

# FS-COCO: Towards Understanding of Freehand Sketches of Common Objects in Context

Pinaki Nath Chowdhury<sup>1, 2</sup> Aneeshan Sain<sup>1, 2</sup> Ayan Kumar Bhunia<sup>1</sup>  
Tao Xiang<sup>1, 2</sup> Yulia Gryaditskaya<sup>1, 3</sup> Yi-Zhe Song<sup>1, 2</sup>

<sup>1</sup>SketchX, CVSSP, University of Surrey, United Kingdom.

<sup>2</sup>iFlyTek-Surrey Joint Research Centre on Artificial Intelligence.

<sup>3</sup>Surrey Institute for People Centred AI, CVSSP, University of Surrey.

**Abstract.** We advance sketch research to scenes with the first dataset of freehand scene sketches, FS-COCO. With practical applications in mind, we collect sketches that convey scene content well but can be sketched within a few minutes by a person with any sketching skills. Our dataset comprises 10,000 freehand scene vector sketches with per point space-time information by 100 non-expert individuals, offering both object- and scene-level abstraction. Each sketch is augmented with its text description. Using our dataset, we study for the first time the problem of fine-grained image retrieval from freehand scene sketches and sketch captions. We draw insights on: (i) Scene salience encoded in sketches using the strokes temporal order; (ii) Performance comparison of image retrieval from a scene sketch and an image caption; (iii) Complementarity of information in sketches and image captions, as well as the potential benefit of combining the two modalities. In addition, we extend a popular vector sketch LSTM-based encoder to handle sketches with larger complexity than was supported by previous work. Namely, we propose a hierarchical sketch decoder, which we leverage at a sketch-specific “pre-text” task. Our dataset enables for the first time research on freehand scene sketch understanding and its practical applications. We release the dataset under CC BY-NC 4.0 license: [FS-COCO dataset](#)<sup>1</sup>.

## 1 Introduction

As research on sketching thrives [24, 47, 18, 6], the focus shifts from an analysis of quick single-object sketches [8, 7, 46, 9] to an analysis of scene sketches [68, 20, 33, 13], and professional [22] or specialised [60] sketches. In the age of data-driven computing, conducting research on sketching requires representative datasets. For instance, the inception of object-level sketch datasets [24, 51, 65, 47, 18, 23] enabled and propelled research in diverse applications [6, 5, 14]. Recently, increasingly more attempts are conducted towards not only collecting the data but also understanding how humans sketch [23, 6, 25, 64, 61]. We extend these efforts to scene sketches by introducing FS-COCO (Freehand Sketches of Common Objects in COntext), the first dataset of 10,000 unique freehand scene sketches, drawn

<sup>1</sup> <https://fscoco.github.io>

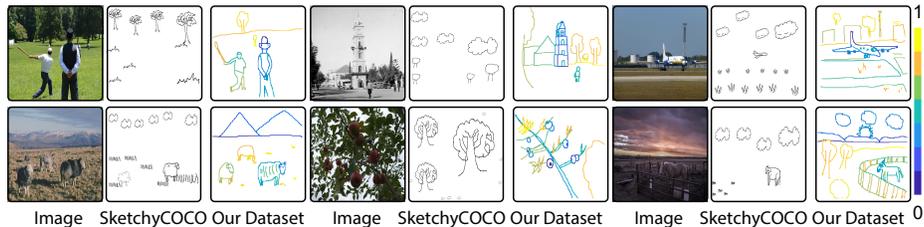


Fig. 1: Comparison of our sketches to the scene sketches from *SketchyCOCO*, the latter are obtained by combining together sketches of individual objects. Our freehand scene sketches contain abstraction at the object and scene level and better capture the content of reference scenes. This figure demonstrates a large domain gap between freehand scene sketches and available scene sketches, motivating the need for new datasets. Our sketches contain stroke temporal order information, which we visualize using the “Parula” color scheme: strokes in “blue” are drawn first, strokes in “yellow” are drawn last.

by 100 non-expert participants. We envision this dataset to permit a multitude of novel tasks and to contribute to the fundamental understanding of visual abstraction and expressivity in scene sketching. With our work, we make the first stab in this direction: We study fine-grained image retrieval from freehand scene sketches and the task of scene sketch captioning.

Thus far, research on scene sketches leveraged semi-synthetic [20, 33, 68] datasets that are obtained by combining together sketches and clip-arts of individual objects. Such datasets lack the holistic scene-level abstraction that characterises real scene sketches. Fig. 1 shows a visual comparison between the existing semi-synthetic [20] scene sketch dataset and ours FS-COCO. It shows interactions between scene elements in our sketches and diversity of objects depictions. Moreover, our sketches contain more object categories than previous datasets: Our sketches contain more than 92 categories from the COCO-stuff [10], while sketches in *SketchyScene* [68] and *SketchyCOCO* [20] contain 45 and 17 object categories, respectively.

Our dataset collection setup is practical applications-driven, such as the retrieval of a video frame given a quick sketch from memory. This is an important task because, while the text-based retrieval achieved impressive results in recent years, it might be easier to communicate via sketching fine-grained details. However, this will only be practical if users can provide a quick sketch and are not expected to be good sketchers. Therefore, we collect *easy to recognize but quick to create* freehand scene sketches from recollection (similar to object sketches collected previously [18, 47]). As reference images, we select photos from the MS-COCO [32], a benchmark dataset for scene understanding that ensures diversity of scenes and is complemented with rich annotations in a form of semantic segmentation and image captions.

Equipped with our FS-COCO dataset, we for the first time study the problem of a fine-grained image retrieval from freehand scene sketches. First, we show the presence of a domain gap between freehand sketches and semi-synthetic ones [68, 20], which are easier to collect, on the example of fine-grained sketch-

based image retrieval. Then, in our work we aim at understanding how scene-sketch-based retrieval compares to text-based retrieval, and what information sketch captures. To obtain a thorough understanding, we collect for each sketch its text description. The text description makes the subject who created the sketch, eliminating the noise due to sketch interpretation. By comparing sketch text descriptions with image text descriptions from the MS-COCO [32] dataset, we draw conclusions on the complementary nature of the two modalities: sketches and image text descriptions.

Our dataset of freehand scene sketches enables analysis towards insights into how humans sketch scenes, not possible with earlier datasets [20]. We continue the recent trend on understanding and leveraging strokes order [23, 6, 22, 61] and observe the same trends of coarse-to-fine sketching in scene sketches: We study stroke order as a factor of its salience for retrieval. Finally, we study sketch-captioning as an example of a sketch understanding task.

Collecting human sketches is costly, and despite our dataset being relatively large-scale, it is hard to reach the scale of the existing datasets of photos [53, 49, 38]. To tackle this known problem of sketch data, recent work [39, 5] to improve the performance of the encoder-decoder-based architectures on the downstream tasks proposed to pre-train the encoder relying on some auxiliary task. In our work, we build on [5] and consider the auxiliary task of raster sketch to vector sketch generation. Since our sketches are more complex than those of single objects considered before, we propose a dedicated hierarchical RNN decoder. We demonstrate the efficiency of the pre-training strategy and our proposed hierarchical decoder on fine-grained retrieval and sketch-captioning.

In summary, our contributions are: (1) We propose the first dataset of freehand scene sketches and their captions; (2) We study for the first time fine-grained freehand-scene-sketch-based image retrieval (3) and the relations between sketches, images and their captions. (4) Finally, to address the challenges of scaling sketch datasets and complexity of scene sketches, we introduce a novel hierarchical sketch decoder that exploit temporal stroke order available for our sketches. We leverage this decoder at the pre-training stage for fine-grained retrieval and sketch captioning.

## 2 Related Work

*Single-Object Sketch Datasets* Most freehand sketch datasets contain sketches of individual objects, annotated at the category level [18, 24] or part level [21], paired to photos [65, 47, 51] or 3D shapes [43]. Category-level and part-level annotations enable tasks such as sketch recognition [66, 48] and sketch generation [21, 6]. *Paired* datasets allow to study practical tasks such as sketch-based image retrieval [65] and sketch-based image generation [59].

However, collecting fine-grained paired datasets is time-consuming since one needs to ensure accurate, fine-grained matching while keeping the sketching task natural for the subjects [27]. Hence, such paired datasets typically contain a few thousand sketches per category, *e.g.*, QMUL-Chair-V2 [65] consists of 1432

Table 1: Properties of scene sketch datasets.

Dataset	Abstraction		# pho-	Stroke	Cap-	Free-
	Object	Scene	tos	temporal order	tions	hand
SketchyScene [68]	✓	✗	7,264	✗	✗	✗
SketchyCOCO [20]	✗	✓	14,081	✗	✗	✗
<b>FS-COCO</b>	✓	✓	10,000	✓	✓	✓

sketch-photo pairs on a single ‘chair’ category, Sketchy [47] has an average of 600 sketches per category, albeit over 125 categories.

Our dataset contains *10,000 scene sketches*, each paired with a ‘reference’ photo and text description. It contains scene sketches rather than sketches of individual objects and excels the existing fine-grained datasets of single-object sketches in the amount of paired instances.

*Scene Sketch Datasets* Probably the first dataset of 8,694 freehand scene sketches was collected within the multi-modal dataset [3]. It contains sketches of 205 scenes, but the examples are not paired between modalities. Scene sketch datasets with the pairing between modalities [68,20] have started to appear, however they are ‘*semi-synthetic*’. Thus, the SketchyScene [68] dataset contains 7,264 sketch-image pairs. It is obtained by providing participants with a reference image and clip-art like object sketches to drag-and-drop for scene composition. The augmentation is performed by replacing object sketches with other sketch instances belonging to the same object category. SketchyCOCO [20] was generated automatically relying on the segmentation maps of photos from COCO-Stuff [10] and leveraging freehand sketches of single objects from [47,18,24].

Leveraging the semi-synthetic datasets, previous work studied scene sketch semantic segmentation [68], scene-level fine-grained sketch based image retrieval [33], and image generation [20]. Nevertheless, sketches in the existing datasets are not representative of freehand human sketches as shown in Fig. 1, and therefore the existing results can be only considered preliminary. Unlike existing semi-synthetic datasets, our dataset of freehand scene sketches captures abstraction at the object level and holistic scene level, and contains stroke temporal information. We provide a comparative statistics with previous datasets in Tab. 1, discussed in Sec. 4.1. We demonstrate the benefit and importance of the newly proposed data on two problems: image retrieval and sketch captioning.

### 3 Dataset Collection

Targeting practical applications, such as sketch-based image retrieval, we aimed to collect representative freehand scene sketches with object- and scene-levels of abstraction. Therefore, we define the following requirements towards collected sketches: (1) created by non-professionals, (2) fast to create, (3) recognizable, (4) paired with images, and (5) supplemented with sketch-captions.

*Data preparation* We randomly select 10k photos from MS-COCO [32], a standard benchmark dataset for scene understanding [45,12,11]. Each photo in this

dataset is accompanied by image captions [32] and semantic segmentation [10]. Our selected subset of photos includes 72 “things” instances (well-defined foreground objects) and 78 “stuff” instances (background instances with potentially no specific or distinctive spatial extent or shape: e.g., “trees”, “fence”), according to the classification introduced in [10]. We present detailed statistics in Sec. 4.1.

*Task* We built a custom web application<sup>2</sup> to engage 100 participants, each annotating a distinct subset of 100 photos. Our objective is to collect easy-to-recognize freehand scene sketches drawn from memory, alike single-object sketches collected previously [18, 47]. To imitate real world scenario of sketching from memory, following the practice of single object dataset collection, we showed a reference scene photo to a subject for a limited duration of 60 seconds, determined through a series of pilot studies. To ensure recognizable but not overly detailed drawings, we also put time limits on the duration of the sketching. We determined the optimal time limits through a series of pilot studies with 10 participants, which showed that 3 minutes were sufficient for participants to comfortably sketch recognizable scene sketches. We allow repeated sketching attempts, with the subject making an average of 1.7 attempts. Each attempt repeats the entire process of observing an image and drawing on a blank canvas. Upon satisfaction with their sketch, we ask the same subject to describe their sketch in text. The instructions to write a sketch caption are similar to that of Lin *et al.* [32] and are provided in supplemental materials. To reduce fatigue that can compromise data quality, we encourage participants to take frequent breaks and complete the task over multiple days. Thus, each participant spent 12 – 13 hours to annotate 100 photos over an average period of 2 days.

*Quality check* We check the quality of sketches. We hired as a *human judge* one appointed person (1) with experience in data collection and (2) non-expert in sketching. The human judge instructed to “mark sketches of scenes that are *too difficult to understand or recognize.*” The tagged photos were sent back to their assigned annotator. This process guarantees the resulting scene sketches are recognizable by a human, and therefore, should be understood by a machine.

*Participants* We recruited 100 non-artist participants from the age group 22–44, with an average age of 27.03, including 72 males and 28 females.

## 4 Dataset composition

Our dataset consists of 10,000 (a) unique freehand scene sketches, (b) textual descriptions of the sketches (sketch captions), (c) reference photos from the MSCOCO [32] dataset. Each photo in [32] contains 5 associated text descriptions (image captions) by different subjects [32]. Figs. 1 and 3 show samples from our dataset, and supplemental materials visualize more sketches from our dataset.

<sup>2</sup> <https://github.com/pinakinathc/SketchX-SST>

Table 2: Comparison of scene sketch datasets based on the distribution of categories in sketch-image pairs. ‘FG’ denotes subsets of datasets that are recommended for use in Fine-Grained tasks, such as fine-grained retrieval.  $e_l/e_c$  denotes estimates based on semantic segmentation labels in images and based on the occurrence of a word in a sketch caption, respectively. See Sec. 4 for details.

Dataset	# photos	# categories	# categories per sketch				# sketches per category			
			Mean	Std	Min	Max	Mean	Std	Min	Max
SketchyScene [68]	7,264	45	7.88	1.96	4	20	1079.76	1447.47	31	5723
SketchyCOCO [20]	14,081	17	3.33	0.9	2	7	1932.41	3493.01	33	9761
SketchyScene FG	2,724	45	7.71	1.88	4	20	394.51	540.30	3	2154
SketchyCOCO FG	1,225	17	3.28	0.89	2	6	164.71	297.79	5	824
FS-COCO ( $e_c$ )	10,000	92	1.37	0.57	1	5	99.42	172.88	1	866
FS-COCO ( $e_l$ )	10,000	150	7.17	3.27	1	25	413.18	973.59	1	6789

#### 4.1 Comparison to existing datasets

Tab. 2 provides comparison with previous dataset and statistics on distribution of object categories in our sketches, which we discuss in more detail below.

*Categories* First, we obtain a joint set of labels from the labels in [68,20] and [10]. To compute statistics on the categories present in [68,20], we use the semantic segmentation labels available in these datasets. For our dataset, we compute two estimates of the category distribution across our data: (1)  $e_l$ , based on semantic segmentation labels in images and (2)  $e_c$ , based on the occurrence of a word in a sketch caption. As can be seen from Fig. 3, the participants do not exhaustively describe in the caption all the objects present in sketches. Our dataset contains  $e_c/e_l = 92/150$  categories, which is more than double the number of categories in previous scene sketch datasets (Tab. 2). On average, each category is present in  $e_c/e_l = 99.42/413.18$  sketches. Among the most common category in all three datasets are ‘cloud’, ‘tree’ and ‘grass’ common to outdoor scenes. In our dataset ‘person’ is also among one of the most frequent categories along with common animals such as ‘horse’, ‘giraffe’, ‘dog’, ‘cow’ and ‘sheep’. Our dataset, according to lower/upper estimates, contains 33/71 indoor categories and 59/79 outdoor categories. We provide detailed statistics in supplemental materials.

*Sketch complexity* Existing datasets of freehand sketches [18,47] contain sketches of single objects. The complexity of scene sketches is unavoidably higher than the one of single-object sketches. Sketches in our dataset have a median stroke count of 64. For comparison, a median strokes count in the popular Tu-Berlin [18] and Sketchy [47] datasets is 13 and 14, respectively.

## 5 Towards scene sketch understanding

### 5.1 Semi-synthetic versus freehand sketches

To study the domain gap between existing ‘semi-synthetic’ and our freehand scene sketches, we evaluate the state-of-the-art methods for Fine Grained Sketch Based Image Retrieval (FG-SBIR) on the three datasets: SketchyCOCO [20], SketchyScene [68] and FS-COCO (ours) (Tab. 3).

Table 3: Evaluation of a domain gap between ‘semi-synthetic’ sketches [68, 20] and freehand sketches FS-COCO. The details on the compared methods are in Sec. 5.1. Top-1/Top-10 accuracy (R@1/R@10) is the percentage of test sketches for which the ground-truth image is among the first 1/10 ranked retrieval results.

Methods	Trained On																	
	SketchyScene (S-Scene) [68]						SketchyCOCO (S-COCO) [20]						FS-COCO (Ours)					
	Evaluate on						Evaluate on						Evaluate on					
	S-Scene		S-COCO		FS-COCO		S-Scene		S-COCO		FS-COCO		S-Scene		S-COCO		FS-COCO	
R@1	R@10	R@1	R@10	R@1	R@10	R@1	R@10	R@1	R@10	R@1	R@10	R@1	R@10	R@1	R@10	R@1	R@10	
Siam.-VGG16 [65]	22.8	43.5	1.1	4.1	1.8	6.6	0.3	2.1	37.6	80.6	<0.1	0.4	5.8	24.5	2.4	11.6	23.3	52.6
HOLEF [52]	22.6	44.2	1.2	3.9	1.7	5.9	0.4	2.3	38.3	82.5	0.1	0.4	6.0	24.7	2.2	11.9	22.8	53.1
CLIP zero-shot [45]	1.26	9.70	–	–	–	–	–	–	1.85	9.41	–	–	–	–	–	–	1.17	6.07
CLIP*	8.6	24.8	1.7	6.6	2.5	8.2	1.3	5.1	15.3	43.9	0.6	3.1	1.6	11.9	2.6	12.5	5.5	26.5

*Methods and training details.* *Siam.-VGG16* adapts the pioneering method of Yu et al. [65] by replacing the Sketch-a-Net [66] feature extractor with VGG16 [50] trained using triplet loss [57, 62], as we observed that this increases retrieval performance. *HOLEF* [52] extends *Siam.-VGG16* by using spatial attention to better capture fine-scale details and introducing a novel trainable distance function in the context of triplet loss.

We also explore CLIP [45], a recent method that has shown an impressive ability to generalize across multiple photo datasets [32, 42]. *CLIP (zero-shot)* uses the pre-trained photo encoder, trained on 400 million text-photo pairs that do not include photos from the MS-COCO dataset. In our experiments, we use the publicly available ViT-B/32 version<sup>3</sup> of CLIP, which uses the visual transformer backbone as a feature extractor. Finally, *CLIP\** means CLIP fine-tuned on the target data. Since we found training CLIP to be very unstable, we train only the layer normalization [4] modules and add a fully connected layer to map the sketch and photo representations to a shared 512 dimensional feature space. We train *CLIP\** using triplet loss [57, 62] with a margin value set to 0.2 with a batch size 256 and a low learning rate of 0.000001.

*Train and test splits.* We train *Siam.-VGG16* and *HOLEF*, and fine-tune *CLIP\** on the sketches from one of three datasets: SketchyCOCO [20], SketchyScene [68] and FS-COCO. For our FS-COCO dataset 70% of each user sketches are used for training and the remaining 30% for testing. This results in a training/tasting sets of 7,000 and 3,000 sketch-image pairs. For [20, 68] we use subsets of sketch-image pairs, since both datasets contain noisy data, which leads to performance degradation when used for the fine-grained tasks such as fine-grained retrieval. For SketchyCOCO [20], following Liu *et al.* [33], we sort the sketches based on the number of the foreground objects and select the top 1,225 scene sketch-photo pairs. We then randomly split those into training and test sets of 1,015 and 210 pairs, respectively. For SketchyScene [68] we follow their approach used to evaluate image retrieval, and manually select sketch-photo pairs that have same categories present in images and sketches. We obtain training and test sets of 2,472 and 252 pairs, respectively. The statistics on object categories in these subsets are given in Tab. 2 (‘FG’). Note that in each experiment, the image

<sup>3</sup> <https://github.com/openai/CLIP>

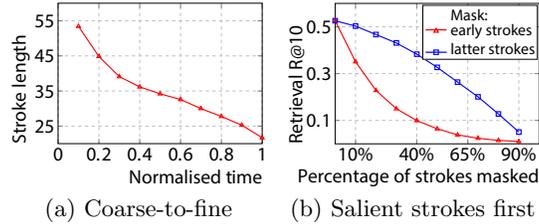


Fig. 2: Sketching strategies in our freehand scene sketches: Sec. 5.2. (a) Humans follow a coarse-to-fine sketching strategy, drawing longer strokes first. (b) Humans draw strokes more salient for the retrieval task early on. We plot the Top-10 (R@10) retrieval accuracy when certain strokes during testing are masked out. Top-10 accuracy calculates the percentage of test sketches for which the ground-truth image is among the first 10 ranked retrieval results.

gallery size is equal to the test set size. Therefore, in the case of our dataset, the retrieval is performed among the largest number of images.

*Evaluation.* Tab. 3 shows that training on ‘semi-synthetic’ sketch datasets like SketchyCOCO [20] and SketchyScene [68] does not generalize to freehand scene sketches from our dataset: training on FS-COCO / SketchyCOCO / SketchyScene and testing on our data results in  $R@1$  of 23.3 /  $< 0.1$  / 1.8. Training with the sketches from [68] rather than from [20] results in better performance on our sketches, probably due to the larger variety of categories in [68] (46 categories) than in [20] (17 categories). Tab. 3 also shows a large domain gap between all three datasets.

As the image gallery is larger when tested on our sketches than for other datasets, the performance on our sketches in Tab. 3 is lower, even when trained on our sketches. For a fairer comparison, we create 10 additional test sets consisting of 210 sketch-image pairs (the size of the SketchyCOCO dataset’s image gallery) by randomly selecting them from the initial set of 3000 sketches. For Siam-VGG16, the average retrieval accuracy and its standard deviation over ten splits are: Top-1 is  $50.39\% \pm 2.15\%$  and Top-10 is  $89.38\% \pm 2.0\%$ . For  $CLIP^*$ , the average retrieval accuracy and its standard deviation over ten splits are: Top-1 is  $42.53\% \pm 3.16\%$  and Top-10 is  $87.93\% \pm 2.14\%$ . These high performance numbers show the high quality of the sketches in our dataset.

## 5.2 What does a freehand sketch capture?

**Sketching strategy** We observe that humans follow a coarse-to-fine sketching strategy in scene sketches: in Fig. 2 (a) we show that the average stroke length decreases with time. Similarly, coarse-to-fine sketching strategies has previously been observed in single object sketch datasets [18, 47, 23, 61]. We also verify the hypothesis that humans draw salient and recognizable regions early [6, 18, 47]. We first train the classical SBIR method [65] on sketch-image pairs from our dataset: 70% of each user’s sketches are used for training and 30% for testing. During the evaluation, we follow two strategies: (i) We gradually mask out a

certain percentage of strokes drawn early, which is indicated by the red line in Fig. 2 (b). (ii) We then gradually mask out strokes drawn towards the end, which is indicated by the blue line in Fig. 2 (b). We observe that masking strokes towards the end has a smaller impact on the retrieval accuracy than masking early strokes. Thus we quantify that humans draw longer (Fig. 2a) and more salient for retrieval (Fig. 2b) strokes early on.

**Sketch captions vs. image captions** To gain insights into what information sketch captures, we compare sketch and image captions (Fig. 3 and 4). The vocabulary of our sketch captions matches 81.50% vocabulary of image captions. Specifically, comparing sketch and image captions for each instance reveals that on average 66.5% words in sketch captions are common with image captions, while 60.8% of words overlap among the 5 available captions of each image. This indicates that sketches preserve a large fraction of information in the image. However, the sketch captions in our dataset are on average shorter (6.55 words) than image captions (10.46). We explore this difference in more detail by visualizing the word clouds for sketch and image captions. From Fig. 4 we observe that, unlike image captions, sketch descriptions do not use “color” information. Also, we compute the percentage of nouns, verbs, and adjectives in sketch and image captions. Fig. 4(c) shows that our sketch captions are likely to focus more on objects (i.e., nouns like “horse”) and their actions (i.e., verbs like “standing”) instead of focusing on attributes (i.e., adjectives like “a brown horse”).

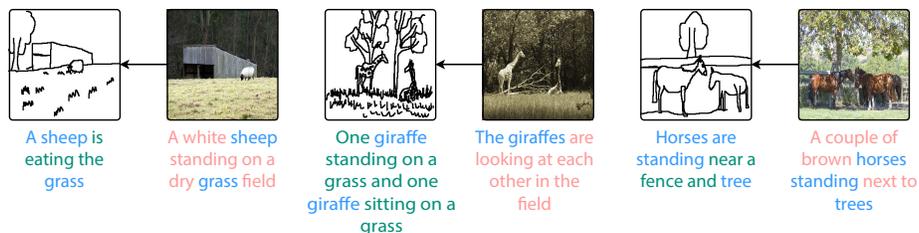


Fig. 3: A qualitative comparison of image and sketch captions. The overlapping words are marked in blue, the words present only in image-captions are marked in red, while the words present only in sketch-captions are marked in green.

**Freehand sketches vs. image captions** To understand the potential of quick freehand scene sketches in image retrieval, we compare freehand scene sketch with textual description as queries for fine-grained image retrieval (Tab. 4).

*Methods.* For text-based image retrieval, we evaluate two baselines: (1) *CNN-RNN* the simple and classic approach where text is encoded with an LSTM and images are encoded with a CNN encoder (VGG-16 in our implementation) [56, 28], and (2) CLIP [45] which is one of state-of-the-art methods alongside [29] in text-based image retrieval. For purity of experiments we evaluate here CLIP, as its training data did not include MS-COCO dataset from which the reference



Fig. 4: (a,b) Word clouds show frequently occurring words in image and sketch captions, respectively. The larger the word, the more frequent it is. It shows that color information such as “white”, “green” is present in image captions but is missing from sketch captions. (c) Percentage of nouns, verbs, and adjectives in image and sketch captions, and their overlapping words.

images in our dataset are coming from. *CLIP zero-shot* uses off-the-shelf ViT-B/32 weights. *CLIP\** is fine-tuned on our sketch-captions by fine-tuning only layer normalization modules [4] with batch size 256 and learning rate  $1e-7$ .

*Training details.* *CNN-RNN* and *CLIP\** are trained with triplet loss [57, 62], with a margin value is set to 0.2. We use the same split to train/test sets as in Sec. 5.1. For retrieval from image captions, we randomly select one of 5 available caption versions.

*Evaluation.* Tab. 4 shows that image captions result in better retrieval performance compared to sketch captions, which we attribute to the color information in image captions. However, we observe that *CLIP\**-based retrieval from image captions is slightly inferior to *Siam.-VGG16*-based retrieval from sketches. Note that *CLIP\** is pre-trained on 400 million text-photo pairs, while *Siam.-VGG16* was trained on a much smaller set of 7000 sketch-photo pairs. Therefore, with even larger sketch datasets the retrieval accuracy from sketches will further increase. There is an intuitive explanation for this since scene sketches intrinsically encode fine-grained visual cues that are difficult to convey in text.

Table 4: Text-based versus sketch-based image retrieval.

Methods	Retrieval accuracy					
	Image Captions		Sketch Captions		Sketches	
	R@1	R@10	R@1	R@10	R@1	R@10
Siam.-VGG16 [65]	–	–	–	–	<b>23.3</b>	<b>52.6</b>
CNN-RNN [51]	11.1	31.1	7.2	23.6	–	–
CLIP zero-shot [45]	21.0	50.9	11.5	35.3	1.17	6.07
CLIP*	<u>22.1</u>	<u>52.3</u>	14.8	36.6	5.5	26.5

**Text and sketch synergy** While we have shown that scene sketches have strong ability in expressing fine-grained visual cues, image captions convey additional information such as “color”. Therefore, we are exploring whether the

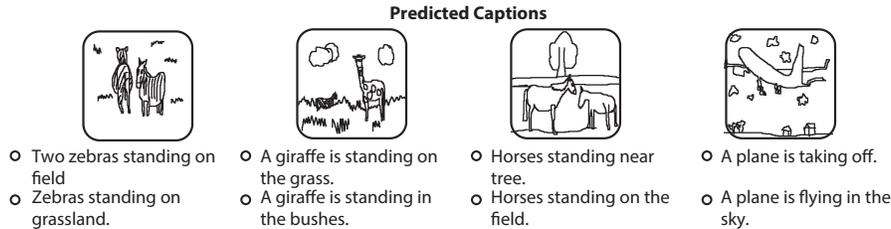


Fig. 5: Qualitative results showing predicted captions from LNFMM (H-Decoder) for scene sketches from our dataset.

two query modalities combined can improve fine-grained image retrieval. Following [34], we use two simple approaches to combine sketch and text: (-concat) we concatenate sketch and text features and (-add) we add sketch and text features. The combined features are then passed through a fully connected layer. Comparing the results in Tab. 5 and Tab. 4 shows that combining image captions and scene sketches improves fine-grained image retrieval. This confirms that the scene sketch complements the information conveyed by the text.

Table 5: Fine-grained image retrieval from the combined input of scene sketches and textual image descriptions.

Methods		R@1	R@10	Methods	R@1	R@10
CNN-RNN [51]	-add	<b>25.3</b>	<b>55.0</b>	CLIP* -add	23.9	53.5
CNN-RNN [51]	-concat	24.3	53.9	CLIP* -concat	23.3	52.6

Table 6: Sketch captioning (Sec. 5.3): our dataset enables captioning of scene sketches. We provide the results of the popular captioning methods developed for photos. For the evaluation, we use the standard metrics: BELU (B4) [40], METEOR (M) [15], ROUGE (R) [30], CIDEr (C) [55], SPICE (S) [1].

Methods	B4	M	R	C	S
Xu <i>et al.</i> [63]	13.7	17.1	44.9	69.4	14.5
AG-CVAE [58]	16.0	18.9	49.1	80.5	15.8
LNFMM [35]	16.7	21.0	52.9	90.1	16.0
LNFMM with pre-training (H-Decoder)	<b>17.3</b>	<b>21.1</b>	<b>53.2</b>	<b>95.3</b>	<b>17.2</b>

### 5.3 Sketch Captioning

While scene sketches are a pre-historic form of human communication, scene sketch understanding is nascent. Existing literature has solidified captioning as a hallmark task for scene understanding. The lack of paired scene-sketch and text datasets is the biggest bottleneck. Our dataset allows us to study this problem for the first time. We evaluate several popular and SOTA methods in Tab. 6: Xu *et al.* [63] is one of the first popular works to use the attention mechanism with an LSTM for image captioning. AG-CVAE [57] is a SOTA image captioning

model that uses a variational auto-encoder along with an additive gaussian prior. Finally, LNFMM [35] is a recent SOTA approach using normalizing flows [17] to capture the complex joint distribution of photos and text. We show qualitative results in Fig. 5 using the LNFMM model with the pre-training strategy we introduce in Sec. 6.

## 6 Efficient “pretext” task

Our dataset is large (10,000 scene sketches!) for a sketch dataset. However, scaling it up to millions of sketch instances paired with other modalities (photos/text) to match the size of the photo datasets [53] might be intractable in the short term. Therefore, when working with freehand sketches, it is important to find ways to go around the limited dataset size. One traditional approach to address this problem is to solve an auxiliary or “pretext” task [67,41,37]. Such tasks exploit self-supervised learning, allowing to pre-train the encoder for the ‘source’ domain leveraging unpaired/unlabeled data. In the context of sketching, solving jigsaw puzzles [39] and converting raster to vector sketch [5] “pretext” tasks were considered. We extend the state-of-the-art sketch-vectorization [5] “pretext” task to support the complexity of scene sketches, exploiting the availability of time-space information in our dataset. We pre-train a raster sketch encoder with the newly proposed decoder that reconstructs a sketch in a vector format as a sequence of stroke points. Previous work [5] leverages a single layer Recurrent Neural Network (RNN) for sketch decoding. However, it can only reliably model up to around 200 stroke points [24], while our scene sketches can contain more than 3000 stroke points, which makes modeling scene sketches challenging. We observe that, on average, scene sketches consist of only 74.3 strokes, with each stroke containing around 41.1 stroke points. Modeling such number of strokes or stroke points *individually* is possible using a standard LSTM network [26]. Therefore, we propose a novel 2-layered hierarchical LSTM decoder.

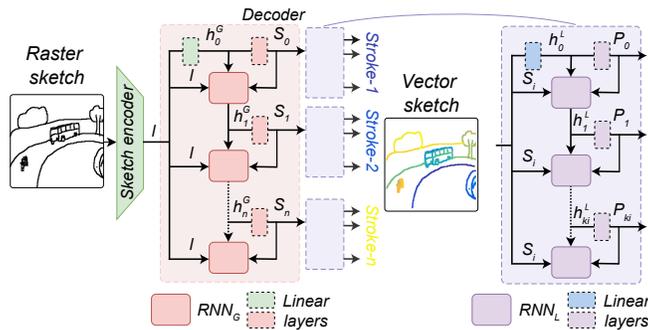


Fig. 6: The proposed hierarchical decoder used for pre-training a sketch encoder.

## 6.1 Proposed Hierarchical Decoder (H-Decoder)

We denote a raster sketch encoder that our proposed decoder pre-trains as  $E(\cdot)$ . Let the output feature map of  $E(\cdot)$  be  $F \in \mathbb{R}^{h' \times w' \times c}$ , where  $h'$ ,  $w'$  and  $c$  denotes height, width, and number of channels, respectively. We apply a global max pooling to  $F$ , with consequent flattening, to obtain a latent vector representation of the raster sketch,  $l_R \in \mathbb{R}^{512}$ .

Naively decoding  $l_R$  using a single layer RNN is intractable [24]. We propose a two-level decoder consisting of two LSTMs, referred to as global and local. The global LSTM (RNN<sub>G</sub>) predicts a sequence of feature vectors, each representing a stroke. The second local LSTM (RNN<sub>L</sub>) predicts a sequence of points for any stroke, given its predicted feature vector.

We initialize the hidden state of the global RNN<sub>G</sub> using a linear embedding as follows:  $h_0^G = W_h^G l_R + b_h^G$ . The hidden state  $h_i^G$  of decoder RNN<sub>G</sub> is updated as follows:  $h_i^G = \text{RNN}_G(h_{i-1}^G; [l_R, S_{i-1}])$ , where  $[\cdot]$  stands for a concatenation operation and  $S_{i-1} \in \mathbb{R}^{512}$  is the last predicted stroke representation computed as:  $S_i = W_y^G h_i^G + b_y^G$ .

Given each stroke representation  $S_i$ , the initial hidden state of local RNN<sub>L</sub> is obtained as:  $h_0^L = W_h^L S_i + b_h^L$ . Next,  $h_j^L$  is updated as:  $h_j^L = \text{RNN}_L(h_{j-1}^L; [S_i, P_{t-1}])$ , where  $P_{t-1}$  is the last predicted point of the  $i$ -th stroke. A linear layer is used to predict a point:  $P_t = W_y^L h_j^L + b_y^L$ , where  $P_t = (x_t, y_t, q_t^1, q_t^2, q_t^3)$  is of size  $\mathbb{R}^{2+3}$  whose first two logits represent absolute coordinate  $(x, y)$ , and the later three denote the pen’s state  $(q_t^1, q_t^2, q_t^3)$  [24].

We supervise the prediction of the absolute coordinate and pen state using the mean-squared error and categorical cross-entropy loss, as in [5].

## 6.2 Evaluation & Discussion

We use our proposed H-Decoder for pre-training a raster sketch encoder for fine-grained image retrieval (Tab. 7) and sketch captioning (Tab. 6).

*Training details* We start pre-training VGG-16 based *Siam.VGG16* (Tab. 7) and *LNFMM* (Tab. 6) encoders on QuickDraw [24], a large dataset of freehand object sketches, by coupling a VGG16 raster sketch encoder with our H-Decoder. For *CLIP\** we start from the model weights in ViT-B/32. We then train *CLIP\** and VGG-16-based encoders with our “pretext” task on *all* sketches from our dataset. We exploit here that the test data is available but does not have the paired data – captions, photos. After pre-training, training for downstream tasks starts with the weights learned during pre-training.

*Evaluation* Tab. 6 shows the benefit of the pre-training with the proposed decoder. With this pre-training strategy the performance of LNFMM [35] on sketches approaches the performance on images (CIDEr score of 98.4<sup>4</sup>), increasing, *e.g.*, the CIDEr score from 90.1 to 95.3.

<sup>4</sup> The performance of image captioning goes up to 170.5 when 100 generated captions are evaluated against the ground-truth instead of 1

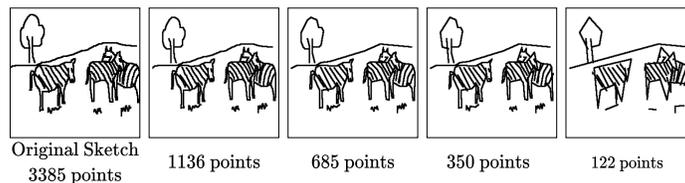


Fig. 7: Simplifying scene sketch with the RDP algorithm loses salient information. RNNs can reliably model around 200 points. The training of a single-layer RNN exploits the simplification level of the most right image.

Table 7: The role of pre-training with H-Decode in retrieval.

Method	Baseline		H-Decoder	
	R@1	R@10	R@1	R@10
Siam.-VGG16	23.3	52.6	<b>24.1</b>	<b>54.3</b>
CLIP*	5.5	26.5	5.7	27.1

This pre-training also slightly improves the performance of sketch-based retrieval (Tab. 7). Next, we compare pre-training with the proposed H-Decoder and a more naive approach. We simplify scene sketches with the Ramer-Douglas Peucker (RDP) algorithm (Fig. 7): On average, the simplified sketches contain 165 stroke points, while the original sketches contain 2437 stroke points. Then, we pre-train with a single layer RNN, as proposed in [5]. In this case *Siam.VGG16* achieves  $R@10$  of 52.1, which is lower than the performance without pre-training (Tab. 7). This further demonstrates the importance of the proposed hierarchical decoder to scene sketches.

## 7 Conclusion

We introduce the first dataset of freehand scene sketches with fine-grained paired text information. With the dataset, we took the first step towards freehand scene sketch understanding, studying tasks such as fine-grained image retrieval from scene sketches and scene sketches captioning. We show that relying on off-the-shelf methods and our data promising image retrieval and sketch captioning accuracy can be obtained. We hope that future work will leverage our findings to design dedicated methods exploiting the complementary information in sketches and image captions. In the supplemental materials, we provide a thorough comparison of modern encoders and state-of-the-art methods, and show how meta-learning can be used for few-shot sketch adaptation to an unseen user style. Finally, we proposed a new RNN-based decoder that exploits time-space information embedded in our sketches for a ‘pre-text’ task, demonstrating substantial improvement on sketch-captioning. We hope that our dataset will promote research on image generation from freehand scene sketches, sketch captioning, and novel sketch encoding approaches that are well suited for the complexity of freehand scene sketches.

## References

1. Anderson, P., Fernando, B., Johnson, M., Gould, S.: Spice: semantic propositional image caption evaluation. In: ECCV (2016) [11](#)
2. Antoniou, A., Edwards, H., Storkey, A.: How to train your maml. In: ICLR (2019) [34](#)
3. Aytar, Y., Castrejon, L., Vondrick, C., Pirsivash, H., Torralba, A.: Cross-modal scene networks. IEEE-TPAMI (2018) [4](#)
4. Ba, J., Kiros, J.R., Hinton, G.E.: Layer normalization. In: NIPS Deep Learning Symposium (2016) [7](#), [10](#)
5. Bhunia, A.K., Chowdhury, P.N., Yang, Y., Hospedales, T.M., Xiang, T., Song, Y.Z.: Vectorization and rasterization: Self-supervised learning for sketch and hand-writing. In: CVPR (2021) [1](#), [3](#), [12](#), [13](#), [14](#)
6. Bhunia, A.K., Das, A., Riaz Muhammad, U., Yang, Y., Hospedales, T.M., Xiang, T., Gryaditskaya, Y., Song, Y.Z.: Pixelor: A competitive sketching ai agent. so you think you can beat me? In: SIGGRAPH Asia (2020) [1](#), [3](#), [8](#)
7. Bhunia, A.K., Gajjala, V.R., Koley, S., Kundu, R., Sain, A., Xiang, T., Song, Y.Z.: Doodle it yourself: Class incremental learning by drawing a few sketches. In: CVPR (2022) [1](#)
8. Bhunia, A.K., Koley, S., Khilji, A.F.U.R., Sain, A., Chowdhury, P.N., Xiang, T., Song, Y.Z.: Sketching without worrying: Noise-tolerant sketch-based image retrieval. In: CVPR (2022) [1](#)
9. Bhunia, A.K., Sain, A., Shah, P., Gupta, A., Chowdhury, P.N., Xiang, T., Song, Y.Z.: Adaptive fine-grained sketch-based image retrieval. In: ECCV (2022) [1](#)
10. Caesar, H., Uijlings, J., Ferrari, V.: Coco-stuff: Thing and stuff classes in context. In: CVPR (2018) [2](#), [4](#), [5](#), [6](#)
11. Chen, J., Guo, H., Yi, K., Li, B., Elhoseiny, M.: Visualgpt: Data-efficient adaptation of pretrained language models for image captioning. arXiv preprint arXiv:2102.10407 (2021) [4](#)
12. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. arXiv preprint arXiv:1606.00915 (2016) [4](#)
13. Chowdhury, P.N., Bhunia, A.K., Gajjala, V.R., Sain, A., Xiang, T., Song, Y.Z.: Partially does it: Towards scene-level fg-sbir with partial input. In: CVPR (2022) [1](#)
14. Das, A., Yang, Y., Hospedales, T., Xiang, T., Song, Y.Z.: Béziersketch: A generative model for scalable vector sketches. In: ECCV (2020) [1](#), [35](#)
15. Denkowski, M.J., Lavie, A.: Meteor universal: Language specific translation evaluation for any target language. In: WMT@ACL (2014) [11](#)
16. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: NAACL (2019) [31](#)
17. Dinh, L., Krueger, D., Bengio, Y.: Nice: non-linear independent components estimation. In: ICLR, Workshop Track Proc (2015) [12](#)
18. Eitz, M., Hays, J., Alexa, M.: How do humans sketch objects? ACM Trans. Graph. (2012) [1](#), [2](#), [3](#), [4](#), [5](#), [6](#), [8](#)
19. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: ICML (2017) [34](#)
20. Gao, C., Liu, Q., Wang, L., Liu, J., Zou, C.: Sketchycoco: Image generation from freehand scene sketches. In: CVPR (2020) [1](#), [2](#), [3](#), [4](#), [6](#), [7](#), [8](#), [19](#), [21](#), [27](#), [33](#)

21. Ge, S., Goswami, V., Zitnick, C.L., Parikh, D.: Creative sketch generation. In: ICLR (2021) [3](#)
22. Gryaditskaya, Y., Hähnlein, F., Liu, C., Sheffer, A., Bousseau: Lifting freehand concept sketches into 3d. In: SIGGRAPH Asia (2020) [1](#), [3](#)
23. Gryaditskaya, Y., Sypsteyn, M., Hoftijzer, J.W., Pont, S., Durand, F., Bousseau, A.: Opensketch: a richly-annotated dataset of product design sketches. ACM Trans. Graph. (2019) [1](#), [3](#), [8](#), [21](#)
24. Ha, D., Eck, D.: A neural representation of sketch drawings. In: ICLR (2018) [1](#), [3](#), [4](#), [12](#), [13](#), [35](#)
25. Hertzmann, A.: Why do line drawings work? Perception (2020) [1](#)
26. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation (1997) [12](#)
27. Holinaty, J., Jacobson, A., Chevalier, F.: Supporting reference imagery for digital drawing. In: ICCV Workshop (2021) [3](#)
28. Karpathy, A., Fei-Fei, L.: Deep visual-semantic alignments for generating image descriptions. IEEE-TPAMI (2017) [9](#)
29. Li, X., Yin, X., Li, C., Zhang, P., Hu, X., Zhang, L., Wang, L., Hu, H., Dong, L., Wei, F., Choi, Y., Gao, J.: Oscar: Object-semantics aligned pre-training for vision-language tasks. In: ECCV (2020) [9](#)
30. Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. In: Text Summarization Branches Out (2004) [11](#)
31. Lin, H., Fu, Y., Jiang, Y.G., Xue, X.: Sketch-bert: Learning sketch bidirectional encoder representation from transformers by self-supervised learning of sketch gestalt. In: CVPR (2020) [33](#)
32. Lin, T.Y., Maire, M., Belongie, S.J., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: common objects in context. In: ECCV (2014) [2](#), [3](#), [4](#), [5](#), [7](#), [28](#), [33](#)
33. Liu, F., Zhou, C., Deng, X., Zuo, R., Lai, Y.K., Ma, C., Liu, Y.J., Wang, H.: Scenesketcher: Fine-grained image retrieval with scene sketches. In: ECCV (2020) [1](#), [2](#), [4](#), [7](#)
34. Liu, K., Li, Y., Xu, N., Natarajan, P.: Learn to combine modalities in multimodal deep learning. arXiv preprint arXiv:1805.11730 (2018) [11](#)
35. Mahajan, S., Gurevych, I., Roth, S.: Latent normalizing flows for many-to-many cross-domain mappings. In: ICLR (2020) [11](#), [12](#), [13](#), [34](#)
36. Noris, G., Sýkora, D., Shamir, A., Coros, S., Whited, B., Simmons, M., Hornung, A., Gross, M., Sumner, R.: Smart scribbles for sketch segmentation. Comp. Graph. Forum **31**(8) (2012) [21](#)
37. Noroozi, M., Favaro, P.: Unsupervised learning of visual representations by solving jigsaw puzzles. In: ECCV (2016) [12](#)
38. Ordonez, V., Kulkarni, G., Berg, T.: Im2text: Describing images using 1 million captioned photographs. In: NIPS (2011) [3](#)
39. Pang, K., Yang, Y., Hospedales, T.M., Xiang, T., Song, Y.Z.: Solving mixed-modal jigsaw puzzle for fine-grained sketch-based image retrieval. In: CVPR (2020) [3](#), [12](#)
40. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: ACL (2002) [11](#)
41. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. In: CVPR (2016) [12](#)
42. Plummer, B.A., Wang, L., Cervantes, C.M., Caicedo, J.C., Hockenmaier, J., Lazebnik, S.: Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In: ICCV (2015) [7](#)

43. Qi, A., Gryaditskaya, Y., Song, J., Yang, Y., Qi, Y., Hospedales, T.M., Xiang, T., Song, Y.Z.: Toward fine-grained sketch-based 3d shape retrieval. *IEEE-TIP* (2021) [3](#)
44. Qi, Y., Su, G., Chowdhury, P.N., Li, M., Song, Y.Z.: Sketchlattice: Latticed representation for sketch manipulation. In: *ICCV* (2021) [31](#), [33](#)
45. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020* (2021) [4](#), [7](#), [9](#), [10](#), [31](#), [33](#), [34](#)
46. Sain, A., Bhunia, A.K., Potlapalli, V., Chowdhury, P.N., Xiang, T., Song, Y.Z.: Sketch3t: Test-time training for zero-shot sbir. In: *CVPR* (2022) [1](#)
47. Sangkloy, P., Burnell, N., Ham, C., Hays, J.: The sketchy database: Learning to retrieve badly drawn bunnies. *ACM Trans. Graph.* (2016) [1](#), [2](#), [3](#), [4](#), [5](#), [6](#), [8](#), [32](#)
48. Schneider, R.G., Tuytelaars, T.: Sketch classification and classification-driven analysis using fisher vectors. In: *SIGGRAPH Asia* (2014) [3](#)
49. Sharma, P., Ding, N., Goodman, S., Soricut, R.: Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In: *ACL* (2018) [3](#)
50. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: *ICLR* (2015) [7](#), [30](#), [31](#)
51. Song, J., Song, Y.Z., Xiang, T., Hospedales, T.M.: Fine-grained image retrieval: the text/sketch input dilemma. In: *BMVC* (2017) [1](#), [3](#), [10](#), [11](#)
52. Song, J., Yu, Q., Song, Y.Z., Xiang, T., Hospedales, T.M.: Deep spatial-semantic attention for fine-grained sketch-based image retrieval. In: *ICCV* (2017) [7](#), [30](#), [33](#), [34](#)
53. Srinivasan, K., Raman, K., Chen, J., Bendersky, M., Najork, M.: Wit: Wikipedia-based image text dataset for multimodal multilingual machine learning. *arXiv preprint arXiv:2103.01913* (2021) [3](#), [12](#)
54. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: *CVPR* (2016) [31](#)
55. Vedantam, R., Zitnick, C.L., Parikh, D.: Cider: Consensus-based image description evaluation. In: *CVPR* (2015) [11](#)
56. Vinyals, O., Toshev, A., Bengio, S., Erhan, D.: Show and tell: A neural image caption generator. In: *CVPR* (2015) [9](#)
57. Wang, J., Song, Y., Leung, T., Rosenberg, C., Wang, J., Philbin, J., Chen, B., Wu, Y.: Learning fine-grained image similarity with deep ranking. In: *CVPR* (2014) [7](#), [10](#), [11](#)
58. Wang, L., Schwing, A.G., Lazebnik, S.: Diverse and accurate image description using a variational auto-encoder with an additive gaussian encoding space. In: *NeurIPS* (2017) [11](#), [34](#)
59. Wang, S.Y., Bau, D., Zhu, J.Y.: Sketch your own gan. In: *ICCV* (2021) [3](#)
60. Wang, T.Y., Ceylan, D., Popovic, J., Mitra, N.J.: Learning a shared shape space for multimodal garment design. In: *SIGGRAPH Asia* (2018) [1](#)
61. Wang, Z., Qiu, S., Feng, N., Rushmeier, H., McMillan, L., Dorsey, J.: Tracing versus freehand for evaluating computer-generated drawings. *ACM Trans. Graph.* (2021) [1](#), [3](#), [8](#)
62. Wen, Y., Zhang, K., Li, Z., Qiao, Y.: A discriminative feature learning approach for deep face recognition. In: *ECCV* (2016) [7](#), [10](#)
63. Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhutdinov, R., Zemel, R., Bengio, Y.: Show, attend and tell: Neural image caption generation with visual attention. In: *ICML* (2015) [11](#), [34](#)

64. Yan, C., Vanderhaeghe, D., Gingold, Y.: A benchmark for rough sketch cleanup. *ACM Trans. Graph.* (2020) [1](#)
65. Yu, Q., Liu, F., Song, Y.Z., Xiang, T., Hospedales, T.M., Loy, C.C.: Sketch me that shoe. In: *CVPR (2016)* [1](#), [3](#), [7](#), [8](#), [10](#), [30](#)
66. Yu, Q., Yang, Y., Song, Y.Z., Xiang, T., Hospedales, T.: Sketch-a-net that beats humans. In: *BMVC (2015)* [3](#), [7](#), [30](#)
67. Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: *ECCV (2016)* [12](#)
68. Zou, C., Yu, Q., Du, R., Mo, H., Song, Y.Z., Xiang, T., Gao, C., Chen, B., Zhang, H.: Sketchyscene: Rickly-annotated scene sketches. In: *ECCV (2018)* [1](#), [2](#), [4](#), [6](#), [7](#), [8](#), [19](#), [21](#), [22](#), [28](#), [33](#)

# Supplementary Material

## FS-COCO: Towards Understanding of Freehand Sketches of Common Objects in Context

Pinaki Nath Chowdhury<sup>1, 2</sup> Aneeshan Sain<sup>1, 2</sup> Ayan Kumar Bhunia<sup>1</sup>  
 Tao Xiang<sup>1, 2</sup> Yulia Gryaditskaya<sup>1, 3</sup> Yi-Zhe Song<sup>1, 2</sup>

<sup>1</sup>SketchX, CVSSP, University of Surrey, United Kingdom.

<sup>2</sup>iFlyTek-Surrey Joint Research Centre on Artificial Intelligence.

<sup>3</sup>Surrey Institute for People Centred AI, CVSSP, University of Surrey.

### S1 Ethical considerations in data collection

Our dataset contains scene sketches of photos with paired textual description of the sketches. It does not include any personally identifiable information. Each sketch and caption are associated only with an ID.

Prior to agreeing to participate in the data collection, each participant was informed of the purpose of the dataset: namely that the dataset would be publicly available and released as part of a research paper with potential for commercial use. The participants were asked to accept the Contributor License Agreement that explains legal terms and conditions, and in particular it specifies that the *data collector* has the rights to distribute the data under any chosen license: The participants granted to the *data collectors* and recipients of the data distributed by the data collectors a perpetual, worldwide, non-exclusive, nocharge, royalty-free, irrevocable copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, sub-license, and distribute participants contributions and such derivative works. We further requested a written confirmation from annotators that they give the *data collector* permission to conduct research on the collected data and release the dataset.

Each participant who approved these terms, was assigned a random user ID. Each participant was given the option of deleting any or all their annotations/collected data at any point during the data collection process.

We also included an anonymous public discussion forum in our annotation web portal which could be used by any participant to raise concerns and collectively inform others. Annotators were also given the option of directly contacting us to raise concerns privately.

### S2 A detailed description of FSCOCO and comparison with existing SketchyCOCO [20] and SketchyScene [68]

In Sec. 4.1 in the main document, we compare with existing datasets SketchyCOCO [20] and SketchyScene [68]. Here, we provide the detailed statistics on cat-

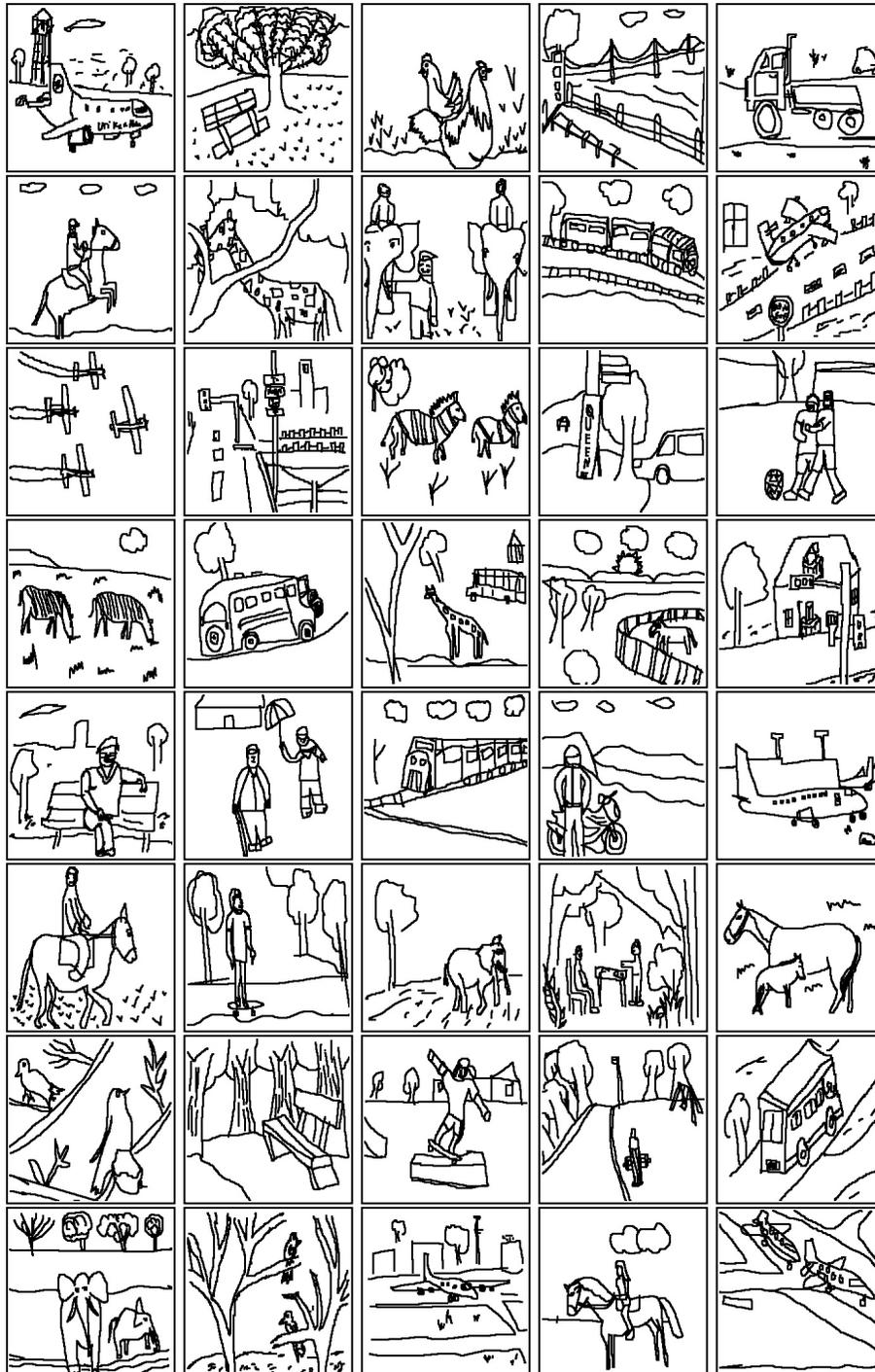


Fig. S1: Sample sketches from our FS-COCO dataset.

egories in SketchyCOCO [20] and SketchyScene [68] and our dataset in Tab. S1, Tab. S2 and Tab. S3, respectively.

Our FS-COCO includes freehand scene sketches of photos along with the textual description of the sketch. However, we did not collect stroke- or object-level annotations. One option would have been to let sketchers to assign labels by selecting a label for each stroke while sketching. Following the arguments from the previous work on data collection [23], we refrained from this option, as that could have disturbed the natural sketching process, resulting in non-representative sketches. Indeed, we observe that objects in sketches in our dataset can share certain strokes and that participants can progress on multiple objects iteratively, not sketching one object at a time. Having done a huge step towards enabling scene sketch understanding, we leave the stroke- and object-level annotations for future work. Such annotations can be done using the tools from [23] or [36]. For our dataset, we compute two estimates of category distribution: (1) based on semantic segmentation labels of images FS-COCO ( $e_l$ ), and (2) based on the occurrence of a word in a sketch caption FS-COCO ( $e_c$ ). The detailed statistics is provided in Tab. S3.

Table S1: We present a detailed list of categories in SketchyCOCO (SketchyCOCO-All) [20] along with the number of sketches that contain each category (# sketches), and the percentage of sketches that include a particular category (# percentage). SketchyCOCO-FG denotes a subset of SketchyCOCO-All that is used for fine-grained scene-level sketch-based image retrieval.

SketchyCOCO-FG			SketchyCOCO-All		
Category	# sketches	# percentage	Category	# sketches	# percentage
clouds	824	67.27	clouds	9761	69.32
tree	784	64.00	tree	9051	64.28
grass	752	61.39	grass	8857	62.90
airplane	80	6.53	airplane	944	6.70
giraffe	60	4.90	giraffe	925	6.57
horse	53	4.33	zebra	595	4.23
zebra	48	3.92	horse	519	3.69
cow	43	3.51	cow	450	3.20
dog	43	3.51	dog	367	2.61
elephant	25	2.04	elephant	351	2.49
car	23	1.88	sheep	339	2.41
sheep	22	1.80	car	255	1.81
motorcycle	14	1.14	motorcycle	139	0.99
traffic light	10	0.82	fire hydrant	112	0.80
fire hydrant	9	0.73	traffic light	96	0.68
cat	5	0.41	bicycle	57	0.40
bicycle	5	0.41	cat	33	0.23

Table S2: A detailed list of categories is presented for SketchyScene (SketchyScene-All) [68] along with the number of sketches that contain each category (# sketches), and the percentage of sketches that include a particular category (# percentage). SketchyScene-FG denotes a subset of SketchyScene-All that is used for fine-grained scene-level sketch-based image retrieval.

SketchyScene-FG			SketchyScene-All		
Category	# sketches	# percentage	Category	# sketches	# percentage
tree	2154	79.07	tree	5723	40.64
grass	2084	76.51	grass	5412	38.43
cloud	1880	69.02	cloud	5170	36.72
road	1168	42.88	road	3067	21.78
sun	1020	37.44	sun	2917	20.72
house	936	34.36	house	2841	20.18
mountain	889	32.64	people	2417	17.16
people	802	29.44	mountain	2357	16.74
flower	786	28.85	flower	2077	14.75
fence	738	27.09	fence	1857	13.19
dog	507	18.61	dog	1485	10.55
bird	463	17.00	bird	1206	8.56
car	422	15.49	car	1084	7.70
bench	334	12.26	bench	971	6.90
cow	308	11.31	cow	781	5.55
sheep	307	11.27	sheep	763	5.42
rabbit	265	9.73	cat	726	5.16
cat	259	9.51	chicken	665	4.72
bus	259	9.51	rabbit	648	4.60
chicken	249	9.14	bus	636	4.52
butterfly	224	8.22	butterfly	603	4.28
duck	212	7.78	street	567	4.03
street	194	7.12	duck	507	3.60
picnic	142	5.21	picnic	437	3.10
basket	125	4.59	basket	384	2.73
apple	107	3.93	pig	333	2.36
bee	105	3.85	apple	330	2.34
pig	103	3.78	truck	293	2.08
truck	89	3.27	bee	243	1.73
horse	73	2.68	horse	235	1.67
moon	57	2.09	grape	214	1.52
grape	54	1.98	table	197	1.40
table	54	1.98	moon	193	1.37
banana	50	1.84	banana	162	1.15
bicycle	48	1.76	bicycle	155	1.10
bucket	45	1.65	chair	138	0.98

Continued on next page

**Table S2 – continued from previous page**

Category	# sketches	# percentage	Category	# sketches	# percentage
cup	37	1.36	bucket	125	0.89
chair	37	1.36	star	114	0.81
airplane	34	1.25	airplane	110	0.78
bottle	32	1.17	cup	109	0.77
star	28	1.03	bottle	106	0.75
balloon	27	0.99	balloon	90	0.64
dinnerware	23	0.84	umbrella	59	0.42
umbrella	20	0.73	dinnerware	51	0.36
sofa	3	0.11	sofa	31	0.22

Table S3: We list all categories present in FSCOCO. For our dataset, we compute two estimates of category distribution: (1) based on semantic segmentation labels of images ( $e_l$ ), and (2) based on the occurrence of a word in a sketch caption ( $e_c$ ). We present the number of sketches (# sketches) and percentage of sketches (# percentage) containing each category.

FS-COCO ( $e_c$ )			FS-COCO ( $e_l$ )		
Category	# sketches	# percentage	Category	# sketches	# percentage
grass	866	8.66	tree	6789	67.89
road	643	6.43	grass	6486	64.86
tree	638	6.38	sky-other	5530	55.3
giraffe	637	6.37	person	3813	38.13
kite	543	5.43	building-other	2235	22.35
zebra	422	4.22	clouds	2161	21.61
horse	407	4.07	bush	1616	16.16
clock	394	3.94	metal	1404	14.04
dog	338	3.38	road	1382	13.82
cow	308	3.08	pavement	1269	12.69
sheep	305	3.05	dirt	1235	12.35
train	305	3.05	fence	1206	12.06
person	292	2.92	car	1162	11.62
bird	267	2.67	airplane	1065	10.65
elephant	232	2.32	clothes	1001	10.01
bench	206	2.06	house	935	9.35
frisbee	200	2	plant-other	916	9.16
airplane	162	1.62	frisbee	777	7.77
light	156	1.56	giraffe	770	7.7
house	156	1.56	kite	743	7.43
car	146	1.46	bird	617	6.17
bear	129	1.29	mountain	617	6.17
mountain	114	1.14	truck	608	6.08

Continued on next page

**Table S3 – continued from previous page**

Category	# sketches	# percentage	Category	# sketches	# percentage
bus	103	10.3	cow	577	5.77
skateboard	90	0.9	zebra	562	5.62
river	88	0.88	bench	544	5.44
umbrella	88	0.88	wall-concrete	529	5.29
branch	87	0.87	horse	528	5.28
fence	84	0.84	sheep	521	5.21
truck	76	0.76	clock	517	5.17
hill	71	0.71	traffic light	496	4.96
bridge	63	0.63	roof	485	4.85
boat	60	0.60	ground-other	484	4.84
wood	38	0.38	wood	452	4.52
bush	30	0.3	dog	438	4.38
rock	28	0.28	hill	434	4.34
fruit	26	0.26	branch	418	4.18
cat	25	0.25	rock	367	3.67
chair	22	0.22	stop sign	356	3.56
bicycle	22	0.22	river	333	3.33
table	20	0.2	train	333	3.33
flower	19	0.19	light	308	3.08
snow	16	0.16	gravel	301	3.01
banana	16	0.16	skateboard	294	2.94
mirror	13	0.13	backpack	293	2.93
apple	13	0.13	elephant	279	2.79
window	11	0.11	water-other	266	2.66
plate	11	0.11	textile-other	259	2.59
motorcycle	10	0.1	leaves	251	2.51
tent	10	0.1	railroad	250	2.5
stone	9	0.09	structural-other	242	2.42
sea	9	0.09	window-other	238	2.38
shoe	8	0.08	handbag	238	2.38
platform	8	0.08	stone	236	2.36
vase	7	0.07	sports ball	229	2.29
orange	7	0.07	plastic	221	2.21
leaves	5	0.05	bus	212	2.12
hat	4	0.04	wall-other	212	2.12
mat	4	0.04	umbrella	196	1.96
banner	4	0.04	wall-brick	178	1.78
metal	4	0.04	flower	178	1.78
donout	4	0.04	cage	173	1.73
railing	4	0.04	straw	172	1.72
net	3	0.03	banner	162	1.62
roof	3	0.03	bicycle	162	1.62

Continued on next page

Table S3 – continued from previous page

Category	# sketches	# percentage	Category	# sketches	# percentage
surfboard	3	0.03	motorcycle	160	1.6
bowl	3	0.03	fire hydrant	158	1.58
carrot	3	0.03	chair	155	1.55
tie	3	0.03	fog	153	1.53
bottle	3	0.03	tent	149	1.49
laptop	3	0.03	bridge	146	1.46
snowboard	3	0.03	boat	143	1.43
sand	3	0.03	bear	141	1.41
book	3	0.03	baseball bat	135	1.35
suitcase	3	0.03	wall-stone	126	1.26
cloth	3	0.03	stairs	118	1.18
cage	2	0.02	railing	115	1.15
paper	2	0.02	baseball glove	108	1.08
cup	2	0.02	wall-wood	86	0.86
pavement	2	0.02	playingfield	83	0.83
pizza	2	0.02	mud	81	0.81
door	2	0.02	furniture-other	80	0.8
bed	2	0.02	door-stuff	78	0.78
cake	2	0.02	solid-other	71	0.71
mud	2	0.02	bottle	70	0.7
toilet	1	0.01	platform	69	0.69
clothes	1	0.01	floor-other	68	0.68
toothbrush	1	0.01	ceiling-other	59	0.59
blender	1	0.01	cloth	59	0.59
railroad	1	0.01	tennis racket	56	0.56
scissors	1	0.01	potted plant	56	0.56
skyscraper	1	0.01	dining table	54	0.54
			table	47	0.47
			cell phone	46	0.46
			tie	45	0.45
			net	45	0.45
			apple	45	0.45
			snowboard	42	0.42
			suitcase	41	0.41
			wall-panel	41	0.41
			teddy bear	40	0.4
			floor-stone	40	0.4
			paper	39	0.39
			cat	37	0.37
			surfboard	35	0.35
			moss	26	0.26
			cup	25	0.25

Continued on next page

**Table S3 – continued from previous page**

Category	# sketches	#	percentage	Category	# sketches	#	percentage
				skis	25		0.25
				bowl	22		0.22
				banana	22		0.22
				vase	21		0.21
				fruit	20		0.2
				orange	19		0.19
				floor-wood	17		0.17
				mirror-stuff	16		0.16
				book	15		0.15
				parking meter	14		0.14
				blanket	12		0.12
				carboard	11		0.11
				laptop	11		0.11
				floor-tile	10		0.1
				food-other	9		0.09
				towel	9		0.09
				hot dog	8		0.08
				sandwich	7		0.07
				window-blind	6		0.06
				carrot	6		0.06
				waterdrops	6		0.06
				cake	6		0.06
				ceiling-tile	4		0.04
				toilet	4		0.04
				wall-tile	4		0.04
				fork	4		0.04
				toothbrush	4		0.04
				rug	3		0.03
				oven	3		0.03
				knife	3		0.03
				vegetable	3		0.03
				pizza	3		0.03
				remote	3		0.03
				couch	2		0.02
				donout	2		0.02
				spoon	2		0.02
				wine glass	2		0.02
				scissors	2		0.02
				mat	1		0.01
				counter	1		0.01
				hair dryer	1		0.01
				napkin	1		0.01

Continued on next page

Table S3 – continued from previous page

Category	# sketches	# percentage	Category	# sketches	# percentage
			keyboard	1	0.01

### S2.1 Indoor categories in FSCOCO

**List of Indoor categories for FSCOCO (l):** toothbrush, banner, orange, donut, pizza, metal, table, book, apple, laptop, cup, fruit, chair, mat, plate, bowl, window, door, carrot, clothes, blender, banana, light, mirror, cloth, scissors, toilet, bed, cake, paper, clock, vase, bottle

**List of Indoor categories for FSCOCO (u):** toothbrush, fork, banner, keyboard, donut, orange, knife, pizza, hot dog, metal, window-blind, table, dining table, book, apple, couch, napkin, wall-stone, laptop, floor-tile, floor-wood, rug, cup, fruit, sandwich, chair, potted plant, floor-stone, towel, blanket, ceiling-tile, mat, mirror-stuff, stairs, cell phone, bottle, counter, bowl, wall-other, door-stuff, ceiling-other, spoon, carrot, clothes, floor-other, banana, wall-brick, wall-panel, furniture-other, light, wall-concrete, window-other, cloth, scissors, hair drier, toilet, remote, textile-other, plastic, teddy bear, wine glass, paper, cardboard, cake, wall-wood, wall-tile, clock, vase, vegetable, oven, food-other

### S2.2 Outdoor categories in FSCOCO

**List of Outdoor categories for FSCOCO (l):** person, house, kite, branch, fence, mud, leaves, mountain, bush, cat, hill, skyscraper, river, umbrella, railing, boat, bridge, horse, sea, pavement, surfboard, airplane, bear, skateboard, frisbee, bird, stone, tie, train, suitcase, flower, tent, snowboard, railroad, rock, grass, motorcycle, dog, net, cow, platform, sheep, giraffe, road, sand, roof, wood, hat, truck, snow, car, shoe, bicycle, bus, tree, bench, elephant, cage, zebra.

**List of Outdoor categories for FSCOCO (u):** person, house, kite, branch, water-other, fence, mud, leaves, mountain, bush, structural-other, cat, hill, moss, fire hydrant, stop sign, dirt, straw, ground-other, river, skis, umbrella, baseball glove, railing, boat, bridge, horse, pavement, surfboard, airplane, bear, traffic light, waterdrops, building-other, bird, stone, tennis racket, train, tie, suitcase, tent, fog, railroad, flower, handbag, plant-other, snowboard, rock, grass, motorcycle, frisbee, dog, net, cow, platform, sports ball, sheep, giraffe, baseball bat, road, clouds, roof, wood, truck, car, skateboard, sky-other, playingfield, backpack, bicycle, bus, tree, gravel, bench, elephant, cage, parking meter, solid-other, zebra.

### S2.3 Categories common between FSCOCO and SketchyCOCO [20]

**List of categories common between FSCOCO (l) and SketchyCOCO:** car, grass, motorcycle, dog, horse, cow, giraffe, cat, bicycle, airplane, tree, sheep, elephant, zebra.

**List of categories common between FSCOCO (u) and SketchyCOCO:** car, grass, motorcycle, dog, horse, cow, cat, bicycle, fire hydrant, airplane, tree, traffic light, sheep, elephant, giraffe, clouds, zebra.

## S2.4 Categories common between FSCOCO and SketchyScene [68]

**List of categories common between FSCOCO (l) and SketchyScene:**  
house, fence, table, mountain, cat, apple, umbrella, horse, cup, chair, airplane, bird, flower, grass, dog, cow, banana, sheep, road, truck, car, bus, bicycle, tree, bench, bottle.

**List of categories common between FSCOCO (u) and SketchyScene:**  
house, fence, table, mountain, cat, apple, umbrella, horse, cup, chair, airplane, bird, flower, grass, dog, cow, banana, sheep, road, truck, car, bus, bicycle, tree, bench, bottle.

## S3 Data collection: Additional detail

### S3.1 Instructions for sketch captioning

The instructions for sketch captioning are similar to that of MS-COCO [32]. Namely, the subjects received the following instructions:

- Describe all the important parts of the scene.
- Do not start the sentence with “There is”.
- Do not describe unimportant details.
- Do not describe things that might have happened in the future or past.
- Do not describe what a person might say.
- Do not give proper names.
- The sentence should contain at least 5 words.

### S3.2 UI of our data collection tool

Figs. S2 to S4 shows the user interface of our data collection tool. We release the frontend and backend scripts at <https://github.com/pinakinathc/SketchX-SST>. The frontend and backend scripts communicate using REST API.

### S3.3 Sample data from our dataset

Fig. 3 shows sample scene sketches from FS-COCO. We released the dataset under CC BY-NC 4.0 license at <https://github.com/pinakinathc/fscoco>.

### S3.4 Pilot study on optimal sketching and viewing duration

As we mention in the main document in Sections 1 and 3: “To ensure recognizable but not too detailed sketches we impose a 3-minutes sketching time constraint, where the optimal time duration was determined through a series of pilot studies. A scene reference photo is shown to a subject for 60 seconds before being asked to sketch from memory. We determined the optimal time limits through a series of pilot studies with 10 participants.” Here we provide the details of the pilot study.

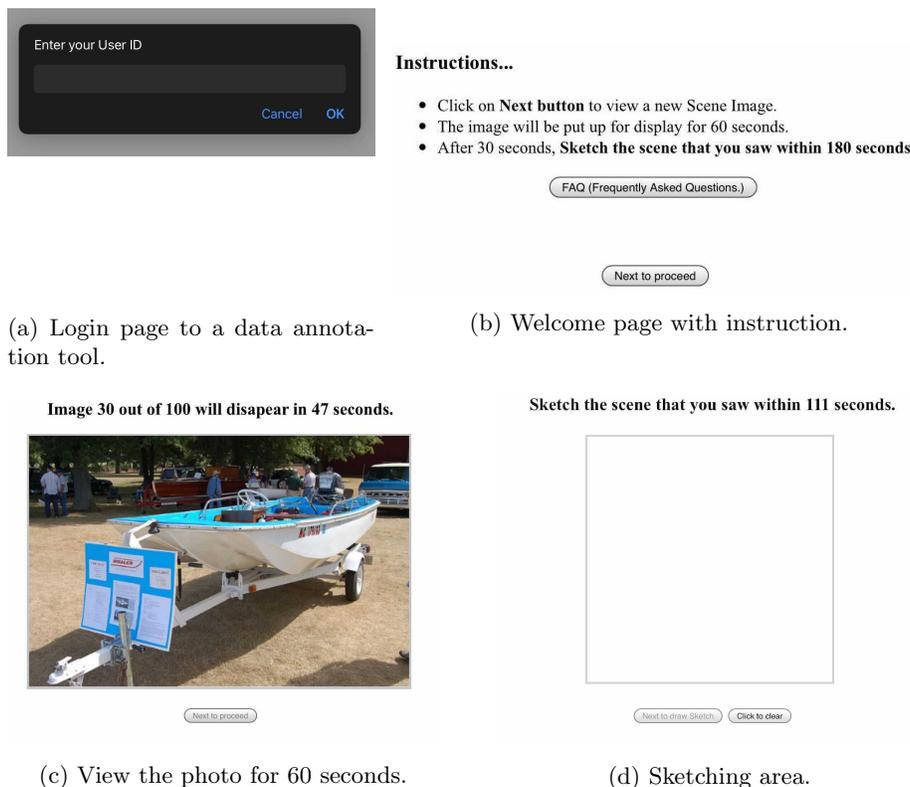


Fig.S2: User sketching interface of our data collection tool. We will release our data collection tool upon acceptance.

We find the optimal duration for viewing a reference scene photo and drawing a scene sketch by conducting a series of pilot study on 10 individuals: (i) We started with a low duration of 30 seconds to view a reference photo and 60 seconds to draw a scene sketch. This resulted in freehand sketches that were flagged as unrecognizable by our human judge. (ii) Next, we increased the drawing time to 120 seconds while keeping the viewing time to 30 seconds. Based on interviews with our human judge and annotators we conclude that while the increase in sketching time results in barely recognizable scene sketches, annotators still missed important scene information due to the short viewing duration of 30 seconds. (iii) In the final phase of our pilot study, we increased the viewing duration to 60 seconds and sketching time to 180 seconds. This helped non-expert annotators to create scene sketches in an average of 1.7 attempts that could be understood or recognized by a human judge.

In our experiments, increasing the viewing or sketching time beyond 60 and 180 seconds resulted in overly detailed sketches. Guided by practical applica-

FAQ (Frequently Asked Questions)

If you are happy with the (Sketch, Caption), enter 'Accept' button. Else you can redraw the sketch using 'Redo' button.



**Instructions for Captions.**

- Describe all the important parts of the scene.
- Do not start the sentence with "There is".
- Do not describe things that might have happened in the future or past.
- Do not describe what a person might say.
- Do not give people proper names.
- The sentence should contain at least 5 words.

Describe your Sketch here...

[Click to view/hide discussion](#)

Fig. S3: Review by an annotator before submitting a sketch and a caption. If annotators are not satisfied with the sketch, they can redo the sketch by first observing the photo and then drawing the scene sketch from scratch on a blank canvas.

tions, we limit the viewing and sketching time to a duration that allows for recognizable, but not overly detailed sketches.

## S4 Additional experiments for Sec. 5.1 in the main document: Fine-grained scene sketch-based image retrieval

We provide additional experiments for Sec. 5.1 in Tab. S5. *Siam.-SN* [65] employs triplet ranking loss with Sketch-a-Net [66] as its baseline feature extractor. *HOLEF-SN* [52] extends over *Siam.-SN* employing spatial attention along with higher-order ranking loss. Our experiments suggest inferior results using Sketch-a-Net [66] backbone feature extractor. Hence, we replace the backbone feature extractor of *Siam.-SN* with VGG16 [50], we refer to this setting as *Siam.-VGG16*. Similarly, we replace Sketch-a-Net [66] backbone in *HOLEF-SN* with VGG16: *HOLEF-VGG16*. In contrast to *Siam.-VGG16* that use a common shared encoder for both sketch and photo, we use different encoders for sketches and photos in *Heter.-VGG16*. However, we note that using separate encoders leads to an inferior result. A similar drop in performance on using a heterogenous sketch/photo encoder was previously observed by Yu *et al.* [65] for object sketch

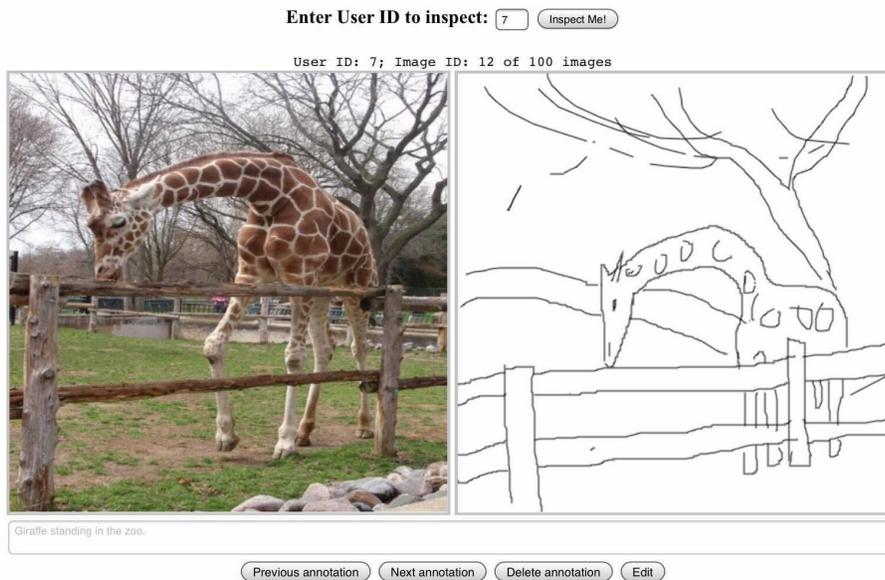


Fig. S4: One dedicated human judge evaluates if a scene sketch is recognizable or understandable. Poorly drawn scene sketches are removed and sent back to the appropriate annotator for rework.

datasets. Instead of using a CNN-based sketch encoder, *SketchLattice* adapts the graph-based sketch encoder proposed by Qi *et al.* [44]. We use a  $32 \times 32$  evenly spaced grid or lattice for sketch representation of a rasterized scene sketch. To encode photos, we use VGG16 [50]. While such a latticed sketch representation is beneficial for sketch manipulation of object sketches, an off-the-shelf adaptation for fine-grained scene sketch-based image retrieval results in inferior to VGG16 performance. In addition, we replace our sketch encoder with a BERT-like model [16] where VGG16 is used to encode photo in *SkBert-VGG16*. Since the sketch encoding module requires vector data, we only show result on our FS-COCO. *SketchyScene* is an extension of *Siam.-SN* by replacing the backbone feature extractor from Sketch-a-Net to InceptionV3 [54]. CLIP [45] is a recent state-of-the-art method that has shown an impressive generalization ability across several photo datasets. In *CLIP (zero-shot)* we use the pre-trained photo encoder from the publicly available ViT-B/32 weights<sup>5</sup> as a common backbone feature extractor for scene sketch and photo. In *CLIP-variant*, we fine-tune the layer normalization layers in CLIP using our train/test split with triplet loss, batch size 256, and a very low learning rate of 0.000001.

<sup>5</sup> <https://github.com/openai/CLIP>

#### S4.1 Are scene sketches more informative than single-object ones?

To answer this question, we evaluate the generalization ability when trained either using object sketch or scene sketches. Training and testing *Siam.-VGG16* on object (Sketchy) and our scene (FS-COCO) sketch datasets gives 43.6 and 23.3 Top-1 retrieval accuracy (R@1), respectively. Next, we perform cross-dataset evaluation where a model trained on object sketches is evaluated on scene sketch dataset and vice-versa. Tab. S4 shows that training on object and testing on scene sketches significantly reduces R@1 from 23.3 to 4.3. However, training on scene and testing on object sketches leads to a smaller drop in R@1 from 43.6 to 29.8. This indicates that scene sketches are more informative than single-object ones for the retrieval task.

Table S4: We evaluate the generalization ability of scene sketches (ours) and object sketches [47] on the fine-grained sketch-based image retrieval task (Appendix S4.1). We show a top-1 retrieval accuracy R@1 in this table.

Trained on object sketches [47]		Trained on scene sketches	
Tested on sketches (R@1):		Tested on sketches (R@1):	
object [47]	scene (ours)	object [47]	scene (ours)
43.6	4.3	29.8	23.3

#### S4.2 Additional discussion on the need for computing two estimates of the category distribution in FSCOCO.

As mentioned in Sec. 4.1 of the main document, to compute the statistics on the categories present in FSCOCO, we use two estimates: (1)  $e_l$ , based on the semantic segmentation labels in images and (2)  $e_c$ , based on the occurrence of a word in a sketch caption. The reason for using two estimates is elaborated in Fig. S5 where counting occurrence of categories in FS-COCO based on the occurrence of a word in a sketch-caption (FS-COCO ( $e_c$ )) would lead to a lower estimate. This is because participants in FS-COCO do not exhaustively describe in sketch-caption all the objects present in sketches. Simultaneously, counting occurrence of categories in FS-COCO based on the semantic segmentation labels in images (FS-COCO ( $e_l$ )) would lead to a higher estimate since not all regions in a photo are drawn by a participant.

### S5 Additional discussion for Sec. 5.2 in the main document: Fine-grained text-based image retrieval

In Sec. 5.2 in the main document, our objective is to judge, given the same amount of training data, if scene sketch or image-caption, or sketch-caption is a better query modality for fine-grained image retrieval. Our FS-COCO dataset consisting of 10,000 scene sketch, photo, image-caption, and sketch-caption is a subset of the larger MS-COCO dataset. While Oscar gives a high R@1 score of

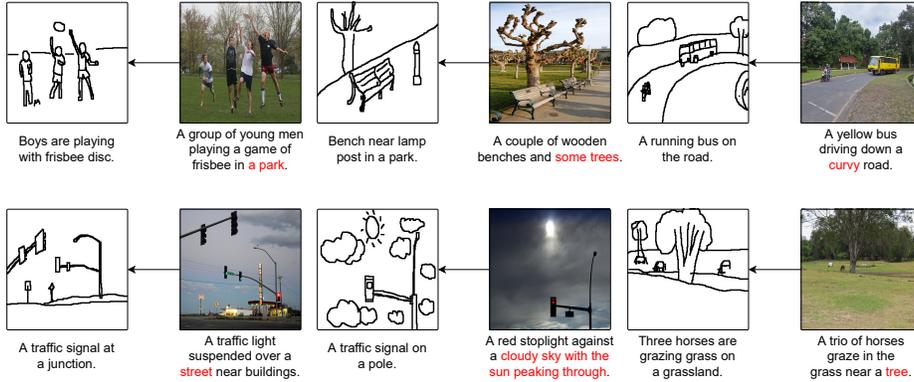


Fig. S5: The Participants in FS-COCO do not exhaustively describe in sketch-captions all the objects present in sketches. The categories that are drawn in sketch but not described in sketch-captions are marked in red.

57.5 for text based image retrieval, it was trained on the entire training set of MS-COCO [32]. This results in an unfair comparison. Hence for a fair evaluation, we use CLIP [45] which in spite of training on a much larger dataset of 400 million text-image pairs, did not include MS-COCO.

Table S5: Fine-grained freehand-scene-sketch-based image retrieval: Additional experiments for Sec. 5.2 in the main document.

Methods	Trained On																	
	SketchyScene (S-Scene) [68]						SketchyCOCO (S-COCO) [20]						FS-COCO (Ours)					
	S-Scene		Evaluate on				S-Scene		Evaluate on				S-Scene		Evaluate on			
R@1	R@10	R@1	R@10	R@1	R@10	R@1	R@10	R@1	R@10	R@1	R@10	R@1	R@10	R@1	R@10	R@1	R@10	
Siam.-SN	2.7	17.3	<0.1	1.1	0.1	3.2	<0.1	<0.1	6.2	32.9	<0.1	<0.1	1.2	9.1	<0.1	3.9	4.7	21.0
Siam.-VGG16	22.8	43.5	1.1	4.1	1.8	6.6	0.3	2.1	37.6	80.6	<0.1	0.4	5.8	24.5	2.4	11.6	23.3	52.6
Heter.-VGG16	15.9	38.4	0.2	3.7	0.8	5.8	0.1	1.6	34.9	76.1	<0.1	0.3	4.2	20.1	1.9	10.7	19.2	47.6
HOLEF-SN [52]	2.9	17.7	<0.1	1.3	0.2	3.2	<0.1	<0.1	6.2	40.7	<0.1	<0.1	1.2	9.3	<0.1	4.1	4.9	21.7
HOLEF-VGG16 [52]	22.6	44.2	1.2	3.9	1.7	5.9	0.4	2.3	38.3	82.5	0.1	0.4	6.0	24.7	2.2	11.9	22.8	53.1
SketchLattice [44]	15.9	37.2	0.1	3.3	0.8	5.6	0.1	1.5	33.7	74.3	<0.1	0.3	3.7	19.4	0.7	9.5	18.9	46.5
Lin <i>et al.</i> [31] (SkBert-VGG16)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	11.3	37.2
SketchyScene [68]	20.6	41.7	0.9	3.9	1.8	6.1	0.2	1.7	36.5	78.6	<0.1	0.4	5.1	24.1	2.4	11.5	23.0	52.3
CLIP (zero-shot) [45]	1.26	9.70	-	-	-	-	-	-	1.85	9.41	-	-	-	-	-	-	1.17	6.07
CLIP-variant	8.6	24.8	1.7	6.6	2.5	8.2	1.3	5.1	15.3	43.9	0.6	3.1	1.6	11.9	2.6	12.5	5.5	26.5

### S5.1 Additional experiments for Sec. 5.3 in the main document: Sketch Captioning

Tab. S6 includes additional experiments for Sec. 5.3 for sketch captioning using existing state-of-the-art methods.

Table S6: Sketch Captioning: Our novel dataset, for the first time, enables captioning of scene sketches. We provide the results of some popular captioning methods originally developed for photos. Empirical results suggests there is significant gap in performance in comparison to image captioning literature. We hope our dataset and quantitative results will inspire future methods to caption scene sketches.

Methods	Belu-1	Belu-2	Belu-3	Belu-4	Meteor	Rouge	CIDEr	Spice
Xu <i>et al.</i> [63]	46.2	29.1	17.8	13.7	17.1	44.9	69.4	14.5
GMM-CVAE [58]	49.6	33.9	18.2	15.5	18.3	48.7	77.6	15.5
AG-CVAE [58]	50.9	34.1	19.2	16.0	18.9	49.1	80.5	15.8
LNFM [35]	52.2	35.7	20.0	16.7	21.0	52.9	90.1	16.0
LNFM (H-Decoder)	<b>54.7</b>	<b>37.3</b>	<b>22.5</b>	<b>17.3</b>	<b>21.1</b>	<b>53.2</b>	<b>95.3</b>	<b>17.2</b>

## S6 User-style adaptation

In this section, we split the dataset differently than in the main paper: we train the models discussed in Sec. 5.1 using sketches from 70 users, and test on the sketches of remaining 30 “unseen” users. Tab. S7 ‘Before Adapt.’ column shows that the performance on sketches of “unseen” users is worse than the one shown in Tab. 3. Hence, it is important to explore techniques that can provide personalization to a new user in a few-shot scenario. Here, we use meta-learning [19, 2] to increase the accuracy of the fine-grained retrieval for a particular subject given just 5 subject-specific sketch examples. We repeat each experiment 5 times with 5 randomly selected sketches each time, and indicate the average performance and the standard deviation among the experiments. Tab. S7 ‘After Adapt.’ column shows that using just 5 subject-specific sketch examples greatly improve scene-level FG-SBIR performance for *Siam.-VGG16* and *HOLEF* models. Tab. S7 shows that such large models as CLIP are less beneficial in the context of personalization.

Table S7: User-style adaptation (Appendix S6). We evaluate generalization of sketch-based fine-grained image retrieval models to “unseen” user styles (Before Adapt.), and the proposed personalization to a user style via meta-learning with just 5 user-scene-sketches (After Adapt.).

Methods	Before Adapt.		After Adapt.	
	R@1	R@10	R@1	R@10
Siam.-VGG16	10.6	32.5	15.5±1.4	37.6±1.9
HOLEF [52]	10.9	33.1	15.5±1.3	38.1±1.5
CLIP* [45]	4.2	22.3	4.2±0.1	22.4±0.1

## S7 H-Decoder: Additional experiments and discussions

### S7.1 H-Decoder implementation details

We use the data format that represents a sketch as a set of pen stroke actions. A sketch is a list of points, and each point is a 5 dimensional vector:  $(x, y, q1, q2, q3)$ . The first two logits  $(x, y)$  represent the absolute coordinate in the  $x$  and  $y$  directions of the pen. The later three  $(q1, q2, q3)$  represent a binary one-hot vector

of 3 possible states: (i) *pen down state*: The first pen state  $q_1$  denotes that the pen is touching the paper. This indicates that a line will be drawn connecting the next point with the current point. (ii) *pen up state*: The second pen state  $q_2$  indicates the pen will be lifted from the paper after the current point to mark the end of a stroke. (iii) *pen end state*: The final pen state  $q_3$  represent that the drawing of scene sketch has ended, and subsequent points will not be rendered.

Our hierarchical decoder consists of two LSTMs: (i) The global LSTM ( $RNN_G$ ) that predicts a sequence of feature vectors, each representing a stroke. (ii) A second local LSTM ( $RNN_L$ ) predicting a sequence of points for any stroke, given its predicted feature vector. The stroke points  $P_t$  are predicted across  $i^{th}$  and  $j^{th}$  steps in  $RNN_G$  and  $RNN_L$  respectively. In more details, let's assume the local  $RNN_L$  predicts  $P_t$  with pen up state  $(0, 1, 0)$  at the  $j^{th}$  unroll step, given input stroke feature  $S_i$ . It will then trigger a single step unroll of the global  $RNN_G$  to predict the next stroke representation  $S_{i+1}$ . This will re-initialise  $RNN_L$  to predict stroke points starting with  $P_{t+1}$  for  $S_{i+1}$  where  $P_t$  is the last predicted point. The unrolling of both  $RNN_L$  and  $RNN_G$  comes to a halt upon predicting  $P_t$  with pen end state  $(0, 0, 1)$ . We define  $P_0$  as  $(0, 0, 1, 0, 0)$ .

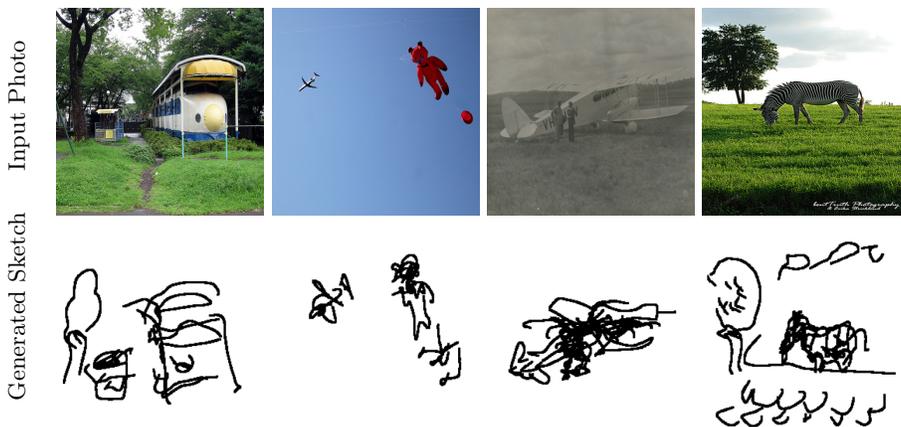


Fig. S6: Photo to vectored sketch synthesis: Our novel dataset allows interesting downstream applications such photo to scene vector sketch synthesis as a byproduct of our hierarchical decoder. Here, we show qualitative results using VGG-16 encoder followed by the hierarchical decoder.

## S7.2 Learning to synthesize human-like sketches

A byproduct of our hierarchical sketch decoder is a naive photo to vector sketch synthesis pipeline. Fig. S6 shows preliminary samples of scene sketches synthesized using our proposed sketch decoder. To improve these results, future work can exploit VAE-based solutions, sequentially generating sketches [24], or parameterized strokes representation [14] to tackle the challenges posed by scene sketches.