Open Vocabulary Semantic Scene Sketch Understanding

Ahmed Bourouis[1] Judith E. Fan[2] Yulia Gryaditskaya[1]

[1]Surrey Institute for People-Centered AI and CVSSP, University of Surrey, UK
[2]Department of Psychology, Stanford University, USA
https://ahmedbourouis.github.io/Scene_Sketch_Segmentation/

Abstract

We study the underexplored but fundamental problem of machine understanding of abstract freehand scene sketches. We introduce a sketch encoder that ensures a semantically-aware feature space, which we evaluate by testing its performance on a semantic sketch segmentation task. To train our model, we rely only on bitmap sketches accompanied by brief captions, avoiding the need for pixel-level annotations. To generalize to a large set of sketches and categories, we build upon a vision transformer encoder pre-trained with the CLIP model. We freeze the text encoder and perform visual-prompt tuning of the visual encoder branch while introducing a set of critical modifications. First, we augment the classical key-query (k-q) self-attention blocks with value-value (v-v) self-attention blocks. Central to our model is a two-level hierarchical training that enables efficient semantic disentanglement: The first level ensures holistic scene sketch encoding, and the second level focuses on individual categories. In the second level of the hierarchy, we introduce cross-attention between the text and vision branches. Our method outperforms zero-shot CLIP segmentation results by 37 points, reaching a pixel accuracy of 85.5% *on the FS-COCO sketch dataset. Finally, we conduct a user study that allows us to identify further improvements needed over our method to reconcile machine and human understanding of freehand scene sketches.*

1. Introduction

Even a quick sketch can convey rich information about what is relevant in a visual scene: what objects there are and how they are arranged. However, little work has been devoted to the task of machine scene sketch understanding, largely due to a lack of data. Understanding sketches with methods designed for images is challenging because sketches have very different statistics from images – they are sparser and lack detailed color and texture information. Moreover, sketches contain abstraction at multiple levels: the holistic scene level and the object level. Here we explore the promise of two main ideas: (1) the use of language to guide the learning of how to parse scene sketches and (2) a two-level training network design for holistic scene understanding and individual categories recognition.

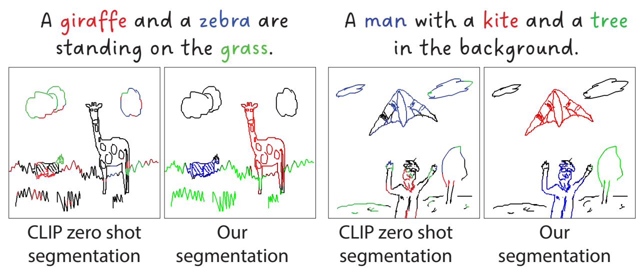


Figure 1. Comparison of the segmentation result obtained with CLIP visual encoder features and features from our model.

Freehand sketches can be represented as a sequence or cloud of individual strokes, or as a bitmap image. As one of the first works on scene sketch understanding, we target a general setting where we assume only the availability of bitmap representations. We also aim at the method that can generalize to a large number of scenes and object categories. To this end, we build our sketch encoder on a Visual Transformer (ViT) encoder pre-trained with a popular CLIP [44] foundation model (Fig. 1). We propose a two-level hierarchical training of our network, where the two levels ("Holistic" and "Category-level") share the weights of our visual encoder. The first level focuses on ensuring that our model can capture holistic scene understanding (Fig. 2: I. Holistic), while the second level ensures that the encoder can efficiently encode and distinguish individual categories (Fig. 2: II. Category-level). We avoid reliance on tedious user per-pixel annotations by leveraging sketch-caption pairs from the FS-COCO dataset [9], and aligning the visual tokens of sketch patches with textual tokens from the sketch captions, using triplet loss training. We strengthen the alignment by introducing sketch-text cross-attention in the second level of the network's hierarchy (Fig. 2: g.). Additionally, we introduce a modified self-attention computation to the visual transformer encoder used in both layers, inspired by recent work by Li et al. [33].

We conduct a comprehensive evaluation of our method comparing it with recent language-supervised image segmentation methods [33, 44, 58], fine-tuned on the FS-COCO dataset. We show that our approach outperforms with a large margin all existing methods on the task of freehand sketch segmentation. We also compare with a previous fully supervised work on scene sketch segmentation [19], trained on a semi-synthetic set of sketches composed of individual category sketches. We demonstrate that their work does not generalize well to freehand scene sketches [9]. Our method demonstrates consistent performance and similarly outperforms [19] on a dataset of freehand sketches provided by Ge *et al*. [19].

Finally, our analysis reveals that although our model consistently produces robust segmentation results across the majority of sketches, there are a few challenging sketching scenarios for our method. We select a subset of representative sketches for each scenario and collect multi-user annotations. We then carefully assess our approach by comparing its performance with that of human participants, drawing insights to guide future work.

In summary, our contributions include: (1) a two-level hierarchical training approach, focusing on holistic scene sketch understanding and category disentanglement, (2) the first language-supervised scene sketch segmentation method, (3) per pixel segmentation annotations of 975 sketches from the FS-COCO dataset, and (4) multi-user annotations of a subset of distinct groups of sketches.

2. Related Work

2.1. Unsupervised and Weekly Supervised Image Semantic Segmentation

The need for pixel-wise segmentation limits the number of instances that supervised segmentation models [1, 6, 7, 17, 36, 64] can use for training, as such annotations are costly to collect. This in turn limits the generalization properties of models trained with pixel-level annotations. To avoid the need for extensive annotations, unsupervised [8, 23, 26, 39, 62], semi-supervised [40, 72] and weakly supervised [13, 14, 24, 37, 38, 41, 56, 58, 69] methods were proposed.

Our method belongs to the group of weakly supervised methods based on text annotations only [5, 13, 14, 37, 38, 58], such methods are not limited to a fixed set of categories and therefore are referred to as open vocabulary semantic segmentation methods. Image methods typically rely on the spatial proximity of semantically similar pixels. This is less applicable in the sparse and largely monochromatic landscape of freehand sketches. For example, recent GroupViT [58] and SegCLIP [38] use learnable group tokens and semantic group modules to aggregate low-layer pixel features. In our work, we propose a two-level training architecture taking sketch sparsity and abstraction into account.

2.2. Sketch Semantic Segmentation

The majority of works on semantic sketch segmentation focus on single-category sketches. Some of these works treat sketch as a bitmap image [32, 70, 71], but most leverage stroke-level information directly [12, 21, 22, 28, 42, 43, 48, 55, 57, 60, 66] or as a segmentation refinement step [32, 71]. All these works are fully supervised except for [42], which segments sketches of a given category provided at least one segmented reference sketch.

Semantic scene sketch segmentation [51], and more broadly scene sketch understanding, is underexplored, to a large extent due to a lack of data. The lack of data is typically addressed by introducing semi-synthetic sketch datasets. The SketchyScene dataset [73] consists of 7,264 sketch-image pairs, obtained by arranging clip-art individual category sketches in alignment with a reference image. SketchyCOCO dataset [18] is generated from COCO-Stuff [4] by semi-automatically arranging freehand sketches of individual categories. Ge *et al*. [19] introduced their own semi-synthetic scene sketch dataset and adopted a DeepLab-v2 [6] architecture to the scene sketch segmentation task. SketchSeger [59] proposed an encoder-decoder model based on hierarchical Transformers, trained with a stroke-based cross-entropy loss on semi-synthetic scene sketches formed by combining sketches from the QuickDraw dataset [21]. Zhang *et al*. [63] proposed an RNN-GCN-based architecture trained on annotated freehand scene sketches. However, neither the dataset nor the code have been released. We do not require stroke-level information or pixel-wise segmentation of the training data, and leverage the FS-COCO dataset [9] of freehand sketches with their textual descriptions.

2.3. ViT-CLIP and Sketch

We build our encoder on a ViT (Vision Transformer) encoder pre-trained with CLIP (Contrastive Language-Image Pre-training) [44]. CLIP is a model trained on roughly 400 million image-text pairs to embed images and text in a shared space. It uses ViT as a visual branch (image) encoder. A ViT encoder pre-trained with CLIP (ViT-CLIP) is used in a range of sketch-related tasks: sketch and drawing generation [16, 49, 53, 54], 2D image retrieval [9, 46, 47], object detection [10], 3D shape retrieval [2, 30, 31, 50, 61], 3D shape generation [65].

While some works use ViT-CLIP purely pre-trained on images, many fine-tune the encoder on sketches for downstream tasks. Some works fine-tune all weights of the encoder [2, 47], some fine-tune Layer Normalization layers only [9], and some rely on prompt-learning [27, 68] or the combination of the latter two [10, 46]. In our work, we also

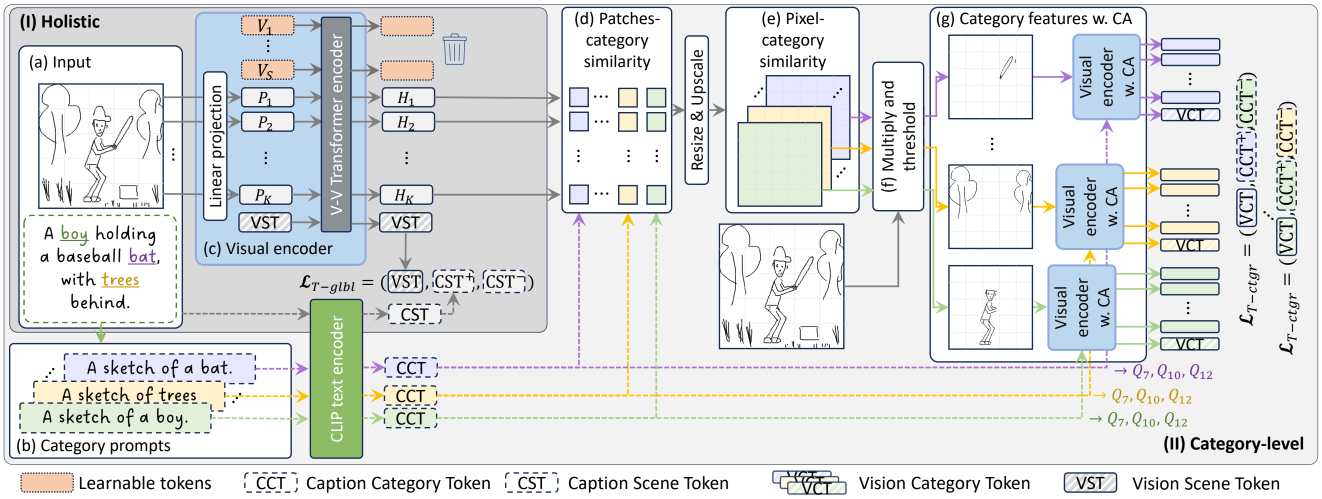


Figure 2. Our framework consists of two levels: I. Holistic Scene Sketch Understanding and II. Targeting individual categories disentanglement. Please refer to Sec. 3 for details.

rely on fine-tuning with visual prompt learning and Layer Normalization layers updates. Unlike previous methods targeting sketch inputs, we additionally leverage a two-path ViT architecture, inspired by Li *et al*. [33].

3. Method

As we mention in the introduction, we build a sketch encoder such that the semantic meaning of individual stroke pixels can be inferred from its feature embeddings. Building on the ViT encoder, pre-trained CLIP [44] model, we fine-tune a modified encoder architecture with a network consisting of two levels: Holistic scene understanding and individual category recognition. We start by describing the first level of our network (Fig. 2 I.) and introducing the architecture of our visual encoder (Fig. 2 c.). We then describe our strategy to improve the model's ability to understand individual categories (Fig. 2 II.).

3.1. Holistic Scene Sketch Understanding

The architecture in the first level (Fig. 2: I. Holistic) is similar to the architecture of the CLIP model [33]. We freeze the weights of the textual encoder and fine-tune the modified architecture of the vision encoder (Sec. 3.1.1). The CLIP model is trained with a contrastive loss, ensuring that the embedding of images and corresponding captions are closer in space than embeddings of images and captions of other images. While our training has a similar goal, we train with a triplet loss with hard triplet mining, as we found it to be more beneficial with the batch size we use:

$$\mathcal{L}_{N_T\text{-}glbl} = \frac{1}{N_T}\sum_{i=1}^{N_T}\max\{||\texttt{VST}_i - \texttt{CST}_i^+|| - ||\texttt{VST}_i - \texttt{CST}_j^-|| + m, \quad 0\}. \tag{1}$$

Here, a holistic visual scene sketch embedding VST (Visual Scene Token) serves as an anchor. An encoding of the matching sketch caption $\texttt{CST}^+$ (Caption Scene Token) serves as a positive sample, and an encoding of the most dissimilar scene caption serves as a negative sample $\texttt{CST}^-$. We set the margin m to a commonly used value of 0.3. The number of triplets N_T is equal to the number of samples in a batch.

3.1.1 Visual encoder

The input scene sketch is divided into non-overlapping patches, which are flattened and linearly projected into the feature space. Concatenating with positional encodings, we obtain one token $P_k \in \mathbb{R}^{1\times d}$ per patch. Additionally, we add a set of learnable tokens, V_s, referred to as *visual prompts* [11]. Finally, these tokens are also augmented with a special token that encodes holistic sketch meaning, VST (Visual Scene Token). Note that in the context of classification, a CLS token has a similar role to our VST token. Therefore, the input to the vision encoder is $X = [\texttt{VST}, P_1, ..., P_K, V_1, ..., V_S] \in \mathbb{R}^{N_X\times d}$, where $N_X = 1 + K + S$.

Attention computation It was observed by Li et al. [33] that CLIP-predicted similarity maps between image and text features emphasize background regions rather than areas that correspond to a category in the text embedding. To address this issue, they proposed to use an instance of self-self attention called v-v attention, which does not require training or fine-tuning the original model. Li *et al*. [33], and later Bousselham *et al*. [3] demonstrated that this leads to improved performance in open vocabulary segmentation tasks: Self-self-attention reinforces the similarity of tokens

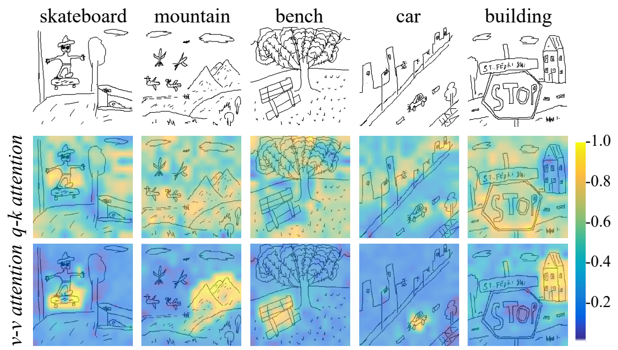


Figure 3. Comparison of similarity maps obtained with classical attention computation (*q-k attention*) in the second row, with the ones obtained from *v-v attention*, given by Eq. (2).

already close to each other (*e.g*. representing the same object), which leads to a clearer separation in the feature space, thereby improving the segmentation quality.

We performed a similar experiment with CLIP features for sketch inputs: The similarity maps in the second row of Fig. 3 show the poor ability of CLIP features to identify target categories. Therefore, we follow [33] and use their two-path configuration of the vision transformer. However, we use it not only for inference but also incorporate this two-path configuration directly into our network training, as we find it more beneficial. We provide a detailed analysis in Sec. 4.5.1.

The first path represents the original vision encoder where identical blocks are repeated L times. Each block consists of *Layer Normalization (LN)*, followed by *Multi-Head Self Attention (MHSA)*, another *LN* and *Fead Forward Network (FFD)*.

The second path blocks contain a modified attention computation in *MHSA*, dubbed as *v-v self-attention*, where *Keys* and *Queries* are ignored, and self-attention is computed using only *Values,* $V \in \mathbb{R}^{N_X \times d}$:

$$\texttt{s-attn}(V, V, V) = \texttt{softmax}\left(VV^T/\sqrt{d}\right)V. \quad (2)$$

In addition, blocks in the second path do not include the second *LN* and *FFN* layers. Finally, in the second path, the input to the *v-v multi-head attention* is always the features from the original path. We use the output from the second path during training and inference.

As shown in Fig. 3 third row, the v-v attention results in feature representations that accurately represent distinct semantic entities present in the scene sketch.

3.2. Categories Disentanglement

Given the sketch caption we automatically identify individual categories and generate a set of textual prompts of the form *"A sketch of *"* (Fig. 2b.). Each of these textual category prompts is encoded with the CLIP text encoder into $\texttt{CCT} \in \mathbb{R}^{1\times d}$ (Caption Category Token).

We then compute the per-patch cosine similarity M_k^c between the class embeddings CCT and the scene sketch patch embeddings H_k, defined as:

$$M_k^c = \frac{\texttt{CCT}^c \cdot H_k^T}{|\texttt{CCT}^c||H_k^T|}, \quad (3)$$

where $k \in [1, K]$ is the patch index and $c \in [1, N_c]$ is an index of a category (*e.g. trees*). The resulting similarity matrix $M^c \in \mathbb{R}^{K \times N_c}$ represents the category label probabilities for each individual patch (Fig. 2d.). To generate a pixel-level similarity map, we reshape each M^c and then upscale to the dimensions of the original scene sketch using bi-cubic interpolation [52]. By multiplying these per category maps with the input scene sketch, as shown in Fig. 2e., we obtain disentanglement into individual sketch categories.

Thresholding with a learnable parameter Only pixels with similarity scores above a certain threshold are retained at this step (Fig. 2f.). We make the threshold learnable, eliminating the need for manual tuning. More importantly, the threshold value increases over epochs as the model becomes more confident in its predictions, allowing the model to obtain strong disentanglement performance.

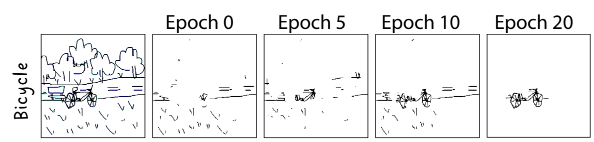


Figure 4. Visualization of disentanglement over epochs.

Visual encoder with Cross-Attention The features of individual category sketches are extracted with the visual encoder identical to the one used in the holistic scene sketch level understanding of our network, described in Sec. 3.1.1, up to one difference.

We enhance the interplay between textual and visual domains through the introduction of cross-attention. Namely, in 7th, 10th, and 12th layers in the *MHSA*, we feed CCT token from the textual encoder representing a target category to the linear projection for the queries. This enables the model to leverage category token embedding from the textual domain to update the sketch token embedding. This results in a better text-to-sketch alignment for individual categories and subsequently improves sketch semantic segmentation. Our ablation study in Tab. 4 underscores the efficacy of this cross-attention strategy.

Text-sketch category-level alignment We train with a triplet loss, $\mathcal{L}_{T\_ctgr}$, so that the category sketch embedding, VCT (Vision Category Token), is used as an anchor, the matching embedding of the category prompt is used as a positive sample and the embedding of the prompt of the most dissimilar category is used as negative. We use the VCT from multiple encoder layers: l_7, l_{10}, l_{12}.

3.3. Efficient CLIP fine-tuning

The two levels (holistic and category) are trained jointly, using the total loss

$$\mathcal{L} = \mathcal{L}_{T\_glbl} + \mathcal{L}_{T\_ctgr}. \tag{4}$$

We leverage the generalization properties of the pre-trained foundation model through careful fine-tuning. We freeze all the weights apart from weights of *LN*, as was proposed in [15], and we use learnable visual prompts, as was proposed in [27]. We introduced visual prompts in Sec. 3.1.1. We also train linear layers which take part in cross-attention computation.

3.4. Inference

Our network design allows segmentation for different sets of categories. Given a desirable set of categories for a given sketch, we obtain sketch segmentation by applying all the steps of our network up to the calculation of pixel-category similarities (Fig. 2e.), followed by upscaling of similarity maps for each category, as discussed in Sec. 3.2. To assign segmentation results we assign to each pixel a label that yields the highest similarity value across category similarity maps M_i^c, where i is an index of a category.

If we want to isolate just a few categories in the sketch, we can use the thresholding strategy that we use during training to isolate the pixels of a given category (Fig. 2f.). We used this strategy to obtain visualizations in Fig. 1, with a threshold value of 0.71 that we found to be optimal on the test set of sketches. We do not use the learned value from the training, as during training the model does not have to select all the pixels of the given category, but only those that are sufficient to confidently predict the category label. We provide an in-depth discussion in the supplemental.

4. Experiments

4.1. Training and Test Data

For training and testing, we use the sketch-caption pairs from the FS-COCO [9] dataset. The dataset comprises 10,000 sketch-caption pairs, associated with reference images from the MS-COCO [34] dataset. The sketches are drawn from memory by 100 non-expert participants. The reference image was shown for 60 seconds, followed by a 3-minute sketching window.

Training/Validation/Test splits We first selected 500 sketches with distinct styles from five participants. We then randomly sample 5 sketches from each of the remaining 95 participants for validation (a total of 475 sketches). We use the remaining 9025 sketches for training.

Annotations One of the co-authors manually annotated test and validation sketches, relying on reference images and category labels from the MS-COCO [34] dataset. We assign each stroke a unique category label. Candidate category labels are extracted from MS-COCO image captions rather than sketch captions to obtain richer *'ground-truth'* annotations. Our test set contains 185 different object classes, with an average of 3.54 objects per sketch.

4.2. Evaluation Metrics

We use standard metrics, commonly used in sketch segmentation literature [25, 57, 63].

Mean Intersection over Union ($mIoU$): evaluates the average of the ratios between the intersection and the union of ground truth and predicted labels over all categories.

Pixel Accuracy ($Acc@P$): measures the ratio of correctly labeled pixels to the total pixel count in a sketch.

Stroke Accuracy ($Acc@S$): evaluates the percentage of correctly classified strokes to total strokes per sketch. A stroke label is determined by its most frequent pixel label.

4.3. Implementation Details

We implemented our method in PyTorch and trained on two 24GB Nvidia RTX A5000 GPUs. We built on CLIP [44] with a ViT backbone using ViT-B/16 weights. The input sketch image size is set as 224×224. We use 3 learnable visual prompts. We use AdamW optimizer with a learning rate of 10^{-6}, and train the model for 20 epochs with a batch size of 16. We pick a checkpoint based on the *mIoU* performance on the validation set. We provide more discussion on the checkpoint choice in the supplemental.

4.4. Comparison against state-of-the-art

4.4.1 Comparison with fully-supervised methods

We first compare with several recent methods for image segmentation that similarly to us utilize either CLIP as a backbone: *DenseCLIP* [45] and *ZegCLIP* [69], or more recent foundational backbones Grounding-DINO [35] and SAM [29], used in *Grounded-SAM* [20]. These methods require pixel-level annotated examples, and therefore can not be fine-tuned on our training data. We also compare to a recent fully supervised method *LDP* (Local Detail Perception) [19] for scene sketch semantic segmentation, which is trained on a dataset of semi-synthetic sketches. Such sketches are obtained as a superposition of freehand category-level sketches. Tab. 1 shows that neither of the these methods generalizes well to freehand scene sketches.

4.4.2 Comparison with language-supervised methods

Next, we compare with several recent methods targeting semantic segmentation with ViT encoders and image-text supervision: *GroupViT* [58] and *SegCLIP* [38]. Additionally,

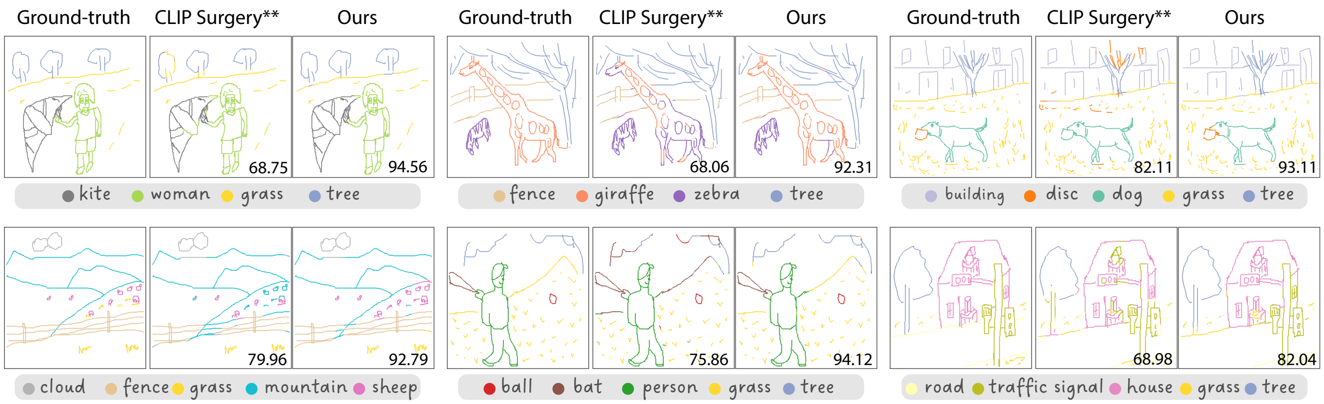


Figure 5. Visual comparison of our method with *CLIP Surgery***. *CLIP Surgery*** represents the fine-tuned ViT from the CLIP model with v-v self-attention introduced at both training and inference stages. The numbers show $Acc@P$ values.

Methods	$mIoU$	$Acc@P$	$Acc@S$
ZegCLIP [69]	15.45	32.48	35.21
DenseCLIP [67]	28.22	50.62	50.25
Grounded-SAM [20]	32.21	50.12	50.02
LDP [19]	33.04	56.23	56.71
Ours	**73.48**	**85.54**	**87.02**

Table 1. Comparison of our method against state-of-the-art fully supervised sketch method and image segmentation methods, relying on the availability of pixel-level annotations, on our test set of freehand sketches from the FS-COCO dataset.

	Methods	$mIoU$	$Acc@P$	$Acc@S$
Zero-shot	CLIP [44]	17.33	28.82	27.15
	GroupViT [58]	38.25	61.39	60.07
	SegCLIP [38]	38.14	61.45	65.56
	CLIP_Surgery [33]	52.63	72.47	75.17
Fine-tuned	CLIP*	22.86	33.41	32.64
	GroupViT*	45.71	66.21	66.89
	SegCLIP*	49.26	69.87	73.64
	CLIP_Surgery*	48.74	65.38	68.78
	CLIP_Surgery**	59.98	78.68	81.11
	Ours	**73.48**	**85.54**	**87.02**

Table 2. Comparison of our method against state-of-the-art language supervised image segmentation methods on our test set of sketches from the FS-COCO dataset. The fine-tuned methods on our training set of freehand sketches are marked with stars. *CLIP Surgery** represents the fine-tuned CLIP model with v-v self-attention introduced only at inference stages. *CLIP Surgery*** represents the fine-tuned model with v-v self-attention introduced at both training and inference stages.

we compare with CLIP [44], as well as CLIP Surgery [33] that introduced the usage of *v-v-attention* at inference time.

Zero-shot In Tab. 2, we first compare the performance of our method with the zero-shot performance of these methods. It shows that image segmentation methods do not generalize well to freehand sketches.

Fine-tuning We fine-tune each of the methods on our training set, by updating all their weights. Since such fine-tuning might be sensitive to a learning-rate choice, we perform several runs with several settings of learning rate parameters. We chose the setting for each method that results in the best performance on our validation set. The fine-tuned methods are marked with stars.

Tab. 2 shows that our method outperforms all considered baselines, and surpasses the best-performing baseline *CLIP Surgery*** by a substantial margin of 13.5, 9.9 and 5.9 points in $mIoU$ score, $Acc@P$ and $Acc@S$, respectively. In Sec. 4.5.1, we evaluate various elements of our architecture and their contribution to overall performance.

Fig. 5 shows the qualitative comparison between our method and the *CLIP Surgery***. We provide additional visual comparisons in the supplemental.

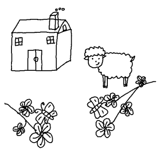
4.4.3 Generalization ability of our method

Next, we evaluate our method on an additional dataset of 50 freehand sketches provided and annotated by Ge *et al*. [19]. Tab. 3 shows that our model again demonstrates superior performance on this dataset over the method [19], fully supervised on semi-synthetic sketches. We do not compute $Acc@S$ as sketches are only available as bitmap images. This experiment highlights that short language captions can be efficiently used for training, eliminating the need for expensive and time-consuming per-pixel annotations.

Method	$mIoU$	$Acc@P$
LDP [19]	37.16	78.84
Ours	**53.94**	**81.63**

Table 3. Comparison on the freehand sketches from [19].

The lower *mIoU* values on these sketches than on FS-COCO sketches can be explained by (1) on larger average number of categories in them (5.74 categories per sketch) than in our FS-COCO test set (3.54 categories per sketch); (2) domain gap. The sketches from [19] contain symbolic representations of objects (see the inset on the left) and look more like a superposition of sketches that can be found in the *QuickDraw* [21] dataset rather than holistic scene sketches. We analyze challenging scenarios for our method in Sec. 5.1.



4.5. Ablation Study

4.5.1 Importance of individual components

We perform an ablation analysis to assess the importance of each component in our architecture. Tab. 4 shows the performance of the complete model with individual elements removed. We discuss them in order of impact on overall performance.

v-v attention First, we show the importance of the v-v attention, by substituting our dual path v-v attention-based ViT encoder with the original configuration used in the CLIP model (**w/o v-v attention**).

Two-level network architecture We keep only the first level of holistic scene understanding of the network (Fig. 2 I.). This architecture is similar to *CLIP Surgery***, but is supervised with the triplet loss and is fine-tuned using *learnable visual prompts* and updates only *LN* layers. Tab. 4 *(w/o category-level)* confirms that two-level network architecture, along *v-v attention*, is central to the superiority of our model.

Thresholding We perform an experiment where instead of thresholding we weight each pixel according to cosine similarity scores in M^c maps (Tab. 4 *(w/o thresholding)*). The learnable threshold more efficiently filters out irrelevant pixels, forcing the model to learn superior disentanglement of individual categories.

Holistic scene encoding Removing the global loss, given by Eq. (1), similarly results in the performance drop *(w/o Global Loss)*. This shows the mutual importance of the two levels of our network.

Cross-Attention Cross attention also substantially contributes to performance. If we use a ViT encoder at the second level of the network (category level), identical to the one used at the first level (holistic level) (Fig. 2c.), then the performance drops by a noticeable 3.35 points in the *mIoU* score (Tab. 4 *(w/o cross-attention)*).

Multi-layer features in the triplet loss Tab. 4 *(w/o cross-attention)* shows that using features from multiple layers (l_7, l_{10}, l_{12}) in the category-level triplet loss is beneficial over using only the features from the last layer (l_{12}).

Model	$mIoU$	$Acc@P$	$Acc@S$
w/o v-v attention	43.55	58.09	59.03
w/o category-level	65.03	79.35	81.82
w/o thresholding	66.93	81.06	82.56
w/o global Loss	69.06	81.35	83.68
w/o cross-attention	70.13	82.86	85.26
w/o multi-layer Loss	71.29	83.04	86.13
Ours-full	**73.48**	**85.54**	**87.02**

Table 4. Ablation of the role of individual components of our model. See Sec. 4.5.1 for details.

4.5.2 Efficient fine-tuning

Fig. 6 shows the comparison of different fine-tuning strategies. We obtain the best results by combining fine-tuning of *LN* (Layer Normalization) layers and the addition of 3 learnable tokens. Adding more or less tokens degrades the performance Fig. 6b.

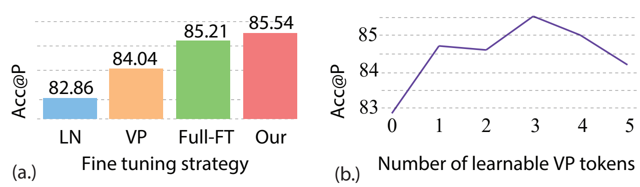


Figure 6. Evaluation of alternative fine-tuning strategies (a.) and the impact of the number of learnable tokens on segmentation accuracy (b.). *LN* means that only *LN* layers are fine-tuned; *VP* means that only learnable Visual Prompt tokens are used; *Full-FT* means that all weights of ViT are fine-tuned.

5. Human-Model Alignment

Fig. 7 shows that for the majority of sketches in our test set from the FS-COCO dataset, our model correctly labels more than 80% pixels.

In this section, we investigate (1) which sketches are likely to get low segmentation accuracy and (2) how the prediction of our model compares with human observers across different groups of sketches.

5.1. Sketch Groups

We identified four distinct sketch groups that are challenging for our model: (1) **Ambiguous sketches**: sketches where it might be hard even for a human observer to understand an input sketch; (2) **Interchangeable categories**: sketches containing multiple objects with labels that can interchange each other, like *'tower'* and *'building'*, or *'girl'* and *'man'*; (3) **Correlated categories**: sketches with

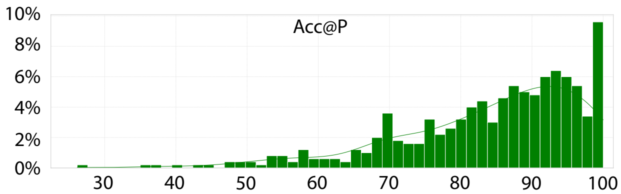


Figure 7. Histogram of Acc@P values for our method on 500 sketches from our FS-COCO test set.

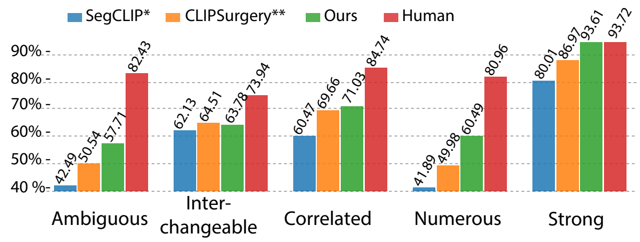


Figure 8. Comparison of the percentage of correctly predicted pixels (*Acc@P*) by different models and human observers across five distinct sketch categories, introduced in Sec. 5.1.

categories that typically co-occur in scenes, *e.g.*, *'train'*-*'railway'* and *'airplane'*-*'runway'*; and (4) **Numerous-categories**: sketches with six or more categories.

We supplement these four groups with sketches where our model labels correctly more than 80% of pixels: (5) **Strong performance**.

5.2. User Study Setting

Data We sample 5 sketches for each of the first 4 categories and 10 sketches for the 5th category. We visualize selected sketches in the supplemental material.

Participants We recruited 25 participants (14 male). Each participant was randomly assigned 6 sketches: 1 from each of the first 4 groups and 2 from the 5th group, such that every sketch was annotated by five unique participants.

Study Procedure Participants were presented with one sketch and one object category at a time and were not able to see their previous annotations. Sketch-category pairs were interlaced, to reduce the effect of memorizing their previous annotation on a certain sketch. The annotation interface enabled precise pixel-level segmentation by allowing participants to "paint" over each sketch using a brush with an adjustable radius. Participants could also use the eraser to correct erroneous annotations. Once a participant has moved to a new sketch-category pair, they were not able to change their previous annotations.

5.3. User Study Analysis and Future Work

***'Human'* segmentation** For each sketch, we generate one *'human'* segmentation using a majority vote. For each pixel and each label, we computed the percentage of annotators that assigned a given label. We then assigned to each pixel the label that was provided most frequently to that pixel by different annotators. In cases where there were multiple labels were provided equally often for a pixel, we randomly sampled one of these labels.

Analysis First, we observed that on sketches that did not fall into any of the challenging categories, our model almost reaches human-level performance, with a negligible gap of 0.11 points on average (Fig. 8 Strong).

Fig. 8 Ambiguous shows that, given a label, humans can correctly identify sketch pixels even in the presence of ambiguity. While none of the models currently match human performance on *ambiguous sketches*, our model surpasses the other methods by a noticeable margin, demonstrating the effectiveness of our two-level training architecture.

The performance across *semantically interchangeable categories* is uniform amongst the three language-supervised models. This potentiality can be alleviated by proposing solutions that assign labels jointly.

On sketches with *correlated categories* our model and ClipSurgery** perform similarly, highlighting the inherent limitation of training using language supervision. For a few such categories, one might need to further fine-tune the model relying on sketches of isolated categories.

Our model represents a substantial improvement over current alternatives, surpassing them by more than 10 points. Future work should seek to improve alignment with human sketch understanding, especially on sketches with more than six categories (Fig. 8 Numerous).

6. Conclusion

While focusing on the task of sketch segmentation, we introduced a strategy to train a ViT encoder that results in the feature space with good semantic disentanglement. Such feature spaces contribute towards improving machine understanding of abstract freehand sketches and underpin a range of downstream tasks such as communication and creative pipelines. In light of the latter, it can enable more potent tools for conditional generation and retrieval. In psychology, sketches are used to analyze cognitive functions. This can be facilitated by the availability of robust sketch understanding tools. Importantly, we for the first time demonstrated how language supervision can be used for the task of scene sketch segmentation. Finally, we conducted a comprehensive analysis of our model's performance, identifying research directions to further align the understanding of sketches by humans and machines.

References

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017. 2

[2] Gianluca Berardi and Yulia Gryaditskaya. Fine-tuned but zero-shot 3d shape sketch view similarity and retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1775–1785, 2023. 2

[3] Walid Bousselham, Felix Petersen, Vittorio Ferrari, and Hilde Kuehne. Grounding everything: Emerging localization properties in vision-language transformers. *arXiv preprint arXiv:2312.00878*, 2023. 3

[4] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1209–1218, 2018. 2

[5] Jun Chen, Deyao Zhu, Guocheng Qian, Bernard Ghanem, Zhicheng Yan, Chenchen Zhu, Fanyi Xiao, Mohamed Elhoseiny, and Sean Chang Culatana. Exploring open-vocabulary semantic segmentation without human labels. *arXiv preprint arXiv:2306.00450*, 2023. 2

[6] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017. 2

[7] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 2

[8] Jang Hyun Cho, Utkarsh Mall, Kavita Bala, and Bharath Hariharan. Picie: Unsupervised semantic segmentation using invariance and equivariance in clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16794–16804, 2021. 2

[9] Pinaki Nath Chowdhury, Aneeshan Sain, Ayan Kumar Bhunia, Tao Xiang, Yulia Gryaditskaya, and Yi-Zhe Song. Fscoco: towards understanding of freehand sketches of common objects in context. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VIII*. Springer, 2022. 1, 2, 5

[10] Pinaki Nath Chowdhury, Ayan Kumar Bhunia, Aneeshan Sain, Subhadeep Koley, Tao Xiang, and Yi-Zhe Song. What can human sketches do for object detection? In *CVPR*, 2023. 2

[11] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. *arXiv preprint arXiv:2309.16588*, 2023. 3

[12] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29, 2016. 2

[13] Jian Ding, Nan Xue, Gui-Song Xia, and Dengxin Dai. Decoupling zero-shot semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11583–11592, 2022. 2

[14] Xiaoyi Dong, Jianmin Bao, Yinglin Zheng, Ting Zhang, Dongdong Chen, Hao Yang, Ming Zeng, Weiming Zhang, Lu Yuan, Dong Chen, et al. Maskclip: Masked self-distillation advances contrastive language-image pretraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10995–11005, 2023. 2

[15] Jonathan Frankle, David J Schwab, and Ari S Morcos. Training batchnorm and only batchnorm: On the expressive power of random features in cnns. *arXiv preprint arXiv:2003.00152*, 2020. 5

[16] Kevin Frans, Lisa Soros, and Olaf Witkowski. Clipdraw: Exploring text-to-drawing synthesis through language-image encoders. *Advances in Neural Information Processing Systems*, 35:5207–5218, 2022. 2

[17] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3146–3154, 2019. 2

[18] 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*, pages 5174–5183, 2020. 2

[19] Ce Ge, Haifeng Sun, Yi-Zhe Song, Zhanyu Ma, and Jianxin Liao. Exploring local detail perception for scene sketch semantic segmentation. *IEEE Transactions on Image Processing*, 31, 2022. 2, 5, 6, 7

[20] GroundedSAM. Grounded-Segment-Anything. https://github.com/IDEA-Research/Grounded-Segment-Anything, 2023. 5, 6

[21] David Ha and Douglas Eck. A neural representation of sketch drawings. *arXiv preprint arXiv:1704.03477*, 2017. 2, 7

[22] F Hähnlein, Y Gryaditskaya, and A Bousseau. Bitmap or vector? a study on sketch representations for deep stroke segmentation. In *Journées Francaises d'Informatique Graphique et de Réalité virtuelle*, 2019. 2

[23] Mark Hamilton, Zhoutong Zhang, Bharath Hariharan, Noah Snavely, and William T Freeman. Unsupervised semantic segmentation by distilling feature correspondences. *arXiv preprint arXiv:2203.08414*, 2022. 2

[24] Wenbin He, Suphanut Jamonnak, Liang Gou, and Liu Ren. Clip-s^4: Language-guided self-supervised semantic segmentation, 2023. 2

[25] Zhe Huang, Hongbo Fu, and Rynson WH Lau. Data-driven segmentation and labeling of freehand sketches. *ACM Transactions on Graphics (TOG)*, 33(6):1–10, 2014. 5

[26] Jyh-Jing Hwang, Stella X Yu, Jianbo Shi, Maxwell D Collins, Tien-Ju Yang, Xiao Zhang, and Liang-Chieh Chen. Segsort: Segmentation by discriminative sorting of segments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7334–7344, 2019. 2

[27] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *Computer Vision–ECCV 2022: 17th*

European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIII*, pages 709–727. Springer, 2022. 2, 5

[28] Kurmanbek Kaiyrbekov and Metin Sezgin. Deep stroke-based sketched symbol reconstruction and segmentation. *IEEE computer graphics and applications*, 40(1):112–126, 2019. 2

[29] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023. 5

[30] Trung-Nghia Le, Tam V Nguyen, Minh-Quan Le, Trong-Thuan Nguyen, Viet-Tham Huynh, Trong-Le Do, Khanh-Duy Le, Mai-Khiem Tran, Nhat Hoang-Xuan, Thang-Long Nguyen-Ho, et al. Sketchanimar: Sketch-based 3d animal fine-grained retrieval. *arXiv preprint arXiv:2304.05731*, 2023. 2

[31] Hyundo Lee, Inwoo Hwang, Hyunsung Go, Won-Seok Choi, Kibeom Kim, and Byoung-Tak Zhang. Learning geometry-aware representations by sketching. *arXiv preprint arXiv:2304.08204*, 2023. 2

[32] Lei Li, Hongbo Fu, and Chiew-Lan Tai. Fast sketch segmentation and labeling with deep learning. *IEEE computer graphics and applications*, 39(2):38–51, 2018. 2

[33] Yi Li, Hualiang Wang, Yiqun Duan, and Xiaomeng Li. Clip surgery for better explainability with enhancement in open-vocabulary tasks, 2023. 2, 3, 4, 6

[34] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 5

[35] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023. 5

[36] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 2

[37] Timo Lüddecke and Alexander Ecker. Image segmentation using text and image prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7086–7096, 2022. 2

[38] Huaishao Luo, Junwei Bao, Youzheng Wu, Xiaodong He, and Tianrui Li. Segclip: Patch aggregation with learnable centers for open-vocabulary semantic segmentation. *arXiv e-prints*, pages arXiv–2211, 2022. 2, 5, 6

[39] Luke Melas-Kyriazi, Christian Rupprecht, Iro Laina, and Andrea Vedaldi. Deep spectral methods: A surprisingly strong baseline for unsupervised semantic segmentation and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8364–8375, 2022. 2

[40] Sudhanshu Mittal, Maxim Tatarchenko, and Thomas Brox. Semi-supervised semantic segmentation with high-and low-level consistency. *IEEE transactions on pattern analysis and machine intelligence*, 43(4):1369–1379, 2019. 2

[41] Deepak Pathak, Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional multi-class multiple instance learning. In *ICLR Workshop*, 2015. 2

[42] Anran Qi, Yulia Gryaditskaya, Tao Xiang, and Yi-Zhe Song. One sketch for all: One-shot personalized sketch segmentation. *IEEE transactions on image processing*, 31:2673–2682, 2022. 2

[43] Yonggang Qi and Zheng-Hua Tan. Sketchsegnet+: An end-to-end learning of rnn for multi-class sketch semantic segmentation. *Ieee Access*, 7:102717–102726, 2019. 2

[44] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 1, 2, 3, 5, 6

[45] Yongming Rao, Wenliang Zhao, Guangyi Chen, Yansong Tang, Zheng Zhu, Guan Huang, Jie Zhou, and Jiwen Lu. Denseclip: Language-guided dense prediction with context-aware prompting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18082–18091, 2022. 5

[46] Aneeshan Sain, Ayan Kumar Bhunia, Pinaki Nath Chowdhury, Subhadeep Koley, Tao Xiang, and Yi-Zhe Song. Clip for all things zero-shot sketch-based image retrieval, fine-grained or not. *arXiv preprint arXiv:2303.13440*, 2023. 2

[47] Patsorn Sangkloy, Wittawat Jitkrittum, Diyi Yang, and James Hays. A sketch is worth a thousand words: Image retrieval with text and sketch. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXVIII*, pages 251–267. Springer, 2022. 2

[48] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1): 61–80, 2008. 2

[49] Peter Schaldenbrand, Zhixuan Liu, and Jean Oh. Styleclipdraw: Coupling content and style in text-to-drawing synthesis. *arXiv preprint arXiv:2111.03133*, 2021. 2

[50] Kristofer Schlachter, Benjamin Ahlbrand, Zhu Wang, Ken Perlin, and Valerio Ortenzi. Zero-shot multi-modal artist-controlled retrieval and exploration of 3d object sets. In *SIGGRAPH Asia 2022 Technical Communications*, pages 1–4. 2022. 2

[51] Zhenbang Sun, Changhu Wang, Liqing Zhang, and Lei Zhang. Free hand-drawn sketch segmentation. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part I 12*, pages 626–639. Springer, 2012. 2

[52] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021. 4

[53] Yael Vinker, Yuval Alaluf, Daniel Cohen-Or, and Ariel Shamir. Clipascene: Scene sketching with different types and levels of abstraction. *arXiv preprint arXiv:2211.17256*, 2022. 2

[54] Yael Vinker, Ehsan Pajouheshgar, Jessica Y Bo, Roman Christian Bachmann, Amit Haim Bermano, Daniel Cohen-Or, Amir Zamir, and Ariel Shamir. Clipasso: Semantically-aware object sketching. *ACM Transactions on Graphics (TOG)*, 41(4):1–11, 2022. 2

[55] Fei Wang, Shujin Lin, Hanhui Li, Hefeng Wu, Tie Cai, Xiaonan Luo, and Ruomei Wang. Multi-column point-cnn for sketch segmentation. *Neurocomputing*, 392:50–59, 2020. 2

[56] Yunchao Wei, Huaxin Xiao, Honghui Shi, Zequn Jie, Jiashi Feng, and Thomas S Huang. Revisiting dilated convolution: A simple approach for weakly-and semi-supervised semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018. 2

[57] Xingyuan Wu, Yonggang Qi, Jun Liu, and Jie Yang. Sketchsegnet: A rnn model for labeling sketch strokes. In *2018 IEEE 28th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 2018. 2, 5

[58] Jiarui Xu, Shalini De Mello, Sifei Liu, Wonmin Byeon, Thomas Breuel, Jan Kautz, and Xiaolong Wang. Groupvit: Semantic segmentation emerges from text supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18134–18144, 2022. 2, 5, 6

[59] Jie Yang, Aihua Ke, Yaoxiang Yu, and Bo Cai. Scene sketch semantic segmentation with hierarchical transformer. *Knowledge-Based Systems*, page 110962, 2023. 2

[60] Lumin Yang, Jiajie Zhuang, Hongbo Fu, Xiangzhi Wei, Kun Zhou, and Youyi Zheng. Sketchgnn: Semantic sketch segmentation with graph neural networks. *ACM Trans. Graph.*, 40(3):1–13, 2021. 2

[61] Ruichen Yao, Ziteng Cui, Xiaoxiao Li, and Lin Gu. Improving fairness in image classification via sketching. *arXiv preprint arXiv:2211.00168*, 2022. 2

[62] Andrii Zadaianchuk, Matthaeus Kleindessner, Yi Zhu, Francesco Locatello, and Thomas Brox. Unsupervised semantic segmentation with self-supervised object-centric representations. *arXiv preprint arXiv:2207.05027*, 2022. 2

[63] Zhengming Zhang, Xiaoming Deng, Jinyao Li, Yukun Lai, Cuixia Ma, Yongjin Liu, and Hongan Wang. Stroke-based semantic segmentation for scene-level free-hand sketches. *The Visual Computer*, pages 1–13, 2022. 2, 5

[64] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017. 2

[65] Xin-Yang Zheng, Hao Pan, Peng-Shuai Wang, Xin Tong, Yang Liu, and Heung-Yeung Shum. Locally attentional sdf diffusion for controllable 3d shape generation. *ACM TOG, Proc. SIGGRAPH*, 2023. 2

[66] Yixiao Zheng, Jiyang Xie, Aneeshan Sain, Yi-Zhe Song, and Zhanyu Ma. Sketch-segformer: Transformer-based segmentation for figurative and creative sketches. *IEEE Transactions on Image Processing*, 2023. 2

[67] Chong Zhou, Chen Change Loy, and Bo Dai. Denseclip: Extract free dense labels from clip. *arXiv preprint arXiv:2112.01071*, 2021. 6

[68] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 2

[69] Ziqin Zhou, Yinjie Lei, Bowen Zhang, Lingqiao Liu, and Yifan Liu. Zegclip: Towards adapting clip for zero-shot semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11175–11185, 2023. 2, 5, 6

[70] Xianyi Zhu, Yi Xiao, and Yan Zheng. Part-level sketch segmentation and labeling using dual-cnn. In *Neural Information Processing: 25th International Conference, ICONIP 2018, Siem Reap, Cambodia, December 13-16, 2018, Proceedings, Part I 25*, pages 374–384. Springer, 2018. 2

[71] Xianyi Zhu, Yi Xiao, and Yan Zheng. 2d freehand sketch labeling using cnn and crf. *Multimed. Tools. Appl.*, 79(1), 2020. 2

[72] Y Zhu, Z Zhang, C Wu, Z Zhang, T He, H Zhang, R Manmatha, M Li, and A Smola. Improving semantic segmentation via self-training. arxiv 2020. *arXiv preprint arXiv:2004.14960*, 2021. 2

[73] Changqing Zou, Qian Yu, Ruofei Du, Haoran Mo, Yi-Zhe Song, Tao Xiang, Chengying Gao, Baoquan Chen, and Hao Zhang. Sketchyscene: Richly-annotated scene sketches. In *Proceedings of the european conference on computer vision (ECCV)*, pages 421–436, 2018. 2