractors.

**QuickDrawCOCO-92c** is a novel dataset we propose, composing scenes with sketches from QuickDraw, the largest public sketch dataset (50M) [33]. Inspired by SketchyCOCO, we similarly leverage scenes from COCO-stuff, taking advantage of the class overlap (92c) with QuickDraw. We synthesize sketch scenes by compositing sketches from QuickDraw onto a canvas, selecting a random sketch within the class corresponding to each object in a COCO-stuff image (Fig. 5). Our dataset is soft-paired, in that the sketches for each object do not necessarily match pose and appearance with the image, only the class label. We use QuickDrawCOCO-92c for all training stages except for the final finetuning stage where we combine samples from both SketchyCOCO and QuickDrawCOCO-92c. We separately fuse the training and test partitions of the QuickDraw and COCO-stuff datasets to partition QuickDrawCOCO-92c into 111,112 training, 2,788 test and 1,907 validation scenes. This dataset is 6x more category-diverse and has 8x more samples than SketchyCOCO.

### 4.2. Evaluating Retrieval

We adopt the evaluation proposed in SceneSketcher [9], the only prior technique exploring SBIR for scenes; retrieved images are relevant only if they match the sketched scene exactly. Models are evaluated on SketchyCOCO, QuickDrawCOCO-92c and 'Extended SketchyCOCO'.

**Compositional search.** Tab. 1 reports recall at rank $k =$

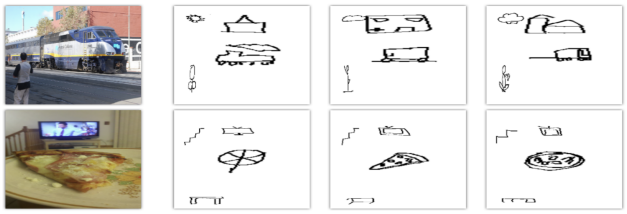


Figure 5. Sample paired data from the proposed QuickDrawCOCO-92c dataset. Each row shows one COCO-stuff [32] image and three examples of synthetic sketched compositions matching that image derived from QuickDraw [33].

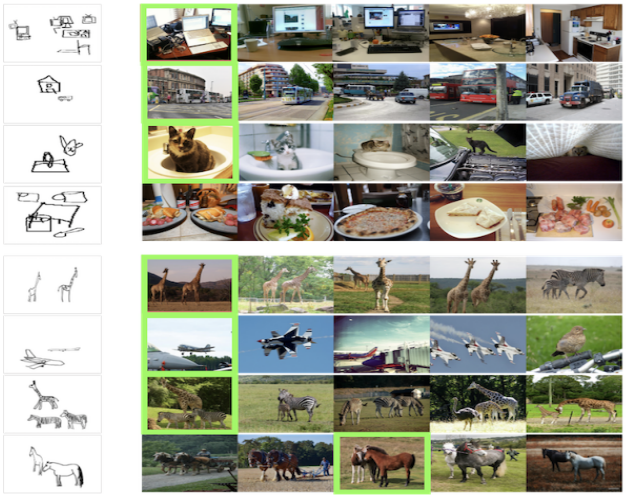


Figure 6. Retrieval results using the QuickDrawCOCO (first block) and SketchyCOCO sets (second). The results often contain the exact match and are contextually consistent, such as in the last row of QuickDrawCOCO, where the context of "table-spoon-mug-etc" retrieves sets of "meals". Relevant results in green; final row of each block represents a failure case.

$\{1, 5, 10\}$ (R@k) for our proposed search embedding for sketched scenes, that of SceneSketcher [9], and three single-object SBIR baselines: Sketch-me-that-shoe [6]; Sketchy [7]; and MSL [8]. We report R@k over SketchyCOCO and Extended SketchyCOCO for all methods; figures for SceneSketcher and Sketch-me-that-shoe are taken from Liu *et al*. [9]. We are unable to report values for these two models over our QuickDrawCOCO-92c test data, due to the lack of public models/code for these methods. QuickDrawCOCO-92c is a more challenging query set due to its increased size and the more messy / abstract sketches within QuickDraw. We present our results alongside public Sketchy [7] and MSL [8] models (final column, Tab. 1).

Our model more than doubles previous SotA recall results and greatly improves other metrics. The penultimate row of Tab. 1 reports performance our method using mixed-domain queries, where we substitute half of the sketched objects for their image cropped counterparts, which yields an even higher result likely due to half of the objects sharing the same domain as the test corpus. The final row reports on our model trained on QuickDrawCOCO-92c only; prior

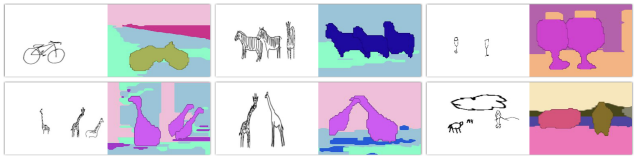


Figure 7. Example layouts synthesized by Scene Designer before they are transformed to images by the SPADE [13] generator. First 2 columns show SketchyCOCO scenes and third column shows QuickDrawCOCO-92c scenes.

to finetuning on SketchyCOCO training set. This demonstrates that our model generalizes sufficiently to beat the SotA on SketchyCOCO without explicit training on it. Visual results for both datasets are in Fig. 6.

**Single object search.** We evaluate the performance of our model at single-object retrieval, baselining against both the Sketchy and MSL models on the SketchyCOCO dataset. We sample 120 queries randomly from SketchyCOCO containing only one object. In contrast to composition search, many relevant results exist in the SketchyCOCO dataset for a given sketched object. We consider a result relevant if both its category and pose match the sketched query. Since no ground-truth annotation with these criteria is available for SketchyCOCO, we crowd-source per-query annotation via Mechanical Turk (MTurk) for the top-$k$ ($k$=15) results and compute both recall@$k$ and precision@$k$ in Tab. 2. For each ranked result we ask 5 MTurk participants to indicate relevance (or not), and filter on consensus level. Our approach significantly outperforms both baselines at all consensus levels. The sensitivity to object orientation and pose underlines the utility of sketch over labeled boxes (per [16]) to specify the appearance of the desired object.

### 4.3. Evaluating Scene Synthesis

We evaluate the performance of our proposed model at image synthesis (see Fig.8) using SketchyCOCO, comparing against that dataset's accompanying public model [19], and against the public model of Ashual *et al*. [16] which synthesizes images from spatial word maps *i.e*. implicit scene graphs. We synthesize from the sketch, and use the paired image as ground truth for evaluating the fidelity of the output. For Ashual *et al*., we create an input scene graph based on object classes and bounding boxes.

**Objective Metrics.** We compute Fréchet Inception Distance (FID) [31] and Object Classification Accuracy [16] to compare the synthesized and ground truth images. A lower FID value indicates that the tested image distribution is closer the original and is more realistic. The Accuracy values come from finetuning a ResNet-101 classifier on crops from COCO and applying it to foreground objects cropped from the generated images.

Tab. 4 shows our model to score higher than both baselines on both FID and Accuracy. We have included results applying SPADE to the ground-truth layout, which serves as an upper bound generation result for our method. Results from mixed-domain and images-only compositions,

| Method | SketchyCOCO | | | Ext. SCOCO | | | QuickDrawCOCO | | |
|---|---|---|---|---|---|---|---|---|---|
| | r@1 | r@5 | r@10 | r@1 | r@5 | r@10 | r@1 | r@5 | r@10 |
| Ours | **75.53** | **96.56** | **99.14** | **66.09** | **90.55** | **96.13** | **62.15** | **78.90** | **85.07** |
| Sketch-me-that-shoe [6] | 06.19 | 17.15 | 32.86 | <0.01 | <0.01 | 01.90 | - | - | - |
| Sketchy [7] | 03.43 | 09.87 | 15.87 | <0.01 | 00.85 | 00.85 | 00.07 | 00.35 | 00.60 |
| MSL [8] | 09.44 | 28.75 | 43.34 | 05.15 | 15.87 | 17.59 | 00.10 | 00.53 | 01.07 |
| SceneSketcher [9] | 31.91 | 66.67 | 86.19 | 12.38 | 26.67 | 38.10 | - | - | - |
| Ours (Mixed-domain) | 89.69 | 99.57 | 100.0 | 87.12 | 96.56 | 98.71 | 93.22 | 97.37 | 98.23 |
| Ours (Before finetuning) | 45.49 | 80.25 | 93.56 | 15.45 | 41.20 | 50.21 | 64.27 | 81.95 | 87.23 |

Table 1. Recall@K metrics on the SketchyCOCO dataset, its extended version and the QuickDrawCOCO-92c set. Our model more than doubles the previous state-of-the-art recall@1 metric. Additionally, we show our performance when using mixed-domain compositions and also with the model trained only with QuickDrawCOCO-92c, before finetuning on SketchyCOCO.

| Method | Precision@1 | | | Precision@15 | | |
|---|---|---|---|---|---|---|
| | C2 | C3 | C4 | C2 | C3 | C4 |
| Ours | **66.12** | **52.07** | **37.19** | **22.98** | **16.14** | **10.30** |
| MSL [8] | 16.53 | 14.05 | 09.09 | 10.41 | 07.00 | 04.41 |
| Sketchy [7] | 35.54 | 23.97 | 18.18 | 17.36 | 11.79 | 07.77 |
| | Recall@1 | | | Recall@15 | | |
| | C2 | C3 | C4 | C2 | C3 | C4 |
| Ours | **66.12** | **52.07** | **37.19** | **96.69** | **90.91** | **71.90** |
| MSL [8] | 16.53 | 14.05 | 09.09 | 84.30 | 72.73 | 56.20 |
| Sketchy [7] | 35.54 | 23.97 | 18.18 | 85.95 | 80.17 | 65.29 |

Table 2. MTurk sketch search results for the proposed and baseline methods on the SketchyCOCO dataset. Precision@k and Recall@k are presented at different (C)oncensus thresholds.

which are a unique capability of our model, are also included. These show the value in applying Scene Designer to incrementally build up a composition as part of an interactive creative process.

**User Perceptual Study.** We assess the quality of our synthesized images following the subjective evaluation methodology proposed by Gao *et al*. [19]. Specifically we score how 'realistic' each image is, and how 'faithful' each synthesized image is in representing the spatial object arrangement of the input scene (in the case of [16], spatial arrangement of labeled boxes). User opinion on 'faithfulness' is scored on a scale 1 ('very dissatisfied') to 4 ('very satisfied') using crowd-sourced annotations collected via MTurk. For realism, images synthesized by each method are presented to participants, inviting selection of the most realistic. In both cases each task is annotated by 4 unique participants, and results tabulated for differing levels of participant consensus. We consider only results where 2 or more MTurkers agree *i.e*. there was consensus. Tab. 3 indicates our method is preferred for content faithfulness and realism, although faithfulness scores are low in general (circa 2.5 on the scale) this is in line with reported figures for other methods e.g. 1.57 for [16] as shown in [19].

### 4.4. Ablation studies

We explore the significance of each stage of our proposed architecture and training methodology, comparing SBIR and generation results on QuickDrawCOCO-92c using several ablated variants as shown in Tab. 5, alongside the QuickDrawCOCO-92c result for the full method. We explore three categories of ablation (A-C). Category A ablates key features of the proposed architecture. Category B ablates key training steps. Category C investigates if training a single model for both search and synthesis does not degrade the model's performance at either.

**(A1) Without Transformer.** Removes the attention modules from the model, aggregating the representations from the GNN using simple addition.

**(A2) Without GNN.** Substitutes the GNN for an fc layer that bridges the OLR into the Transformer module.

**(A3) Without Positional Encoding.** Retains the proposed architecture but removes our grid-based positional encoding from the Transformer module.

**(B1) Without object-level pretraining.** Skips the first stage of training the OLR.

**(B2) Without shuffled-objects negatives.** Does not use our synthesized negative samples in the contrastive loss.

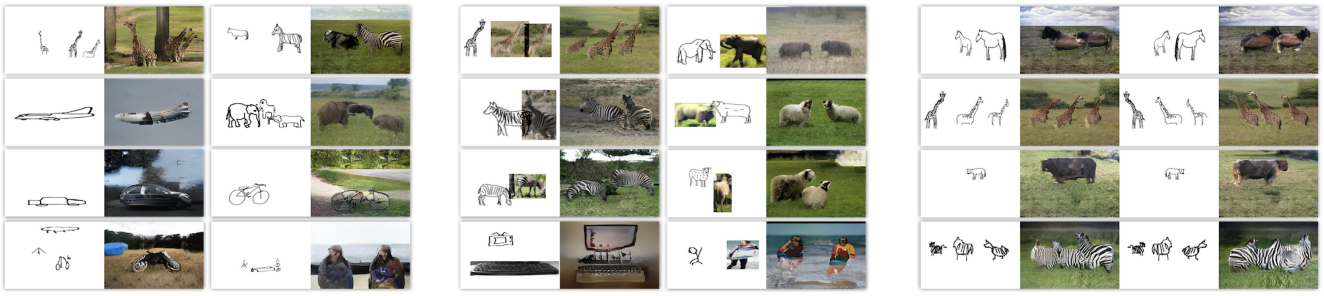
**(C1) No Contrastive Loss.** Does not use contrastive loss. The results on this model are driven by the masks and boxes generation losses and object-level triplet loss only.

**(C2) No Generation Losses.** By removing the masks and boxes generation losses, this ablated model is only trained with search associated losses.

Removing either the GNN (A2) or the grid-based positional encoding (A3) degrades both tasks' performance significantly while the Transformer (A1) is shown to be helpful for generation (improves FID by 25%), and crucial for search.

Our synthesized negative examples (B2) improve search by 8% and are essential for generation to work. This follows the result of (C1) where removing the Contrastive Loss also diminishes generation performance; given that both of those are associated with search but also impact generation show that the strength of our multi-task model lies on the synergy between those tasks.

When we then look at (C2) that hypothesis is further proven, as removing the synthesis-related losses impact the search results as well with a drop of 6% in recall@1. Training for both tasks does not decrease performance, but boosts it meaningfully. We conclude that learning general representations requires the model to retain both visual and semantic information, which we achieve by mixing generation and feature learning losses.



(a) Sketch composition (b) Mixed domain composition (c) Posing of individual objects

Figure 8. Generating images with Scene Designer. The first 3 rows show compositions and sketches from SketchyCOCO whilst the last one shows those from QuickDrawCOCO-92c. There are three features shown: (a) sketch composition. (b) compositions that mix image and sketched objects. (c) individual object posing, where we flipped each object individually and compare the generations.

| Method | SketchyCOCO | | | | | | QuickDrawCOCO | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Realism | | | Faithfulness | | | Realism | | | Faithfulness | | |
| | C2 | C3 | C4 | C2 | C3 | C4 | C2 | C3 | C4 | C2 | C3 | C4 |
| Ours | **64.17** | **72.73** | **82.61** | 2.67 | 2.50 | **2.16** | **62.77** | **63.64** | **75.68** | **2.66** | 2.56 | 2.58 |
| Ashual *et al*. [16] | 13.33 | 13.64 | 04.35 | 2.56 | 2.34 | 1.90 | 37.23 | 36.36 | 24.32 | 2.61 | **2.58** | **2.88** |
| SketchyCOCO [19] | 22.50 | 13.64 | 13.04 | **2.79** | **2.65** | 2.04 | - | - | - | - | - | - |

Table 3. User perceptual study (via MTurk) evaluating the generated images. Realness is a comparative score between the models while faithfulness is individually scored per method on continuous scale [1,4]. Results are thresholded for different levels of participant (C)onsensus, from 2 to 4 (out of 4) participant agreements.

| Method | SketchyCOCO | | QDCOCO-92c | |
|---|---|---|---|---|
| | FID↓ | Acc.↑ | FID↓ | Acc.↑ |
| Ours (sketch-based) | **130.87** | **63.46** | **76.64** | **65.47** |
| Ashual *et al*. [16] | 170.40 | 56.20 | 103.14 | 50.69 |
| SketchyCOCO [19] | 198.17 | 29.14 | - | - |
| Ours (image-based) | 138.82 | 57.36 | 69.08 | 59.52 |
| Ours (mixed-domain) | 143.24 | 65.74 | 83.13 | 58.53 |
| SPADE [13] | 111.25 | 81.21 | 58.39 | 80.95 |

Table 4. Generation metrics when using samples from SketchyCOCO and QuickDrawCOCO-92c. Both quantitative metrics were computed on the overlapping test set across the three/two models for each dataset (137 samples on the former and 1300 on the later). A lower FID and higher Object Classification Accuracy represent a better result. We've also included the metrics for the SPADE generator, representing an upper bound for our model.

| Model Settings | r@1 | r@10 | FID↓ |
|---|---|---|---|
| (A1) W/o Transformer | 07.20 | 20.62 | 102.79 |
| (A2) W/o GNN | 02.69 | 17.61 | 118.04 |
| (A3) W/o positional encoding | 14.45 | 45.87 | 122.01 |
| (B1) W/o obj.-level pretraining | 55.84 | 85.43 | 102.14 |
| (B2) W/o synthesized negatives | 54.12 | 81.16 | 128.58 |
| (C1) No contrastive loss | 45.48 | 74.67 | 129.88 |
| (C2) No generation losses | 55.84 | 81.34 | 129.98 |
| Final model | **62.15** | **85.07** | **76.64** |

Table 5. Ablation Studies, showing Recall@k and FID on QuickDrawCOCO. With each set of models, we want to show that (A) each component is necessary, (B) the training procedure aids performance and (C) the single model can multi-task well.

## 5. Conclusion

We introduced Scene Designer; a single unified model for searching and generating images using free-hand sketches of scenes. We developed a mixed-task learning framework with three levels of representation (OLR, CCR, FCR) using a hybrid GNN-Transformer architecture. Scene Designer learns to embed sketched and photographic scenes into a common space, producing latent representations for both synthesizing layouts from sketched scenes and for measuring compositional similarity between sketch and image scenes. The model is trained via an expanded dataset of sketch compositions and corresponding images (QuickDrawCOCO-92c); a secondary contribution of our work. We show that the combination of feature learning and generation losses aided by our novel take on contrastive learning for scenes are responsible for obtaining SotA performance at scene search and synthesis tasks. The ability to sketch an initial composition, and incorporate components of results into hybrid queries for synthesis (or further searches) creates a novel mechanism for interactively constructing scenes from digital image collections. Further work could explore this interactive model in creative practice, and perhaps explore how other facets of generative artwork (*e.g*. neural style transfer) might be incorporated into the framework for example to enable fine-grained control over the appearance of objects within the composition.

## Acknowledgments

This work was supported by FAPESP (grants 2017/22366-8, 2019/02808-1, 2019/07316-0), CNPq Fellowship (304266/2020-5), and a charitable donation from Adobe Inc.

## References


[1] Mathias Eitz, Kristian Hildebrand, Tamy Boubekeur, and Marc Alexa. Sketch-based image retrieval: Benchmark and bag-of-features descriptors. *IEEE Trans. Visualization and Computer Graphics*, 17(11):1624–1636, 2011. 2

[2] Rui Hu and John Collomosse. A performance evaluation of gradient field hog descriptor for sketch based image retrieval. *Computer Vision and Image Understanding*, 117(7):790–806, 2013. 2

[3] Fang Wang, Le Kang, and Yi Li. Sketch-based 3d shape retrieval using convolutional neural networks. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 1875–1883. IEEE, 2015. 2

[4] Yonggang Qi, Yi-Zhe Song, Honggang Zhang, and Jun Liu. Sketch-based image retrieval via siamese convolutional neural network. In *Image Processing (ICIP), 2016 IEEE International Conference on*, pages 2460–2464. IEEE, 2016. 2

[5] Tu Bui, Leo Ribeiro, Moacir Ponti, and John Collomosse. Compact descriptors for sketch-based image retrieval using a triplet loss convolutional neural network. *Computer Vision and Image Understanding*, 2017. 2

[6] Qian Yu, Feng Liu, Yi-Zhe Song, Tao Xiang, Timothy M Hospedales, and Chen-Change Loy. Sketch me that shoe. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 799–807, 2016. 2, 3, 5, 6, 7

[7] Patsorn Sangkloy, Nathan Burnell, Cusuh Ham, and James Hays. The sketchy database: learning to retrieve badly drawn bunnies. *ACM Transactions on Graphics (TOG)*, 35(4):119, 2016. 2, 3, 5, 6, 7

[8] Tu Bui, Leo Ribeiro, Moacir Ponti, and John Collomosse. Sketching out the details: Sketch-based image retrieval using convolutional neural networks with multi-stage regression. *Computers & Graphics*, 71:77–87, 2018. 2, 3, 5, 6, 7

[9] Fang Liu, Changqing Zou, Xiaoming Deng, Ran Zuo, Yu-Kun Lai, Cuixia Ma, Yong-Jin Liu, and Hongan Wang. Scenesketcher: Fine-grained image retrieval with scene sketches. 2020. 2, 5, 6, 7

[10] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 2, 4

[11] P. Isola, J. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5967–5976, 2017. 2

[12] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2

[13] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 2, 4, 6, 8

[14] Justin Johnson, Ranjay Krishna, Michael Stark, Li Jia Li, David A. Shamma, Michael S. Bernstein, and Fei Fei Li. Image retrieval using scene graphs. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 07-12-June, pages 3668–3678. IEEE, jun 2015. 2

[15] Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image Generation from Scene Graphs. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1219–1228. IEEE, jun 2018. 2, 3

[16] Oron Ashual and Lior Wolf. Specifying object attributes and relations in interactive scene generation. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2019-Octob, pages 4560–4568, sep 2019. 2, 3, 4, 5, 6, 7, 8

[17] Wengling Chen and James Hays. Sketchygan: Towards diverse and realistic sketch to image synthesis. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[18] Yongyi Lu, Shangzhe Wu, Yu-Wing Tai, and Chi-Keung Tang. Image generation from sketch constraint using contextual gan. In *The European Conference on Computer Vision (ECCV)*, September 2018. 2

[19] Chengying Gao, Qi Liu, Qi Xu, Limin Wang, Jianzhuang Liu, and Changqing Zou. Sketchycoco: Image generation from freehand scene sketches. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2, 5, 6, 7, 8

[20] Tao Chen, Ming-Ming Cheng, Ping Tan, Ariel Shamir, and Shi-Min Hu. Sketch2photo: Internet image montage. *Proc. ACM SIGGRAPH*, 28(5):124, 2009. 2

[21] Leo Sampaio Ferraz Ribeiro, Tu Bui, John Collomosse, and Moacir Ponti. Sketchformer: Transformer-based representation for sketched structure. In *Proc. CVPR*, 2020. 3

[22] Kaiyue Pang, Yi-Zhe Song, Tony Xiang, and Timothy M Hospedales. Cross-domain generative learning for fine-grained sketch-based image retrieval. In *BMVC*, pages 1–12, 2017. 3

[23] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 3

[24] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 3

[25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017. 4

[26] Longteng Guo, Jing Liu, Jinhui Tang, Jiangwei Li, Wei Luo, and Hanqing Lu. Aligning linguistic words and visual semantic units for image captioning. In Laurent Amsaleg, Benoit Huet, Martha A. Larson, Guillaume Gravier, Hayley Hung, Chong-Wah Ngo, and Wei Tsang Ooi, editors,

*Proceedings of the 27th ACM International Conference on Multimedia, MM 2019, Nice, France, October 21-25, 2019*, pages 765–773. ACM, 2019. 4

[27] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 4

[28] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27*, 2014. 4

[29] Xudong Mao, Qing Li, Haoran Xie, Raymond Y. K. Lau, and Zhen Wang. Least squares generative adversarial networks, 2016. cite arxiv:1611.04076. 4

[30] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations (ICLR)*, 2015. 5

[31] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6629–6640, Red Hook, NY, USA, 2017. Curran Associates Inc. 5, 6

[32] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Cocostuff: Thing and stuff classes in context. In *Computer vision and pattern recognition (CVPR), 2018 IEEE conference on*. IEEE, 2018. 5, 6

[33] The Quick, Draw! Dataset. https://github.com/googlecreativelab/quickdraw-dataset, 2018. Accessed: 2018-10-11. 5, 6