Class-Agnostic Visio-Temporal Scene Sketch Semantic Segmentation - Supplementary Material -

Aleyna Kütük
KUIS AI Center
Koç University, İstanbul, Turkey
akutuk21@ku.edu.tr

Tevfik Metin Sezgin
KUIS AI Center
Koç University, İstanbul, Turkey
mtsezgin@ku.edu.tr

The Supplementary Material is organized as follows. The details regarding the Post-Processing Module of CAVT are provided in Section S1. RGB Coloring Technique is detailed in Section S2. Additional analysis on external classifiers is provided in Section S3. Additional visual results are shared for scene sketch segmentation in Section S4. Lastly, additional analysis and discussions regarding to FrISS dataset and UI of data collection web application are shared in Section S5.

S1. Details on Post-Processing Module

S1.1. Hyperparameter Optimization

The complete algorithm for the post-processing module is outlined in Algorithm 1. Furthermore, we provide details of our grid-search approach used to determine the optimal hyperparameter combination for the post-processing module. We evaluated the AoN and S-IoU scores on the validation sets of both CBSC and FrISS and selected the top-performing parameter combination based on the average of all scores. Table S1 presents the results for the top-performing parameter combination. The parameters in the ablation study are explained as follows:

- ***IoU_threshold:*** The threshold value determines the Intersection over Union (IoU) of stroke sequences to boxes. For each box, if the IoU between the box and the longest intersecting stroke sequence exceeds *IoU_threshold*, the sequence is assigned to that box. For the ablation study, we adjusted the threshold within a range of 25% to 85%, increasing by 10% increments.

- ***OR_threshold:*** This is the threshold value that determines the assignment of remaining stroke sequences to boxes. If the overlap ratio of the longest unassigned stroke sequence with its nearest box exceeds *OR_threshold*, the sequence is assigned to that box. For the ablation study, we set the threshold ranges from 30% to 80% in 5% increments.

Algorithm 1: Post-Processing Module

Input: boxes, IoU_threshold, OR_threshold
Output: segmented stroke groups
while *there is alternation in stroke grouping* **do**
  Mark all strokes as unassigned.
  Sort the boxes by area in ascending order.
  for *each box* b_i *in boxes* **do**
    Find the longest stroke sequence S that has the highest IoU with the box b_i.
    if *the overlap ratio between* S *and* b_i *is more than IoU_threshold* **then**
      Assign stroke sequence S to bounding box b_i.
  for *each unassigned longest stroke sequence* S_u **do**
    Find the nearest bounding box b_i.
    if *the overlap ratio between* S_u *and* b_i *is more than OR_threshold* **then**
      Assign strokes in S_u to bounding box b_i.
  for *each longest stroke sequence* S_u *that are unassigned* **do**
    Find the boundaries S_u: x_min, y_min, x_max, y_max.
    Define a new box b_{new} from values x_min, y_min, x_max, y_max.
    Append b_{new} to the boxes.
    Assign each stroke in S_u to b_{new}.
  for *each box* b_i *in boxes* **do**
    Update the coordinates of each b_i according to the most recent assignment of strokes.

- ***num_repeats:*** This refers to the total number of iterations the post-processing module undergoes to complete the stroke assignment process. The post-processing module continues until stroke group assignments reach a stable state. However, this approach

Index	*IoU_threshold*	*OR_threshold*	*num_repeats*	*stroke_thickness*	CBSC		FrISS		Avg
					AoN	S-IoU	AoN	S-IoU	
1	65%	60%	3	2	74,17	88,19	57,96	79,20	**74,88**
2	75%	45%	1	2	73,53	87,56	59,37	78,98	74,86
3	55%	60%	3	2	74,17	88,19	57,71	79,32	74,85
4	65%	75%	3	2	73,56	87,94	58,08	79,25	74,71
5	65%	70%	3	2	73,56	87,94	58,10	79,16	74,69
6	55%	55%	5	2	74,07	88,11	57,38	78,99	74,64
7	55%	70%	3	2	73,56	87,94	57,85	79,18	74,63
8	25%	60%	3	2	74,40	88,47	56,67	78,99	74,63
9	45%	60%	3	2	74,17	88,27	56,79	79,11	74,59
10	65%	70%	1	2	73,43	88,02	57,75	79,05	74,56
11	25%	75%	3	2	73,79	88,20	57,03	78,96	74,49
12	45%	75%	3	2	73,56	88,00	57,14	79,09	74,45
13	75%	70%	1	2	72,69	87,64	58,45	78,97	74,44
14	25%	70%	3	2	73,79	88,20	56,81	78,85	74,41
15	35%	75%	3	2	73,34	87,97	57,14	79,09	74,38
16	75%	50%	1	2	72,89	87,61	58,32	78,58	74,35
17	55%	50%	5	2	73,76	87,84	57,02	78,75	74,34
18	35%	75%	1	2	73,21	88,05	56,68	79,17	74,28
19	55%	50%	1	2	73,63	87,89	56,67	78,81	74,25
20	85%	75%	3	1	73,23	87,41	58,40	77,78	74,21
Lowest	85%	50%	7	3	66,97	83,00	53,17	75,74	69,72

Table S1. The top-performing hyperparameter combinations for the post-processing module are presented in descending order.

can increase runtime, so we limited the number of iterations to evaluate the impact of different repetition counts. We tested the effect of the *num_repeats* parameter with values of 1, 3, 5, 7, and 9.

- ***stroke_thickness:*** We assessed the effect of stroke line thickness by evaluating the *stroke_thickness* parameter with values of 1, 2, 3, and 4, where higher values correspond to thicker stroke lines in the scene.

Table S1 illustrates the impact of each parameter, revealing that the best-performing hyperparameter combination includes these values: *IoU_threshold* set to 65%, *OR_threshold* to 60%, *num_repeats* to 3, and *stroke_thickness* to 2. As demonstrated, using a value for *stroke_thickness* different than 2 degrades performance by distorting the features of the sketches. The *num_repeats* parameter does not significantly affect performance when increased, indicating that the stroke assignment operation completes effectively within a few iterations, minimizing the need for extended runtime. Setting *OR_threshold* to a low percentage can lead to incorrect stroke assignments, as some strokes that should be labeled as separate objects are merged with other stroke sequences. Therefore, setting *OR_threshold* higher than 50% generally results in better performance. A range of 55%-65% for *IoU_threshold* yields the best results. Lower *IoU_threshold* values can lead to incorrect stroke-to-box assignments, while higher values may prevent the accurate stroke assignment.

S1.2. Post-Processing Time & Memory Footprint

Our post-processor takes on average 345 milliseconds per scene on CPU and has the memory upper bound of 5 times the scene in vector format.

S2. Additional Details on RGB Coloring Technique

We adopted an RGB coloring technique to maintain a 3-channel input and values ranging from 0 to 255 for the detector. In our design, the neighboring strokes are represented with colors closer in the spectrum that spans from blue to red. Therefore, the strokes of the same object are expected to contain similar colors. Although a single object may not be entirely drawn in one stroke sequence, individual sequences are expected to exhibit consistent patterns. Besides the shape and distance of strokes, we expect our detector to recognize groups of consecutively sketched strokes. An illustrative example of a scene colored according to stroke order is given in Figure S1.

Model	Top-1 Accuracy			Top-3 Accuracy			Top-5 Accuracy		
	CBSC	FrISS-QD	Avg.	CBSC	FrISS-QD	Avg.	CBSC	FrISS-QD	Avg.
SketchR2CNN [7]	63.04	48.65	55.85	71.57	59.13	65.35	74.12	63.06	68.59
MGT [13]	65.29	51.78	58.54	79.22	67.85	73.54	83.63	73.40	78.52
Sketchformer [9]	65.88	52.82	59.35	80.81	66.57	73.69	85.69	71.36	78.53
Inception-V3 [12]	**67.45**	**55.48**	**61.47**	**82.84**	**70.27**	**76.56**	**86.04**	**74.62**	**80.33**

Table S2. Analysis on state-of-the-art single sketch classifiers

Figure S1. Sample scene sketch from the CBSC, which demonstrates the input for our object detector model. Each stroke within the scene is color-coded based on drawing order, utilizing a spectrum ranging from blue to red, as illustrated at the bottom.

Model	Train Dataset		Accuracy		
	QD	FrISS	CBSC	FrISS-QD	Avg.
Inception-V3 [12]	✓		65.69	50.07	57.88
VGG19 [11]	✓		64.02	50.69	57.36
ResNet18 [5]	✓		63.04	48.79	55.92
ResNet50 [5]	✓		62.84	48.03	55.44
MobileNetV3-Small [6]	✓		61.37	46.85	54.11
MobileNetV3-Large [6]	✓		60.88	48.89	54.89
MobileNet-V2 [10]	✓		62.55	47.75	55.15
Inception-V3	✓	✓	67.45	55.48	**61.47**
VGG19	✓	✓	65.98	55.24	60.61
ResNet18	✓	✓	67.65	53.11	60.38

Table S3. The ablation study is performed to measure the effect of different backbone architectures and the effect of including the FrISS dataset in the training set. The highest average score is highlighted in green, the second highest in blue, and the third highest in red for each aspect (i.e., backbone type and FrISS contribution).

S3. Additional Analysis on External Classifiers

To develop a CNN-based sketch classifier, I first train several models, including Inception-V3 [12], VGG19 [11], ResNet18 [5], ResNet50 [5], MobileNet-V3 [6], and MobileNet-V2 [10], using only the QuickDraw dataset. Afterward, I select the top three performing models and conduct further training by incorporating the FrISS training set along with QuickDraw. In both phases of the experiment, Inception-V3 consistently outperforms the other classifiers. Additionally, including the FrISS training set improves overall performance across both datasets. The results are summarized in Table S3. Based on these results, our pretrained Inception-V3 is selected as the external CNN-based classifier in our experiments.

I evaluate the performance of several state-of-the-art stroke-based sketch classifiers [7, 9, 12, 13], and results are provided in Table S2. The highest-performing transformer-based classifier, Sketchformer [9] is outperformed by our pretrained Inception-V3 [12]. To demonstrate the compatibility of CAVT with a stroke-based external classifier, Sketchformer is utilized in an end-to-end manner.

S4. Additional Visual Results on Scene Sketch Semantic Segmentation

In Sec. 5.5 of the main document, we provide a numerical comparison of the segmentation results obtained using our pipelines and two state-of-the-art methods: LDP [3] and OV [1]. Additionally, in Figure 4 from the main document, we present a visual comparison of our method against LDP and OV. Here, we provide additional visual results of our method against state-of-the-art models, assessed on FrISS and CBSC [14] datasets in Figures S2 and S3, respectively. To visualize class-level segmentation results, we colored each pixel or stroke within the scene regarding its predicted object category.

The additional visual outcomes depicted in Figures S2 and S3 demonstrate consistent segmentation results from both our primary pipelines (CAVT-S and CAVT-I) and its variant (CAVT-S* and CAVT-I). Therefore, we can observe that leveraging stroke representations of sketches and the temporal order of stroke sequences is a promising solution for the scene sketch segmentation problem. In some cases, although our class-agnostic approach successfully segments object instances, our adopted classifier may cause a performance drop due to its misclassification. For instance, in the 3rd row of Figure S2, our class-agnostic approach accurately segments the *'sheep'* object. However, our adopted classifiers mislabel *'sheep'* as *'horse'* and *'dog'*, thus impacting the segmentation results at the class level. This highlights the potential for our class-agnostic method's im-

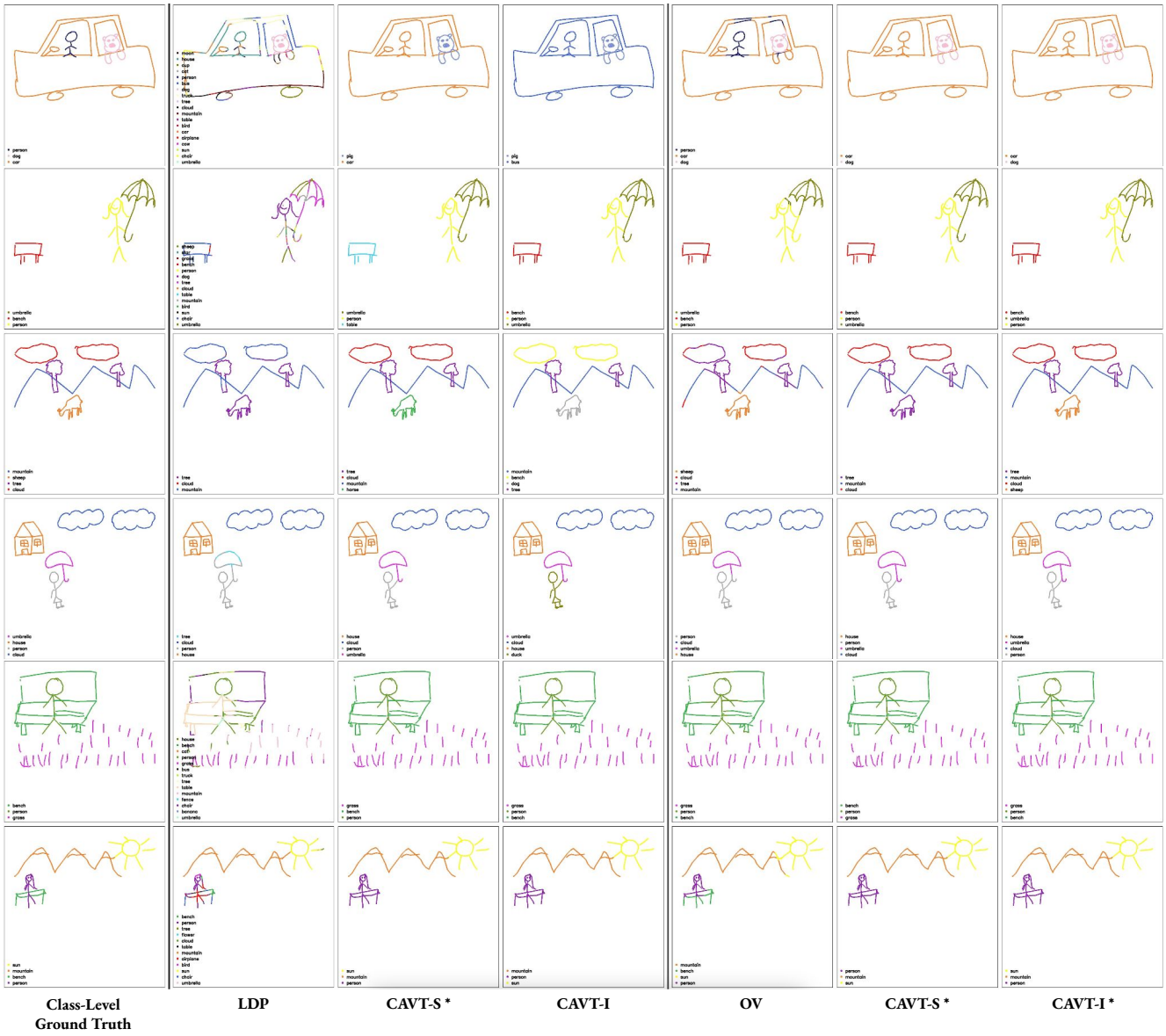

Figure S2. Visual comparison of our method with LDP [3] and OV [1] models, tested on FrISS dataset. We utilize CAVT with the external classifier Sketchformer [9] (CAVT-S) and our pre-trained Inception-V3 [12] (CAVT-I) in an end-to-end manner.

proved performance when paired with a classifier offering more accurate object class predictions. A similar issue is observed for the *'cloud'* object in the 2nd row of Figure S3.

In addition to the class-level results, we share additional instance-level segmentation results in Figure S4. In this figure, we can see that our pipelines successfully segment the objects from the same categories. While two houses are successfully differentiated in the 3rd row, the clouds are successfully detected and identified in the 6th row. However, there also exist some rare cases in which CAVT fails to segment (see individual birds and clouds in the 1st row).

S5. Additional Details on FrISS Dataset

S5.1. UI of Data Collection Web Application

In Sec. 4 of the main document, we provide a detailed discussion of our data collection process. In Figures S5 and S6, we present visuals from the user interface of our data collection web application. As we discussed in the main document, our data collection consists of two distinct phases: sketch collection and sketch annotation. Figure S5 provides an example of the sketch collection phase, where participants are tasked with illustrating a scene within a time frame of 1.5 minutes, using a provided text description as a

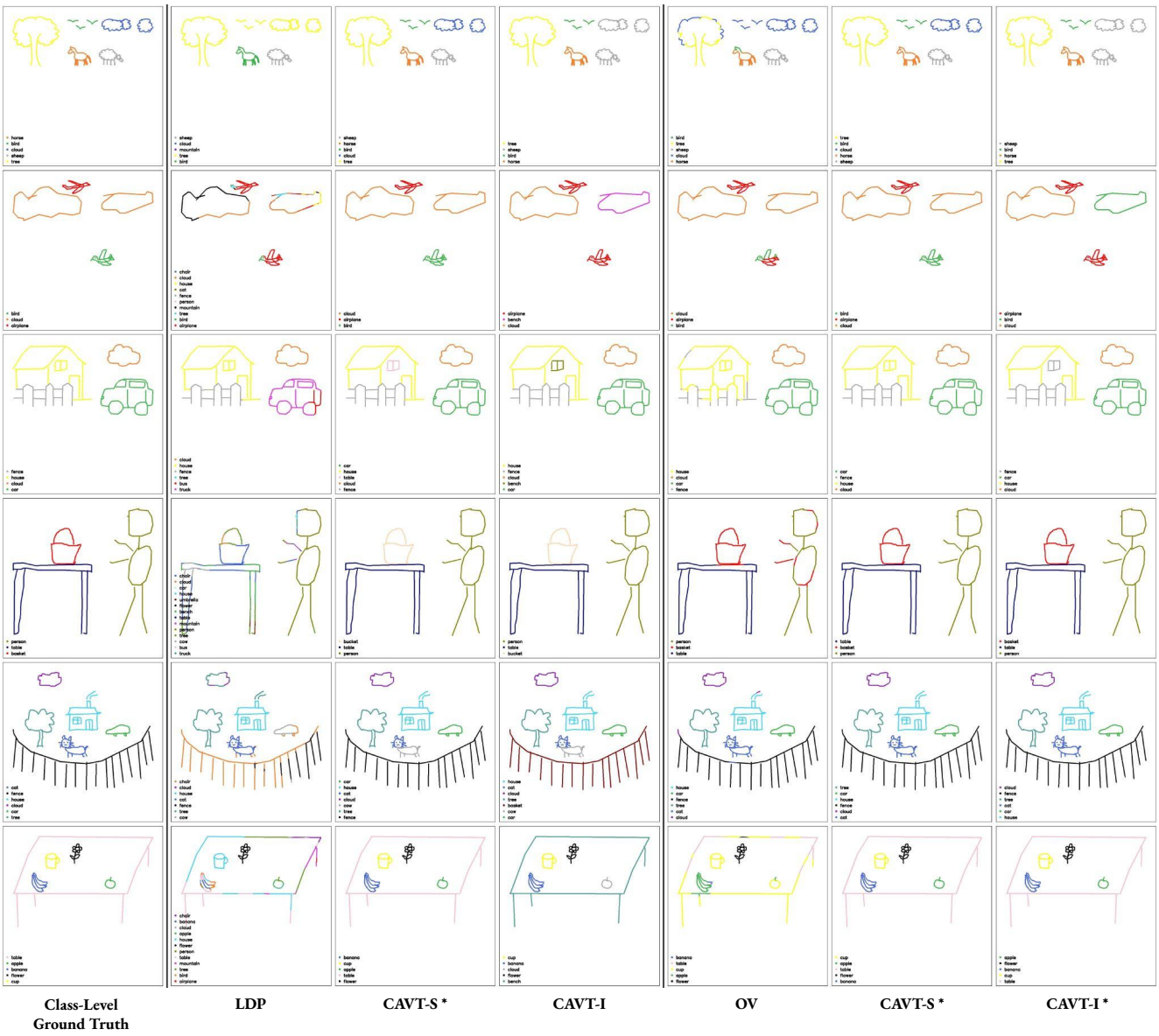

Figure S3. Visual comparison of our method with LDP [3] and OV [1] models, tested on CBSC dataset. We utilize CAVT with the external classifier Sketchformer [9] (CAVT-S) and our pre-trained Inception-V3 [12] (CAVT-I) in an end-to-end manner.

reference. Each participant sequentially draws 10 distinct scene sketches by referring to the corresponding descriptions. Upon completing the sketch collection phase, participants proceed to the second phase, where they annotate their previously drawn sketches.

During the annotation phase, depicted in Figure S6, selected strokes turn from *'gray'* to *'black'* and participants assign a category to each stroke that turns into *'black'*. The annotation process continues until each object instance within the scene is labeled (i.e., each stroke turns into *'black'*). In the process of assigning categories, participants have the option to select from a predetermined list or introduce new categories by entering them into a designated text box (see Figure S6). The predetermined list includes all QuickDraw [4] classes and additional well-known categories not included in QuickDraw but likely to be sketched by participants (e.g., balloon, plate, carpet). This list is provided to ease the labeling process. Finally, strokes that are labeled as incompletely sketched or unrecognizable are marked as *'incomplete'* and excluded from the dataset. Upon acceptance, we will release our data collection web application to the public.

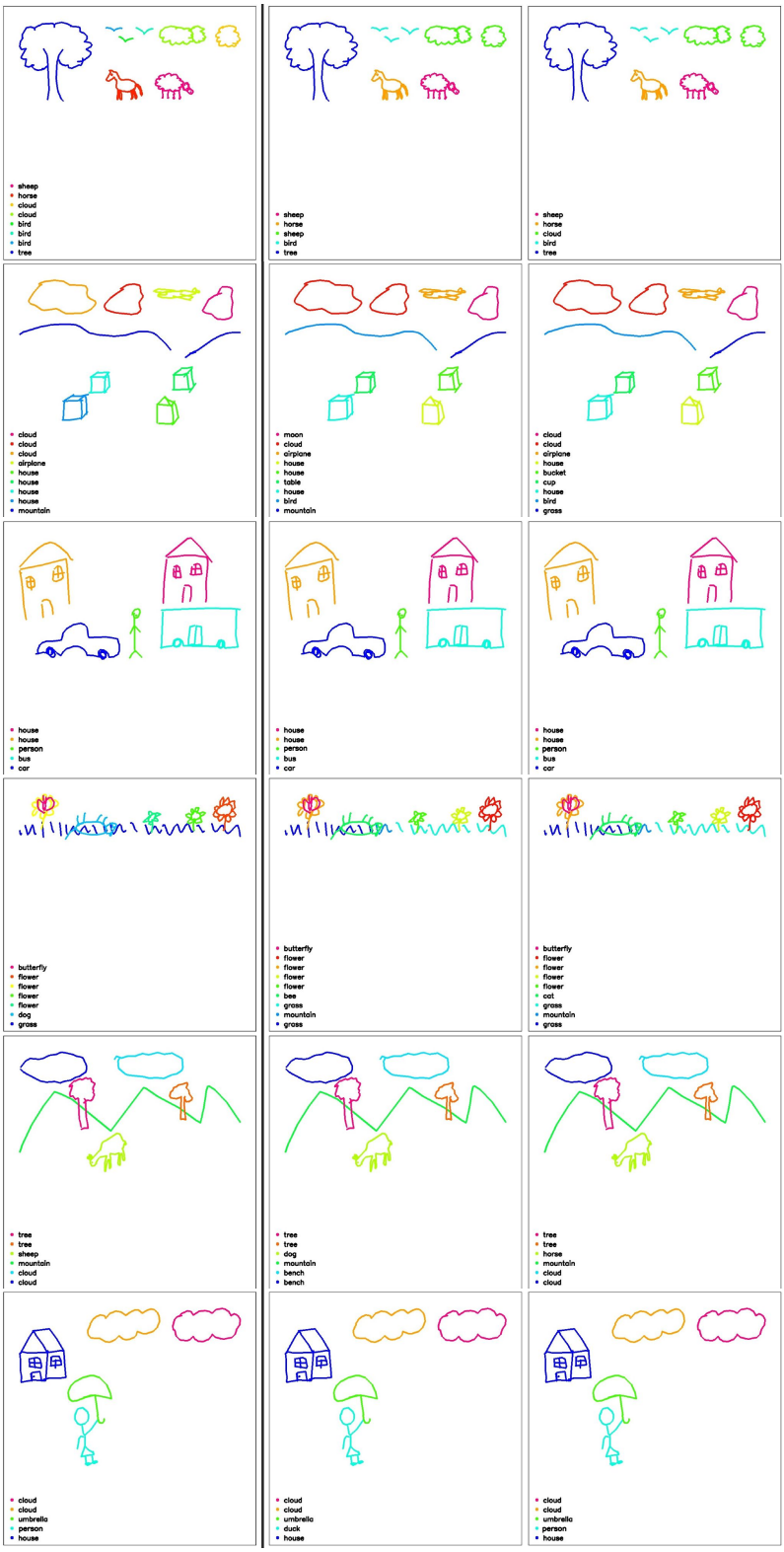

Figure S4. Instance-level visual results of CAVT in FrISS and CBSC datasets combined.

Context	Scene Description	Expected Objects	COCO Img Id
bathroom	In the bathroom, there is a toilet, a bathtub, and a hair dryer.	toilet, bathtub, hair dryer	-
beach	A group of people stand on the beach and fly a kite.	person, kite, beach	92478
outdoor	A girl is standing next to a stop sign with an umbrella in her hand.	person, umbrella, stop sign	-
garden	Four sheep are eating grass, and a child is approaching them.	person, sheep, grass	-
laboratory	A computer workstation with a printer, computer, mouse, and keyboards.	printer, computer, mouse, keyboard	102609
park	A skateboarder with a hat is riding his skateboard to walk his dog.	person, skateboard, dog, hat	304173
living room	A child eats ice cream and his eyeglasses fall on the carpet.	person, ice cream, carpet, eyeglasses	-
hospital	A doctor is holding a syringe and test tube.	person, syringe, test tube, bed	-

Table S4. Sample scene descriptions paired with the expected objects to be drawn by participants during the drawing phase of FrISS. The corresponding real-life image id is provided if the textual description is taken from the MS COCO dataset [8].

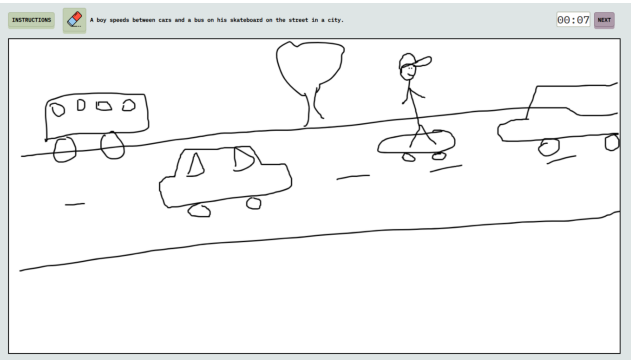

Figure S5. The screenshot from the UI of the data collection web application during the drawing phase

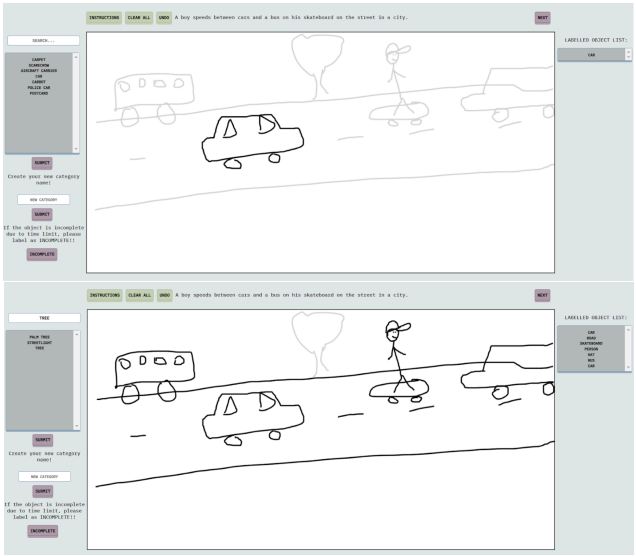

Figure S6. The screenshot of data collection UI during the annotation phase. The upper image is taken while labeling the strokes corresponding to the initial object, *'car'*. The lower image is taken before labeling the final drawn object, *'tree'*. Annotated object classes are listed in the upper-right corner of the UI, in the order of labeling.

S5.2. Visual Comparison of FrISS to Other Datasets

In Sec. 4.3 of the main document, Table 2 provides a statistical comparison of various scene sketch datasets, focusing on category, object, and stroke counts per sketch. Among these datasets provided in Table 2, CBSC [14], FS-COCO [2], and SFSD [15] contain free-hand scene sketches stored in vector format. In Figure S8, we provide a detailed visual comparison between FrISS and these datasets. However, we could only share the visual comparisons between CBSC and FS-COCO, as SFSD is not publicly available. Additionally, we include extra sample scene sketches from FrISS along with their corresponding textual scene descriptions in Figure S7.

CBSC [14] and FS-COCO [2] are collected under similar conditions: participants are permitted multiple drawing attempts, with an average completion time of 3 minutes per scene. In contrast, we imposed a drawing time limit of 1.5 minutes for each scene in our dataset, allowing redraw attempts only within this constrained timeframe, without permitting complete redraws. As depicted in Figure S8, our free-hand scene sketches exhibit significantly fewer strokes per object compared to those in FS-COCO. Furthermore, in the creation of FS-COCO, participants were presented with natural images as references during the drawing process. This results in scene sketches with similar object positions and postures as those in the referenced images. Conversely, the CBSC dataset was collected by instructing participants to quickly draw simple scene sketches that convey semantic meaning to humans, without any time restrictions. Our scene sketches demonstrate comparable object complexities to those in CBSC. However, while CBSC comprises 331 scene sketches covering 74 object categories, FrISS consists of 1K free-hand scene sketches, spanning a broader spectrum of object categories, totaling 403.

S5.3. Details of Textual Scene Descriptions

Scene descriptions are sourced either from the MS COCO dataset image captions [8] or manually created by

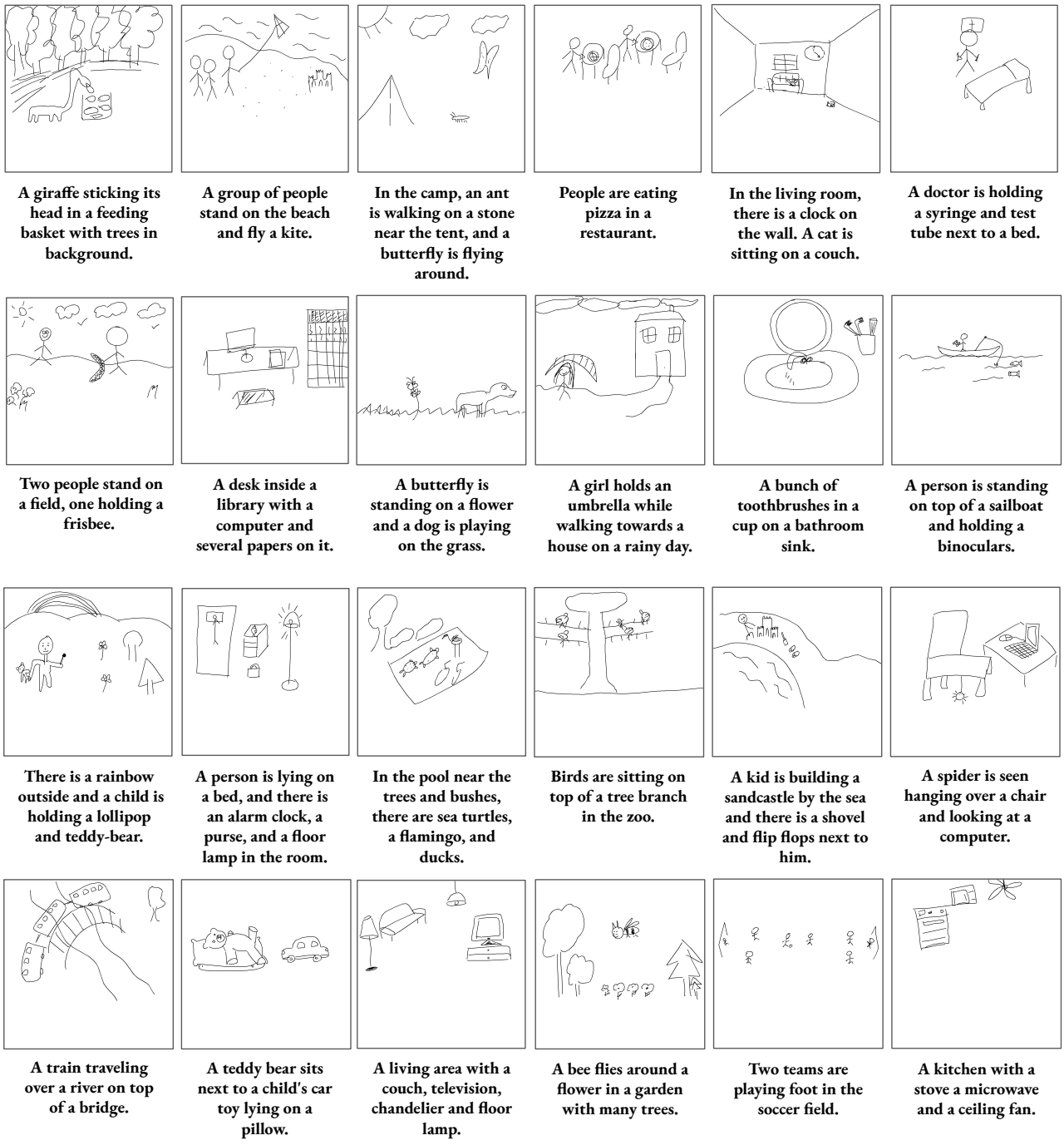

Figure S7. Sample scene sketches from our FrISS dataset paired with their textual scene descriptions

us. Relying solely on MS COCO captions was insufficient to cover a wider range of object categories due to the dataset's limited variety. To ensure a broader representation, we aimed to include descriptions with at least three objects per scene, making sure the prompts were simple and drawable by individuals without professional drawing skills.

To increase scene variety, most of the descriptions were manually constructed. We first gathered a list of environments likely to contain everyday objects. Then, we constructed scene descriptions featuring approximately 3 to 5 objects, ensuring they could be easily drawn within a specified time limit. In total, 180 unique scene descriptions were

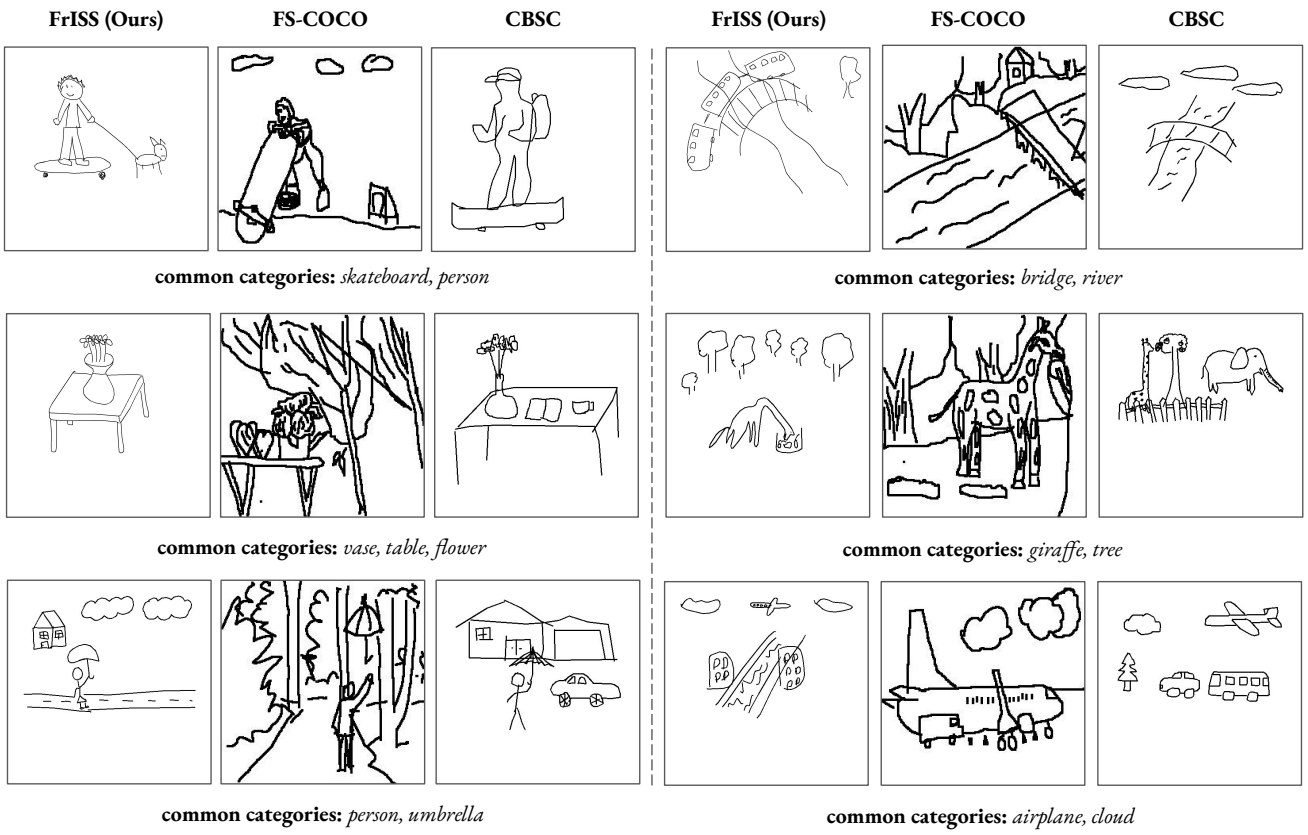

Figure S8. A comparison of scene sketches from FrISS with those from FS-COCO [2] and CBSC [14]. The visuals are selected to ensure that each set of scene sketches shares at least two object categories in common, with the common classes listed below each group of three.

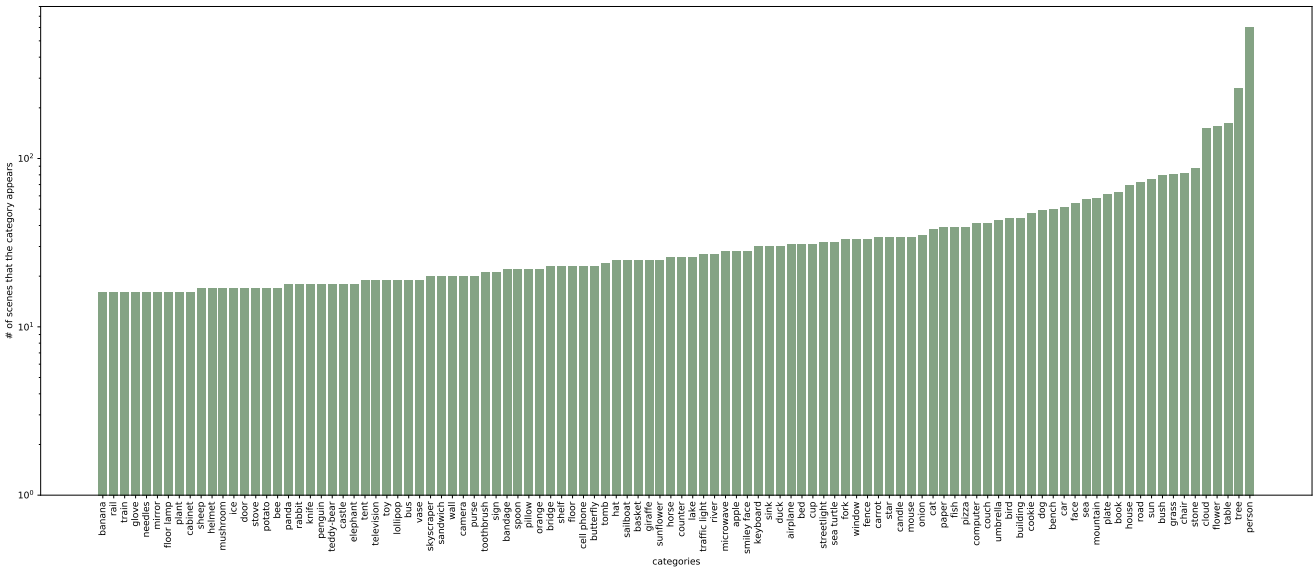

Figure S9. The visualization of the number of scene sketches that each object category appears in. For visualization purposes, we selected the categories that have more than 15 appearances in the FrISS dataset.

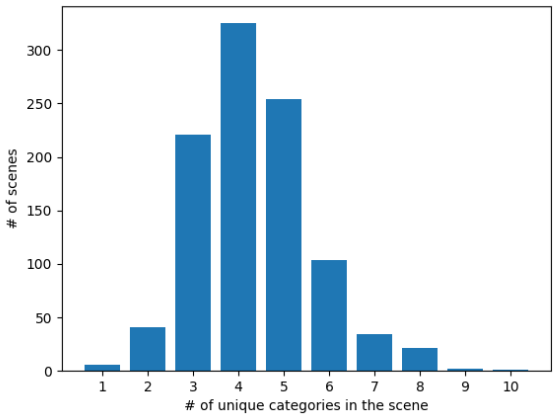

Figure S10. The visualization of the number of unique object categories per scene in the FrISS dataset

created, covering 403 object categories in FrISS. Table S4 presents a subset of our scene descriptions along with their environments. The list of contexts is as follows: *beach, zoo, sky, living room, ocean, kitchen, military base, stadium, concert hall, river, airport, hospital, jungle, graveyard, laboratory, camping site, restaurant, garden, gym, bedroom, gas station, battlefield, library, tower, school, cave, police station, space, museum, hotel, court, farm, hairdresser, park, bathroom, business center, music store, outdoor*.

S5.4. Detailed Analysis of FrISS

Here, we provide additional analysis on our collected dataset in Figures S10 and S9. In Figure S10, we observe that the count of distinct object categories within a scene varies between 1 and 10, with a dominant accumulation between 3 and 6. Additionally, Figure S9 reveals that the most frequently occurring object categories in FrISS are person, tree, table, flower, and cloud, with the remaining categories distributed more balanced throughout the dataset.

List of Categories in FrISS: airplane, alarm clock, ambulance, ant, apple, arm, asparagus, axe, backpack, banana, bandage, barn, baseball, baseball bat, basket, basketball, bathtub, beach, bear, bed, bee, belt, bench, bicycle, binoculars, bird, birthday cake, blackberry, book, boomerang, bowtie, bracelet, bread, bridge, broom, bucket, bus, bush, butterfly, cake, calendar, camera, campfire, candle, cannon, canoe, car, carrot, castle, cat, ceiling fan, cell phone, cello, chair, chandelier, clarinet, clock, cloud, coffee cup, compass, computer, cookie, cooler, couch, cow, crab, crayon, crown, cruise ship, cup, dishwasher, dog, dolphin, donut, door, dresser, drill, drums, duck, dumbbell, elephant, eraser, eyeglasses, face, fan, fence, fire hydrant, fireplace, fish, flamingo, flashlight, flip flops, floor lamp, flower, fork, garden, giraffe, grapes, grass, guitar, hammer, hand, harp, hat, headphones, helmet, horse, hot air balloon, hot dog, hourglass, house, ice cream, key, keyboard, knife, ladder, laptop, leaf, light bulb, lighter, lighthouse, lightning, lion, lollipop, mailbox, map, microphone, microwave, moon, motorbike, mountain, mouse, mug, mushroom, necklace, ocean, octopus, onion, oven, palm tree, panda, pants, paper clip, pear, peas, pencil, penguin, picture frame, pig, pillow, pizza, police car, pond, pool, popsicle, potato, purse, rabbit, radio, rain, rainbow, rake, remote control, rhinoceros, river, sailboat, sandwich, saw, saxophone, school bus, scissors, screwdriver, sea turtle, see saw, shark, sheep, shoe, shovel, sink, skateboard, skull, skyscraper, sleeping bag, smiley face, snake, snorkel, snowflake, snowman, soccer ball, sock, spider, spoon, squirrel, stairs, star, steak, stereo, stop sign, stove, strawberry, streetlight, string bean, submarine, suitcase, sun, swan, swing set, syringe, t-shirt, table, teddy-bear, telephone, television, tennis racquet, tent, toilet, toothbrush, toothpaste, tractor, traffic light, train, tree, truck, trumpet, umbrella, vase, washing machine, watermelon, waterslide, wheel, windmill, wine bottle, wine glass, wristwatch, yoga*, zebra, anchor, bag, ball, balloon, barrier, baseball field, basketball hoop, bee nest, bell, billboard, board, bone, bottle, bowl, box, branch, building, button, cabinet, cable, cage, candy, carpet, cave, ceiling, cheese, chicken, cockroach, coconut, computer case, container, coral, counter, crosswalk, cupboard, curly hair, curtain, dagger, desk, dirt, dog collar, drain, drawer, earth, egg, exhibition, field, fish tank, fishing net, fishing rod, flag, floor, football field, footprint, fridge, frisbee, gas pump, gas station, glass, glass shard, glove, goal, gun, hair, hair dryer, hair tie, hammock, handcuffs, hanger, heart, hook, ice, jellyfish, kite, lake, lamp, light effect, marshmallow, meat, mirror, monitor, moon crater, mousepad, mud, museum, music note, necktie, needles, net, notebook, notes, orange, paddle, paper, path, pathway, peach, pepper, phone box, picnic rug, pipe, plant, plate, plug, present, printer, propeller, rail, restaurant, ribbon, road, rocket, roof, room, rope, ruler, safe, salt, sand, sandcastle, sausage, scarecrow, scarf, sea, sea fish, sea goggles, sea horse, sea shell, seagull, serum, shelf, shower head, sidewalk, sign, slide, smoke, soccer field, speaker, spider web, stage, stage lights, stand, staple, station, stick, stone, stool, strainer, street, suit, sunflower, sunglasses, surfboard, swim goggles, tape player, tennis court, test tube, toilet paper, tomb, tower, toy, traffic cone, trash bin, tray, tribune, turnstile, wall, walnut, water, weapon, wind, window, wing, wood.

Please note that in the FrISS dataset, *yoga** denotes the *person* class. This mapping between the two classes is due to their visual similarity.

S5.5. Common Categories of FrISS and Other Datasets

- **List of common categories between FrISS and SKY-Scene [3]:** airplane, apple, banana, bee, bench, bicycle, bird, butterfly, car, cat, chair, couch, cow, cup, dog, duck, flower, horse, house, mountain, pig, rabbit, sheep, strawberry, table, tree, truck, umbrella, wine bottle.
- **List of common categories between FrISS and SketchyScene [16]:** airplane, apple, banana, basket, bee, bench, bicycle, bird, bucket, bus, butterfly, car, cat, chair, cloud, couch, cow, cup, dog, duck, fence, flower, grass, horse, house, moon, mountain, pig, rabbit, sheep, star, streetlight, sun, table, tree, truck, umbrella, person.
- **List of common categories between FrISS and QuickDraw [4]:** airplane, helicopter, alarm clock, clock, wristwatch, ambulance, firetruck, pickup truck, truck, leaf, van, apple, asparagus, onion, peas, potato, string bean, mushroom, backpack, banana, house, baseball, basketball, soccer ball, baseball bat, bear, panda, bed, bench, bicycle, bird, parrot, birthday cake, cake, blackberry, blueberry, grapes, pear, pineapple, strawberry, watermelon, book, bread, peanut, steak, bridge, broccoli, bus, school bus, bush, canoe, cruise ship, sailboat, speedboat, car, police car, carrot, cat, cell phone, chair, church, hospital, castle, cloud, coffee cup, cup, mug, computer, laptop, cooler, couch, cow, dog, donut, cookie, door, dresser, elephant, fence, fire hydrant, floor lamp, lantern, light bulb, flashlight, flower, fork, giraffe, hamburger, sandwich, horse, hot dog, house plant, jail, keyboard, knife, microwave, motorbike, mountain, mouse, ocean, oven, stove, dishwasher, washing machine, pillow, pizza, purse, rain, remote control, scissors, sheep, sink, skateboard, skyscraper, spoon, stairs, stop sign, suitcase, backpack, table, teddy-bear, television, tennis racquet, tent, toaster, toilet, toothbrush, traffic light, train, umbrella, vase, boomerang, basket, table, wine bottle, wine glass, person, zebra, stop sign, streetlight, hat, helmet, shoe, flip flops, eyeglasses, table, chandelier, ceiling fan, t-shirt, pants, dresser, pencil, eraser, grass, mountain, fence, river, sun, moon, star, snowflake, tree, palm tree
- **List of common categories between FrISS and CBSC [14]:** candle, bus, backpack, keyboard, car, camera, clock, mug, television, truck, banana, couch, elephant, flower, oven, pillow, cow, helmet, sheep, bridge, bench, table, spoon, horse, sandwich, bread, ladder, skateboard, tree, suitcase, bed, giraffe, house, fence, train, laptop, hat, bird, zebra, eyeglasses, fork, carrot, toilet, cat, person, airplane, baseball, bicycle, computer, basket, tent, stairs, chair, cell phone, river, cloud, knife, vase, umbrella, leaf, mountain, pizza, bucket, bear, cup, dog, bush, apple, key, cake, book, mouse, ocean.
- **List of common categories between FrISS and FS-COCO [2]:** cloud, orange, cow, net, hot dog, car, couch, laptop, frisbee, road, chair, wine glass, roof, bed, horse, fork, knife, pizza, bird, river, sandwich, fire hydrant, floor, banana, apple, counter, backpack, bear, plate, mud, toothbrush, shoe, cup, airplane, umbrella, mountain, book, scissors, window, donut, bush, spoon, stairs, keyboard, vase, grass, wood, fence, bottle, kite, plant, mirror, traffic light, cat, door, oven, dog, truck, bus, zebra, toilet, bridge, skateboard, bench, table, dirt, bicycle, cage, giraffe, tent, tree, cake, picnic rug, bowl, stop sign, branch, house, sand, elephant, clock, cell phone, paper, skyscraper, baseball bat, carrot, suitcase, field, train, stone, sheep, surfboard, flower, hat, sea, person, tennis racquet.

S5.6. Ethical Considerations in Data Collection

Our dataset contains free-hand scene sketches paired with their textual descriptions, audio clips of participants, and video recordings of drawing processes. During the drawing process, participants were asked to verbally explain their sketches in their native languages. At the beginning of the data collection, participants received detailed information regarding the following: the recording of their drawing screen in video format, the retention of their verbal descriptions as audio clips, and the potential release of their data in a research paper. Each participant was kindly requested to review and sign the consent form acknowledging our data collection procedures:

'I confirm that I have thoroughly read and understood the instructions. I hereby authorize the utilization of my anonymized data (i.e., drawings, video, and audio recordings) for scientific research purposes.'

Participants who consented to our data collection terms were assigned a random ID and proceeded with the data collection process. Additionally, we provided a contact address to allow participants to confidentially address any concerns regarding the release of their data.

References

[1] Ahmed Bourouis, Judith Ellen Fan, and Yulia Gryaditskaya. Open vocabulary semantic scene sketch understanding. *arXiv preprint arXiv:2312.12463*, 2023. 3, 4, 5

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

coco: Towards understanding of freehand sketches of common objects in context. In *European Conference on Computer Vision*, pages 253–270. Springer, 2022. 7, 9, 11

[3] 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*, 31:1447–1461, 2022. 3, 4, 5, 11

[4] David Ha and Douglas Eck. A neural representation of sketch drawings. In *International Conference on Learning Representations*, 2018. 5, 11

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[6] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1314–1324, 2019. 3

[7] Lei Li, Changqing Zou, Youyi Zheng, Qingkun Su, Hongbo Fu, and Chiew-Lan Tai. Sketch-r2cnn: an rnn-rasterization-cnn architecture for vector sketch recognition. *IEEE transactions on visualization and computer graphics*, 27(9):3745–3754, 2020. 3

[8] 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. 7

[9] Leo Sampaio Ferraz Ribeiro, Tu Bui, John Collomosse, and Moacir Ponti. Sketchformer: Transformer-based representation for sketched structure. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14153–14162, 2020. 3, 4, 5

[10] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018. 3

[11] Karen Simonyan. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 3

[12] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 3, 4, 5

[13] Peng Xu, Chaitanya K Joshi, and Xavier Bresson. Multigraph transformer for free-hand sketch recognition. *IEEE Transactions on Neural Networks and Learning Systems*, 33(10):5150–5161, 2021. 3

[14] Jianhui Zhang, Yilan Chen, Lei Li, Hongbo Fu, and Chiew-Lan Tai. Context-based sketch classification. In *Proceedings of the Joint Symposium on Computational Aesthetics and Sketch-Based Interfaces and Modeling and Non-Photorealistic Animation and Rendering*, pages 1–10, 2018. 3, 7, 9, 11

[15] 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*, 39(12):6309–6321, 2023. 7

[16] 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 *ECCV*, pages 438–454. Springer International Publishing, 2018. 11