Supplementary Material
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/

S1. Overview of the Supplementary Material

- In Sec. S2.1, we provide **additional visual comparisons** of the results obtained with our method versus results obtained with the state-of-the-art language-supervised image segmentation methods.
- In Sec. S2.2, we analyze **segmentation accuracy per category**.
- In Sec. S2.3, we further investigate **the generalization properties** of our method and how it compares with fully-supervised methods.
- In Sec. S3, we provide a more in-depth discussion of Sec. 5: *Human-model alignment* of the main paper.
- In Sec. S4.1, we provide a detailed analysis of the **benefit of using cross-attention**.
- In Sec. S4.2, we analyze different models' performance depending on **the choice of a checkpoint**: the last checkpoint versus the checkpoint optimal on the validation set.
- In Sec. S4.3, we discuss in detail **the choice of a threshold value** for segmenting out pixels corresponding to **individual categories**.
- In Sec. S5, we provide the **computational cost** of our method.

S2. Additional Performance Analysis

S2.1. Additional Qualitative Comparisons

In the main paper, we show in Tab. 2 a numerical comparison of the segmentation results obtained with our method and the segmentation results obtained with the state-of-the-art language-supervised image segmentation methods. Also, in the main paper, in Fig. 5, we show a comparison of our model with CLIP_Surgery**, where *CLIP Surgery*** represents the fine-tuned CLIP_Surgery [5] model with v-v self-attention introduced at both training and inference stages. Here, in Figs. S1 and S2, we provide an additional visual comparison between our method and state-of-the-art language-supervised image segmentation methods: GroupViT [8], SegCLIP [6], CLIP_Surgery [5], fine-tuned on the FS-COCO dataset. The fine-tuned versions of these models are denoted as GroupViT**, SegCLIP*, CLIP_Surgery**, respectively. In Figs. S1 and S2, we show segmentation results and the error maps (in red), which visualize incorrectly labeled pixels for each method.

S2.2. Segmentation Accuracy Analysis by Category

In this section, we analyze segmentation accuracy *per category in both the train and test sets*. We show in Fig. S3 the pixel accuracy (*Acc@P*) for each selected object category. For the figure, we selected categories that appear more than ten times in the FS-COCO dataset [2] captions. First, we can see that the segmentation accuracy is smoothly distributed across different categories.

Next, we investigate whether more frequent categories are more likely to be labeled accurately. To evaluate this, we approximate the frequency of a category by counting its occurrence in both the train and test sets, then consider only categories that appear in the test set. We plot with green and red lines in Fig. S3 the train and test sets category frequency, respectively.

The figure clearly shows a lack of correlation between the frequency of category occurrence and its segmentation accuracy.

We further evaluate it numerically by computing the correlation between x, the pixel accuracy (Acc@P) of each category, and y the occurrence frequency of this category:

$$Corr = \frac{N(\sum xy) - (\sum x)(\sum y)}{\sqrt{[N\sum x^2 - (\sum x)^2][N\sum y^2 - (\sum y)^2]}} \quad (1)$$

where N is the number of categories in the test set.

The resulting correlation coefficients for both train and test sets are 0.16 and 0.14, respectively. This suggests a very weak accuracy-frequency correspondence, indicating that our model is not biased toward more frequently occurring categories. We hypothesize that this is in part due to

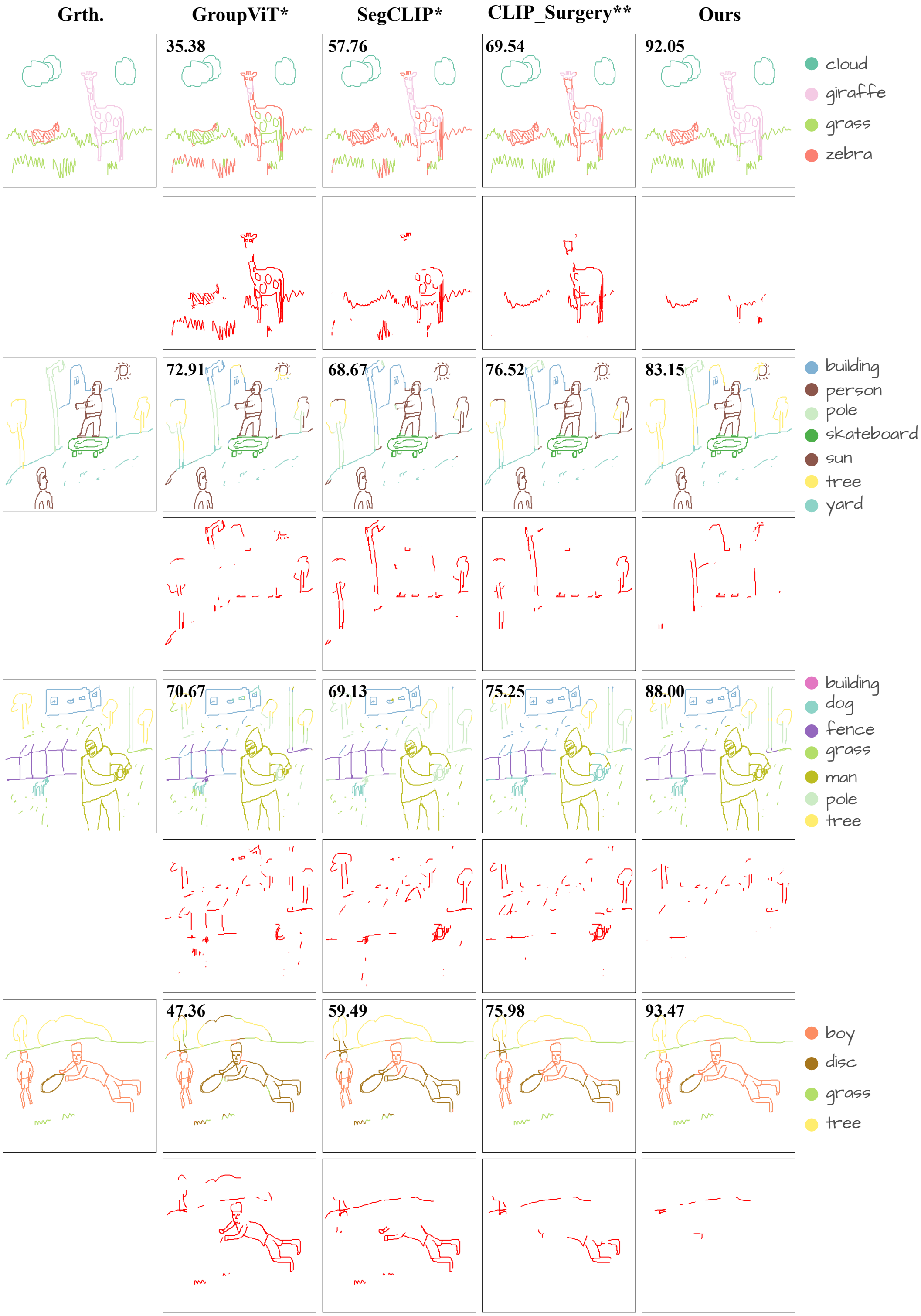

Figure S1. **Part-1**: Visual comparison of our method against state-of-the-art language supervised image segmentation methods, trained on the FS-COCO dataset [2]. The numbers show Acc@P values. The error maps in red represent the misclassified pixels.

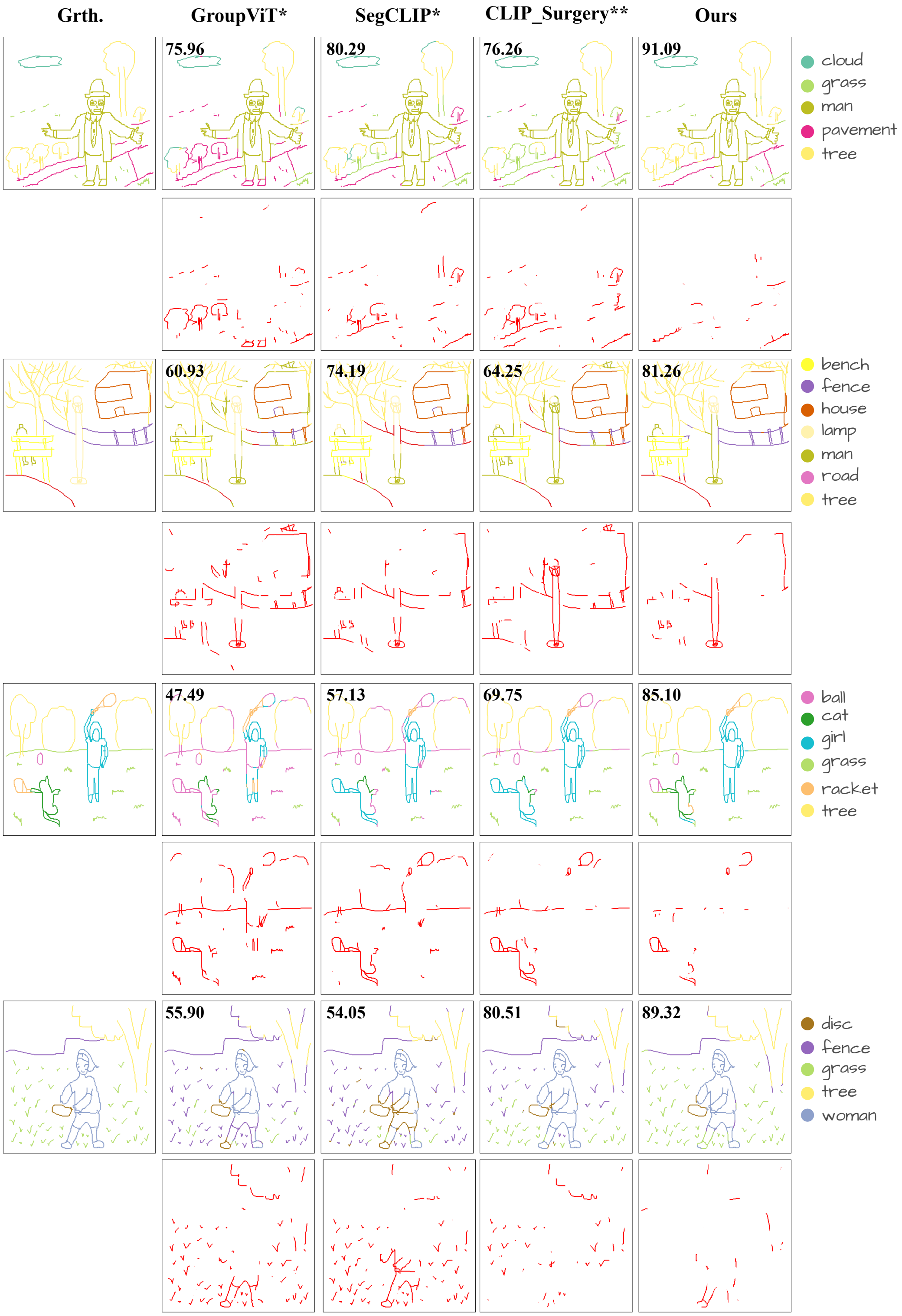

Figure S2. **Part-2**: Visual comparison of our method against state-of-the-art language supervised image segmentation methods, trained on the FS-COCO dataset [2]. The numbers show Acc@P values. The error maps in red represent the misclassified pixels.

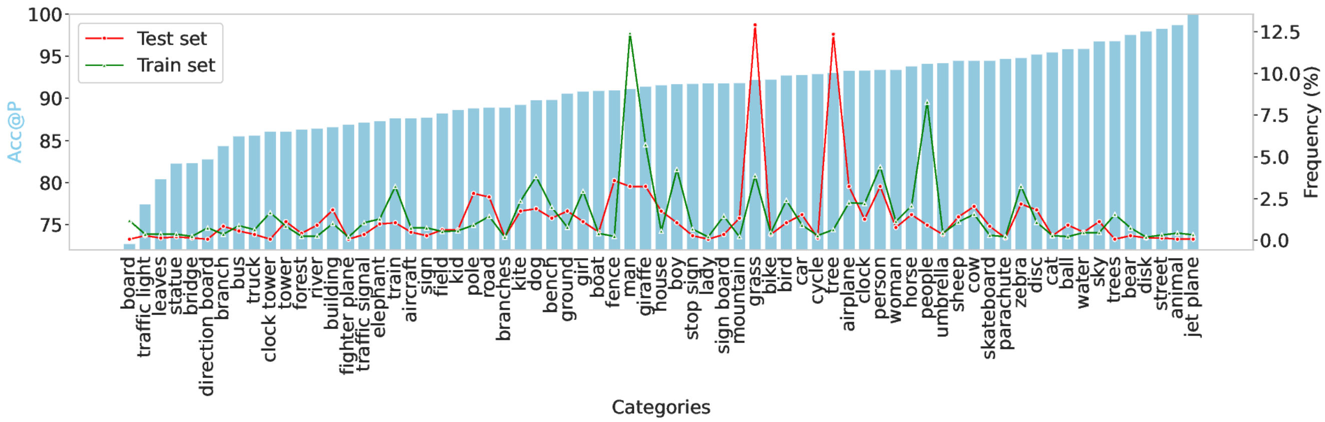

Figure S3. Blue bars show pixel accuracy (Acc@P) for each object category with more than 10 appearances in FS-COCO dataset [2] captions. The green line shows the frequency of occurrence of each category in the train set. The red line shows the frequency of occurrence of each category in the test set. Please see Sec. S2.2 for an additional discussion.

our careful fine-tuning strategy, which prevents over-fitting. Therefore, the model efficiently leverages pre-training on a large image dataset.

Model generalization to new object categories Our test set includes 185 object classes, with 125 seen and 60 unseen during training. The accuracy on seen categories is 86.35% and 84.68% on unseen. These results demonstrate *good generalization* of our model to unseen categories.

S2.3. Synthetic vs. Freehand sketches

In the Sec. 4.4.3 in the main paper, to better understand the generalization properties of our model, we evaluated our method trained on the sketches from the FS-COCO dataset [2] on the freehand sketches from [4]. Here, we provide additional analysis of generalization properties.

S2.3.1 Generalization to sketches consisting of clip-art-like object sketches

Here, we additionally evaluate our method on the SketchyScene [10] dataset. The SketchyScene [10] 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. This is a dataset with sketches with a large domain gap from the freehand scene sketches we target. Yet, it is interesting to evaluate the generalization properties of our method. Tab. S1 shows a comparison of the zero-shot performance of our method (*third line: Ours*) with the two fully-supervised methods trained on semi-synthetic sketches. The *Acc@P* and *mIoU* are the metrics we use in the main paper. We additionally report results for two additional measures:

Method	$Acc@P$	$MAcc$	$mIoU$	$FWIoU$
LDP [4]	93.46	85.84	74.93	88.13
SketchSeger [9]	95.44	88.18	81.17	91.52
Ours	87.99	66.59	60.91	76.33
Ours*	92.87	79.54	71.73	85.19
Ours**	91.23	77.87	70.51	84.72

Table S1. Comparison of our method with state-of-the-art fully supervised scene sketch segmentation methods on the sketches from the SketchyScene [10] dataset. *Ours*: trained on freehand sketches from the FS-COCO dataset [2] (zero-shot performance), *Ours** is trained on synthetic sketches [10], *Ours*** is trained on both freehand [2] and synthetic sketches [10].

- **Mean Pixel Accuracy (MeanAcc)**: It measures the average pixel accuracy Acc@P of each category.
- **Frequency Weighted Intersection over Union (FWIoU):** It introduces category occurrence frequency to the mIoU, by weighting per-category pixel IoU (intersection over union) by the frequency of occurrence.

Our model reaches high accuracy on these sketches, even in the presence of a large domain gap. In particular, the performance of our model on these sketches is higher than on the freehand and more challenging sketches from the FS-COCO dataset [2]. This, combined with the results in Tab. 3 in the main paper, is a strong argument towards usage of *true* freehand sketches with weak annotation in the form of captions over the semi-synthetic dataset of scene sketches.

Fine-tuning on semi-synthetic sketches While our model does reach high accuracy on these sketches, it does

Method	Trained on	Supervision		Tested on	Segmentation accuracy				
		Pixel labels	Captions		mIoU	Acc@P	Acc@C	MAcc	FWIoU
LDP [4]	SketchyScene ∪ ∪ SKY-Scene ∪ ∪ TUB-Scene	✓		FS-COCO	33.04	56.23	56.71	51.16	52.63
				LDP freehand	37.16	78.84	-	47.25	66.98
				SketchyScene	74.93	93.46	-	85.84	88.13
SketchSeger [9]	SketchyScene ∪ ∪ SKY-Scene ∪ ∪ TUB-Scene	✓		FS-COCO	-	-	-	-	-
				LDP freehand	-	-	-	-	-
				SketchyScene	81.17	95.44	-	88.18	91.52
Ours	FS-COCO		✓	FS-COCO	73.48	85.54	87.02	82.27	84.09
				LDP freehand	53.94	81.63	-	59.36	69.37
				SketchyScene	60.91	87.99	-	66.59	76.33
Ours*	SketchyScene		✓	FS-COCO	61.79	74.43	75.62	69.41	71.75
				LDP freehand	49.72	71.96	-	48.71	59.15
				SketchyScene	71.73	92.87	-	79.54	85.19
Ours**	FS-COCO ∪ ∪ SketchyScene		✓	FS-COCO	68.84	79.21	81.29	74.08	77.63
				LDP freehand	50.13	76.07	-	55.83	62.97
				SketchyScene	70.51	91.23	-	77.87	84.72

Table S2. Comparison of our method with state-of-the-art fully supervised scene sketch segmentation methods in different setups.
Ours: trained on freehand sketches from the FS-COCO dataset [2], *Ours** is trained on synthetic sketches [10], *Ours*** is trained on both freehand [2] and synthetic sketches [10].
We test all methods on three datasets: our FS-COCO-based test set, LDP [4] freehand sketches test set, and SketchyScene [10] synthetic sketches test set.
Training datasets: The SketchyScene [10] dataset contains 7,265 synthetic scene sketches spanning 46 categories with 5,617 images for training, and 1,113 for test. SKY-Scene and TUB-Scene were introduced in [4], and are composed of object sketches from the Sketchy [7] and TU-Berlin [3] datasets, respectively. They both have 7,265 synthetic scene sketches and follow the same data split.

Method	$mIoU$	$Acc@P$	$Acc@C$
LDP [4]	33.04	56.23	56.71
Ours	73.48	85.54	87.02
Ours*	61.79	74.43	75.62
Ours**	68.84	79.21	81.29

Table S3. Comparison on the freehand sketches from the FS-COCO dataset [2] of our method with state-of-the-art fully supervised scene sketch segmentation method LDP [4]. LDP [4] is trained on semi-synthetic sketches. *Ours*: trained on freehand sketches from the FS-COCO dataset [2], *Ours** is trained on synthetic sketches [10], *Ours*** is trained on both freehand [2] and synthetic sketches [10]. *We do not compare here with SketchSeger [9], as there is no code available and we can not run it on sketches from the FS-COCO dataset [2].*

not reach the performance of fully supervised methods trained on semi-synthetic sketches when tested on semi-synthetic sketches. Therefore, we investigate whether fine-tuning our model on semi-synthetic sketches can close the gap – while relying only on textual labels and not pixel-level annotations.

We perform two additional experiments:

1. **Training exclusively on Synthetic Sketches (Ours*):** We train our model on the SketchyScene synthetic sketches [10] using language supervision. Captions are constructed by concatenating scene sketch category names into one text token.
2. **Training on Both Synthetic and Freehand Sketches (Ours**):** We train the model on both SketchyScene synthetic sketches and FS-COCO freehand sketches.

The results are shown in Tab. S1: Ours* and Ours**.

We observe a performance increase for *Ours** on the sketches from the SketchyScene [10] dataset, reaching competitive performance with fully supervised methods [4, 9]. *This highlights the generalization properties of our training pipeline for different data distributions and highlights that succinct captions can serve as a robust supervisory signal, lifting the need for extensive annotations.*

However, when freehand sketches are added to the training data (*Ours***), there is a slight decrease in performance across all metrics. *This further emphasizes the existence of a domain gap between freehand sketches and semi-synthetic sketches, which again motivates the usage of freehand sketches with weak annotations.*

Similar observations are made in Tab. S3 when the model is trained on the synthetic sketches (*Ours**) and tested on the FS-COCO freehand sketches. Even when both synthetic and freehand sketches are used for training (*Ours***), the model's performance degrades compared to training solely on freehand sketches. This further emphasizes our observations regarding the domain gap between synthetic and freehand sketches.

Tab. S2 shows a full comparison of our method against fully supervised sketch segmentation methods: LDP [4] and SketchSeger [9], across the free datasets: FS-COCO-based test set, LDP [4] freehand sketches test set, and SketchyScene [10] synthetic sketches test set. It shows the superiority of our method on both datasets of freehand scene sketches.

S2.3.2 Pre-training on synthetic sketches

We also experiment with fine-tuning CLIP and CLIP-Surgery on synthetic sketches. However, training on millions of synthetic sketches is out of the scope of this work due to computational constraints. As a feasible experiment, we generated 9025 synthetic sketches for the reference images in our training set, using [1], in 'contour' style (as the closest to the test set sketches style). This is the number of sketches identical to the number of sketches we use to train our model. The accuracy on our test set of fine-tuned this way CLIP and CLIPSurgery increases by negligible 2 to 3 points compared to their zero-shot performance. In comparison, our model outperforms their zero-shot performance by 56.72% and 13.07% points, respectively. Training our model from CLIP weights pre-trained on synthetic sketches boosts the performance only by 0.42 points.

S3. Detailed human Study Analysis

In this section, we provide a more in-depth discussion of Sec. 5: *Human-model alignment* of the main paper.

S3.1. Human Study Categories

In the main paper, in Sec. 5.1, we introduced four challenging categories of sketches for our method, that we used for the user study. We show all the sketches used in the user study in Fig. S4. For convenience, below we repeat the definition of each category:

(1) **Ambiguous sketches**: sketches where it might be hard even for a human observer to understand an input sketch. We selected the sketches by visually examining the test set sketches alongside reference images.

(2) **Interchangeable categories**: sketches containing multiple objects with labels that can interchange each other, such as *'tower/building'*, *'girl/man'*, and *'ground/grass'*.

(2) **Correlated categories**: sketches with categories that typically co-occur in scenes. These categories are semantically related. We selected sketches containing the most common pairs with significant co-occurrence. Specifically, *'branch/bird'* (52%), *'runway/airplane'* (44%), *'railway/train'* (39%), and *'road/car'* (29%), were chosen.

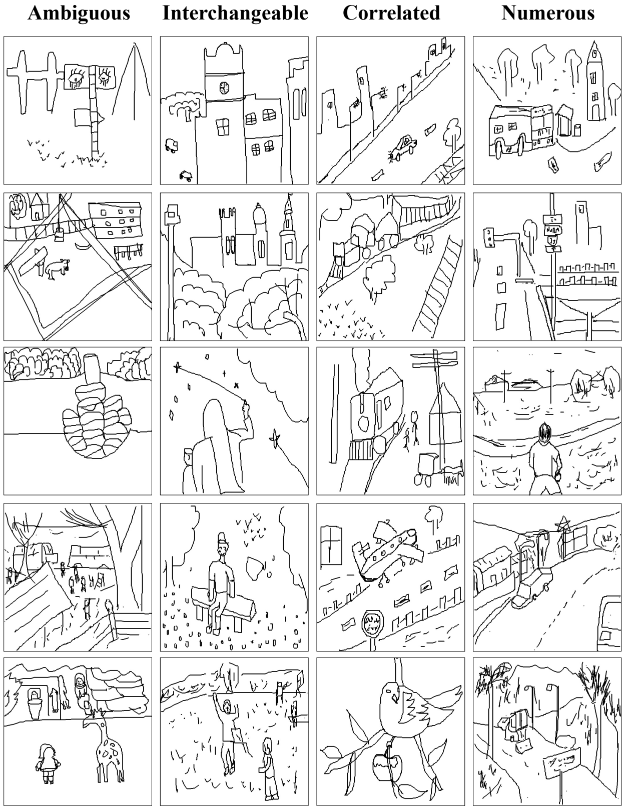

Figure S4. Visualization of the selected sketches for the four challenging sketch categories used in the user study. Please see Sec. S3.1 for the description of categories.

(4) **Numerous-categories**: sketches with six or more object categories and a model accuracy (*Acc@P*) below 80%. The sampled sketches have an average of 6.4 categories per sketch $(7, 7, 6, 6, 6)$.

Additionally, we included a **Strong performance** category, comprising ten sketches where the model's accuracy (*Acc@P*) exceeded the average performance (85.54%), to demonstrate scenarios of effective model segmentation.

S3.2. Annotators

We recruited 25 participants (14 male). The annotators are PhD students in diverse disciplines and of diverse nationalities aged from 22 to 42 years (average age 29.32). We believe this group represents well the general population and each individual performed the task carefully.

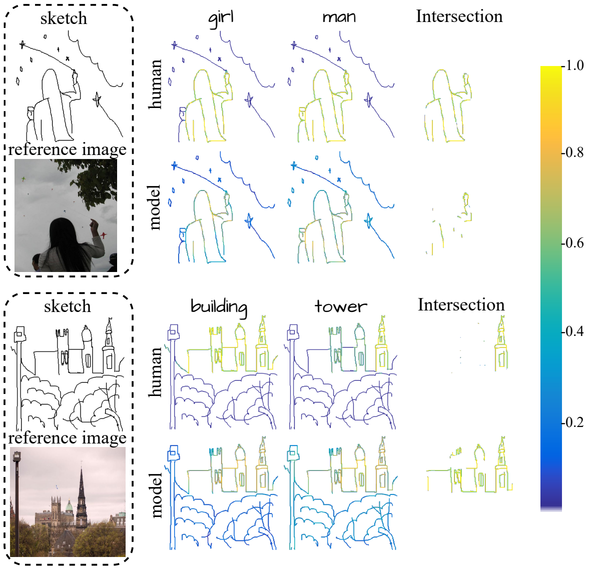

Figure S5. Visualizations of the confidence in segmenting semantically similar objects by human annotators and our model. *Intersection* shows the pixels that are confidently assigned to belong to both considered categories (with a confidence threshold higher than 60%). Please see Sec. S3.3 for the discussion.

S3.3. Visual Analysis of Interchangeable Categories Segmentation Results

We conducted a visual analysis to compare the confidence in segmenting semantically similar objects by human annotators and our model. For each object category, we obtain a category confidence map by counting how many participants assigned a given label to a category. For our model, we obtain segmentation confidence as a result of a cosine similarity computation between the sketch patch features and the category textual embedding. We visualized in Fig. S5 the obtained confidence maps for the most frequently confused by our model categories: 'girl/man' and 'building/tower'. We also show the pixels that are confidently assigned to belong to both considered categories (with a confidence threshold higher than 60%). We can observe that our model is less confident than humans in assigning labels to these categories.

S3.4. Statistical Significance: Ours vs CLIPSurgery

On the 20 sketches from the 4 challenging groups, our model outperforms CLIPSurgery with a p-value of 2×10^{-5}. In the 'strong' group, we have 10 sketches, in which, while our model performs on par with humans, it outperforms CLIPSurgery with a p-value of 0.005.

Dropout	0	0.2	0.4	0.6	0.8	1
Acc@P	85.54	84.61	84.38	84.16	83.06	82.86

Table S4. Acc@P with different cross attention dropout ratios

S4. Additional Ablation Studies

S4.1. Detailed Ablation on Cross Attention vs. Self Attention

To validate the effectiveness of our cross-attention module, we added a residual connection to demonstrate that relying solely on self-attention features, without the integration of cross-attention, leads to suboptimal segmentation results. We run several experiments with varying dropout ratios in the cross-attention block. This allows us to assess its impact on model performance. The results, presented in Tab. S4, show model accuracy across different dropout levels, from 0 (no dropout) to 1 (complete dropout). This shows the benefit of the design used in the main paper, equivalent to using only cross-attention in the category-level encoder.

S4.2. Models Checkpoint Choice

As described in Sec. 4.3 of the main paper, for each of the models fine-tuned on sketch data: ours and competing methods, we select a checkpoint based on the performance on the validation set with pixel-level segmentation annotations, consisting of 475 sketches. This requires at training time having a small set of pixel-level annotated sketches, which can be limiting. However, we observe that the loss gradually decreases for our model, and it is safe to choose a last checkpoint if such an annotated set is not available. In Tab. S5, we provide a comparison with the results when for our model and competing models the last checkpoint is used. We trained for 20 epochs. We observe that after that the convergence rate is very low for each of the considered models.

We observe only a marginal performance drop (less than one point in all metrics) for our model when the last checkpoint is used compared to a checkpoint selected based on the performance on the validation set (referred to as *optimal* in the table). This implies that competitive model performance can be achieved without using any pixel-level annotations.

We also observe that with either of the choices of a checkpoint, the performance on the validation and test sets is similar, with just a small decrease in performance on the test set compared to the validation set. Our test set includes sketches from five non-expert artists whose sketches were not present in either the training or validation sets. Therefore, this analysis implies that there is no over-fitting to the training data and our model robustly generalizes to the unseen sketches and drawing styles.

Model	Checkpoint	Test set			Validation set		
		mIoU	Acc@P	Acc@S	mIoU	Acc@P	Acc@S
CLIP*	Optimal	22.86	33.41	32.64	25.76	36.34	35.17
	Last	19.34	28.89	27.64	22.11	31.49	31.07
GroupViT*	Optimal	45.71	66.21	66.89	47.26	68.28	68.76
	Last	43.83	64.03	64.48	46.58	67.70	68.13
SegCLIP*	Optimal	49.26	69.87	73.64	51.27	71.79	75.67
	Last	46.41	66.91	70.31	50.86	70.12	74.41
CLIP_Surgery*	Optimal	48.74	65.38	68.78	50.84	67.32	70.88
	Last	47.29	63.94	67.13	48.33	66.01	68.82
CLIP_Surgery**	Optimal	59.98	78.68	81.11	62.41	80.69	83.23
	Last	58.64	77.34	79.88	61.53	79.41	82.07
Ours	Optimal	73.48	85.57	87.02	74.76	86.83	88.41
	Last	72.51	84.74	86.39	74.12	85.97	87.76

Table S5. Models performance comparison on test and validation sets using two different checkpoint choices: (a) *Optimal:* A checkpoint selected based on the performance on the pixel-level annotated validation set, and (b) *Last:* The checkpoint obtained after training each model for 20 epochs. Please see Sec. S4.2 for the in-depth discussion.

	Training			Parameters		Inference	
	GFLOPS	Hours	Epochs	# Trainable	# Full	GFLOPS	Secs.
CLIP	189.47	3.76	20	149M	149M	89.02	0.21
CLIPSurgery	231.41	4.08	20	162M	162M	113.64	0.73
Ours	346.36	7.31	20	16M	165M	121.59	1.05

Table S6. Note that the parameters of cross-attention layers (added complexity in our model over CLIPSurgery) are used only during training.

S4.3. Segmenting out Individual Categories

To explore the model's ability to isolate individual sketch categories through thresholding, as described in Sec. 3.4 in the main paper, we assess two model versions, where (1) the *optimal* checkpoint is used, selected based on the performance on the validation set and (2) the *last* checkpoint is used (from the 20th epoch). We measure pixel accuracy (*Acc@P*) of segmenting a sketch into an individual category and the rest (background), employing varying threshold values. Fig. S6 shows the plot of segmentation accuracy with different threshold values on test and validation sets when either optimal Fig. S6(a.) or last Fig. S6(b.) checkpoints are used.

When optimal checkpoint is used When using the optimal checkpoint, the model consistently achieves strong performance on validation and test sets, achieving 86.06% and 85.71% *Acc@P*, respectively, albeit at different threshold values (0.79 and 0.71, respectively). This implies that the label assignment confidence is slightly lower on the unseen sketches in new styles. However, despite this, the model maintains a consistently strong performance on these new sketches and styles.

When the last checkpoint is used When we use the model from the last checkpoint, the best performance on the validation and test sets is obtained with slightly lower threshold values of 0.73 and 0.68, respectively. This implies that there is a correlation between the model's confidence and its performance.

S5. Computational Cost

We detail in Tab. S6 the computational cost of our method compared to CLIP and CLIPSurgery. Our two-level hierarchical network design introduces additional complexity, through value-value self-attention and cross-attention blocks. However, we maintain a comparable level of complexity to CLIPSurgery during inference. This slight computational increase is justified given the substantial 13 mIoU points improvement over CLIP_Surgery** (as shown in Tab. 4 in the main paper). Our code can be further optimized to reach the performance of CLIPSurgery at inference time.

References

[1] Caroline Chan, Fredo Durand, and Phillip Isola. Learning to generate line drawings that convey geometry and semantics. 2022. 6

[2] Pinaki Nath Chowdhury, Aneeshan Sain, Ayan Kumar Bhunia, Tao Xiang, Yulia Gryaditskaya, and Yi-Zhe Song. Fs-

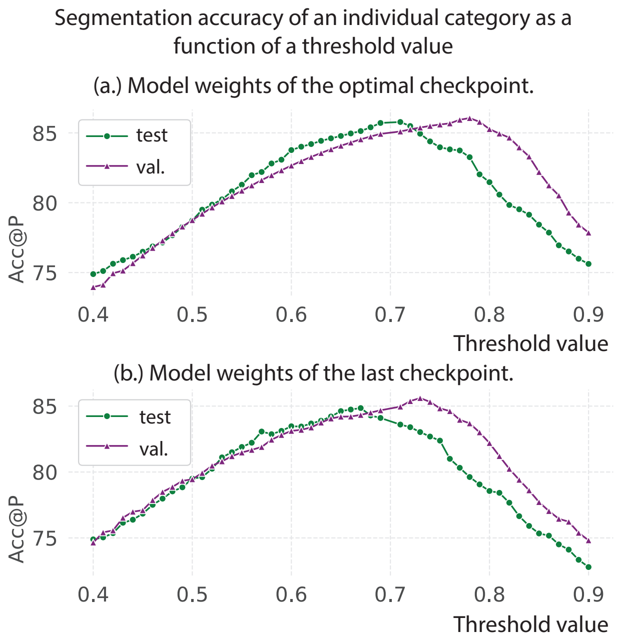

Figure S6. *Acc@P* values on test and validation sets (green and purple lines, respectively) for single category versus the rest segmentation task, as a function of a threshold value. The plots are shown for the two different choices of a checkpoint. (a) *Optimal:* A checkpoint is selected based on the performance on the pixel-level annotated validation set, and (b) *Last:* The checkpoint is obtained after training each model for 20 epochs. Please see Sec. S4.3 for an in-depth discussion.

coco: 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, 3, 4, 5

[3] Mathias Eitz, James Hays, and Marc Alexa. How do humans sketch objects? *ACM Transactions on graphics (TOG)*, 31 (4):1–10, 2012. 5

[4] 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. 4, 5, 6

[5] Yi Li, Hualiang Wang, Yiqun Duan, and Xiaomeng Li. Clip surgery for better explainability with enhancement in open-vocabulary tasks, 2023. 1

[6] 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. 1

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

[8] 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. 1

[9] Jie Yang, Aihua Ke, Yaoxiang Yu, and Bo Cai. Scene sketch semantic segmentation with hierarchical transformer. *Knowledge-Based Systems*, page 110962, 2023. 4, 5, 6

[10] 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. 4, 5, 6