ContextSeg: Sketch Semantic Segmentation by Querying the Context with Attention

Supplementary Material

7. More Results

Figure 9 presents more segmentation results of our *ContextSeg* from the SPG and CreatureSketch (CS) datasets. Our method is robust to sketches with various levels of detail.

Statistical Analysis. Table 1 in the main paper demonstrates the comparison with three competitive methods, and our *ContextSeg* outperforms these competitors on all three metrics (except for the face category). Specifically, *ContextSeg*, in particular, exhibits an average improvement of 1.1% in Stroke Accuracy (SAcc), 0.4% in Group Accuracy (GAcc), and 1.4% in Component Accuracy (CAcc) over Sketch-Segformer, which stands out as the most effective among alternative methods. This discrepancy can be attributed to Sketch-Segformer's reliance on absolute coordinates encoded within its graph representation, which, unfortunately, struggles to encapsulate essential structural information. Furthermore, the proposed *ContextSeg* demonstrates an average improvement of 1.6% in SAcc, 0.6% in GAcc, and 1.8% in CAcc compared to SketchGNN, which similarly relies on absolute coordinates to represent graph-based sketches. However, like Sketch-Segformer, SketchGNN also struggles to accurately capture the structural information inherent in strokes. Ultimately, the proposed *ContextSeg* showcases significant superiority over SPGSeg, a sequence-based method, with an average outperformance of 17.4% in SAcc, 10% in GAcc, and 18.7% in CAcc. SPGSeg employs sequential encoding of sketches using relative coordinates and stroke point pen states. However, it overlooks the proximity of strokes, contributing to its comparative shortcomings.

7.1. Additional Ablation Study

Positional Encoding. In stroke embedding learning (Sec. 3.1), we have used two additional coordinate channels to augment the stroke information. To understand better its effectiveness, we replaced the 2D coordinates with the popular 2D sinusoidal positional encoding (denoted as 2DPE and Ours-PE for embedding and segmentation). The stroke reconstruction result and the evaluation statistics are shown in Fig. 10 and Tab. 5, respectively, where the reconstructed sketch is blurry, and the segmentation metrics are all significantly inferior to ours.

Distance Field-only Embedding. To further validate the efficacy of the distance field prediction branch, we have trained the embedding network with only the distance field (denoted as DF). The predicted distance field is shown in Fig. 11, where individual strokes are barely recognized. Besides, we have linked the embedding network with our segmentor (denoted as Ours-DF), and report the evaluation metrics in Tab. 5. The results are inferior to Ours and even worse than Ours w/o DF because the sketch instead of a dense and rough approximation is the key to the segmentation task.

Stroke-based Decoding. The design philosophy of the group-based prediction and its effectiveness are discussed in Sec. 3.2 and Sec. 4.2. We further experiment with stroke-based auto-regressive decoding (denoted as Ours-S) since it is more intuitive. Statistics are shown in the Tab. 5, where our method achieved a remarkable 4.4% increase in terms of average SAcc. Besides, group-based prediction is more efficient, e.g., the average inference time on the airplane category is two times faster (0.73s vs. 1.86s).

Group Order in Auto-regressive Decoding. By design, the stroke order serving as the positional encoding in the Transformer encoder does not matter the decoder prediction, however, the group order matters. By default, we empirically use the more intuitive stroke frequency-based descend order. We have tried two alternatives: stroke frequency ascend order (denoted as Ours-InvG), and a random order (denoted as Ours-RanG). The statistics are shown in Tab. 5, where ours archives the best while the other two are on par with ours.

7.2. Invariance Test

Following [31, 34], we also conducted invariance tests in terms of the anti-rotation and anti-offset ability of our approach. The experiments are conducted on four typical categories - Airplane, Calculator, Face, and Icecream, and statistical results are reported in Tab. 6.

Anti-rotation test. Regarding the anti-rotation test, we adopted an identical experimental setup to previous studies. This setup involved the inclusion of seven distinct rotation angles (i.e., $-45°$, $-30°$, $-15°$, $0°$, $+15°$, $+30°$, and $+45°$) added to the entire sketch. The findings presented in Table 6 reveal a trend wherein the performance of both competitive methods declines with increasing rotation angles. Notably, the proposed *ContextSeg* exhibits a superior mean performance and a narrower standard deviation when compared to other models. This outcome underscores the model's exceptional reliability, particularly in handling sketches subjected to significant degrees of rotation.

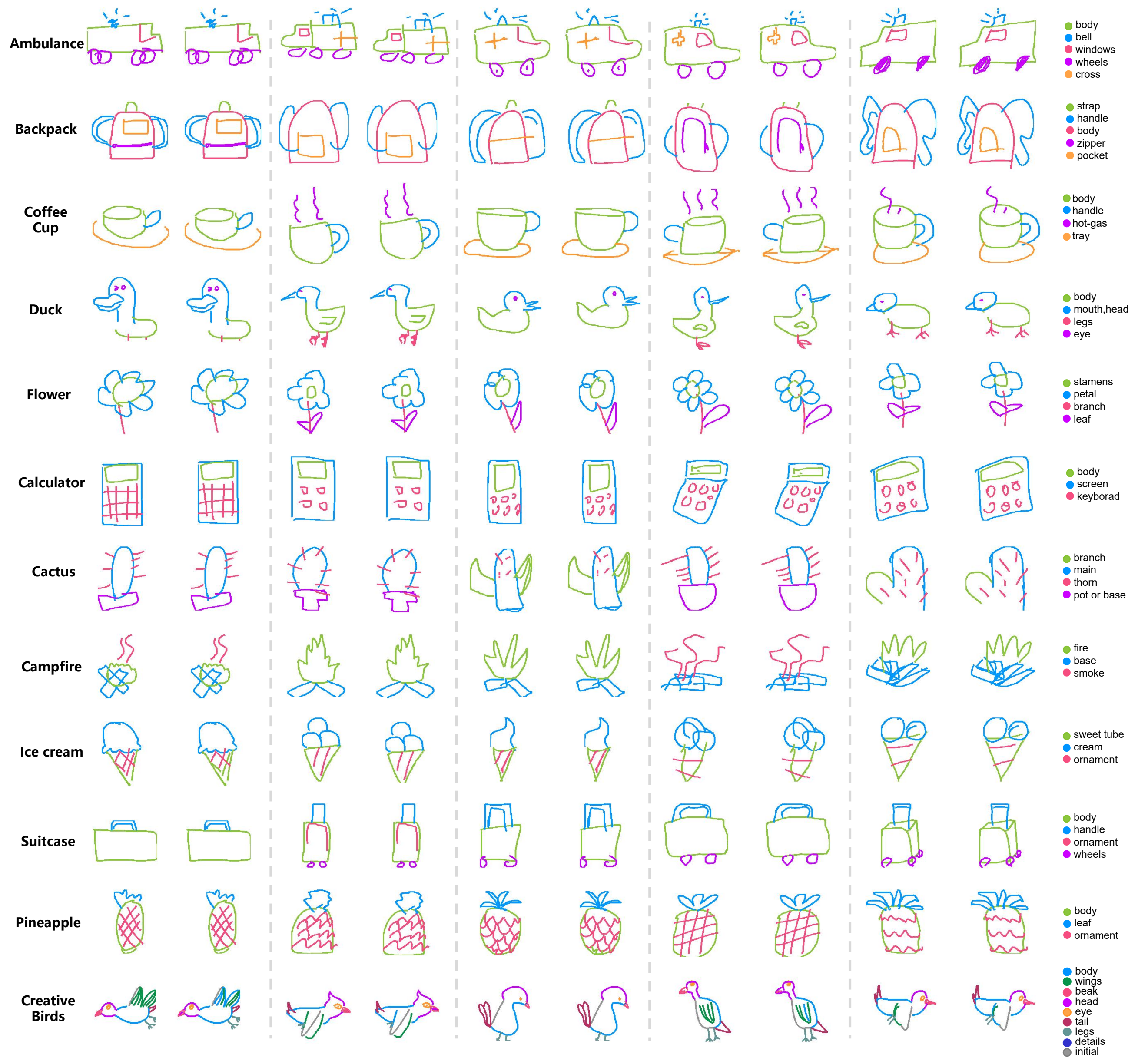


Figure 9. More visual results on benchmarking datasets. For each category, we select five sketch instances, and for each sketch, the ground truth (left) and our segmentation results (right) are displayed.

Table 5. Statistical results of additional ablation studies on the SPG dataset.

Method	Airplane			Calculator			Face			Ice cream			Average		
	SAcc	GAcc	CAcc	SAcc	GAcc	CAcc	SAcc	GAcc	CAcc	SAcc	GAcc	CAcc	SAcc	GAcc	CAcc
Ours-PE	89.3	91.9	85.6	96.1	95.7	96.8	93.4	95.4	93.4	93.1	94.5	91.2	93.0	94.4	91.8
Ours-DF	83.4	86.6	81.3	90.1	91.7	87.2	89.4	89.2	86.3	86.7	89.3	84.2	87.4	89.2	84.8
Ours-S	88.6	91,1	84.7	94.4	95.8	93.6	92.1	94.6	91.7	91.8	92.3	89.6	91.7	93.5	89.9
Ours-InvG	92.6	94.0	88.6	98.6	98.1	97.0	95.9	97.9	93.8	95.2	96.1	91.7	95,6	96.5	92.8
Ours-RanG	92.8	94.1	88.7	98.7	98.2	97.1	96.1	98.0	94.1	95.4	96.2	91.9	95.8	96.8	93.0
Ours	**93.2**	**94.9**	**89.5**	**99.2**	**98.7**	**97.5**	**96.4**	**98.4**	**94.8**	**95.9**	**96.5**	**92.4**	**96.1**	**97.1**	**93.6**

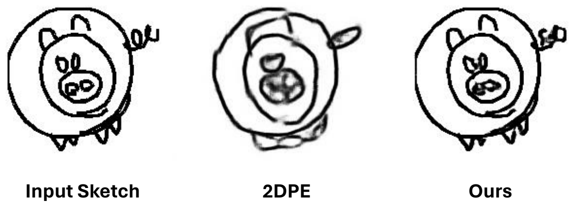


Figure 10. Sketch reconstruction from 2DPE and Ours.

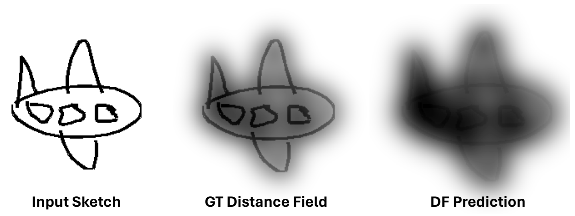
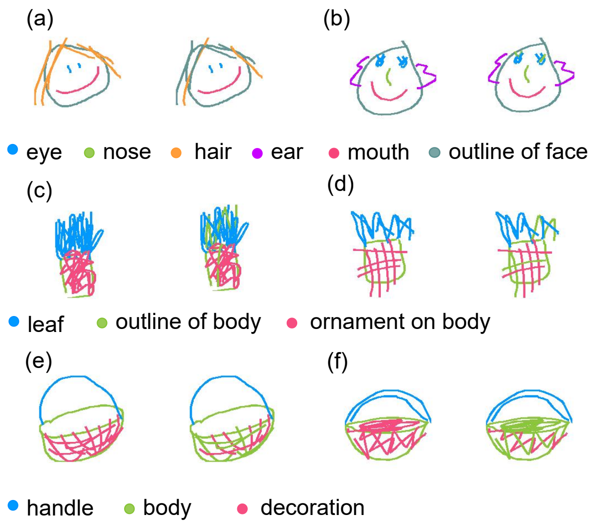


Figure 11. Distance field prediction from DF.

Anti-offset test. For the offset test, given a testing sketch, we first calculate the diagonal length d of the bounding box, and for each stroke, we randomly sample an offset from a uniform distribution - $(\Delta x, \Delta y) \sim \mathcal{N}(0, \sigma^2)$, where σ is set at $0.05d$, $0.1d$, $0.15d$, and $0.20d$, respectively. As expected, our performance drops when increasing the offset distance, but we still obtain superior accuracy than SketchGNN [31] and Sketch-Segformer [34] at each variation, which indicates the excellent robustness of our approach.

The results from both tests strongly suggest that ContextSeg possesses greater robustness, showcasing its ability to sustain segmentation accuracy despite offsets or rotations. This signifies its strength in effectively managing spatial variations within the data.


(a)
(b)
● eye ● nose ● hair ● ear ● mouth ● outline of face
(c)
(d)
● leaf ● outline of body ● ornament on body
(e)
(f)
● handle ● body ● decoration
Figure 12. Exemplar results with imperfect segmentation. For each case, the ground truth (left) and our prediction (right) are shown.

7.3. Failure Cases

Figure 12 demonstrates several failure cases of our method. The imperfection primarily arises from two contributing factors. Firstly, our segmentation Transformer encounters difficulties in accurately labeling strokes within heavily overlapped areas. For instance, in Fig. 12 (a) and (b), distinguishing between hair and eyelash strokes becomes challenging due to their dense concentration, leading to segmentation inaccuracies. Similarly, in Fig. 12 (e) and (f), the intricate decorations on the basket pose challenges as our segmentation model erroneously categorizes them as part of the body. Secondly, our embedding network encounters limitations in encoding strokes characterized by rapid and substantial variations. For instance, in Fig. 12 (c) and (d), the leaves and ornamentation of the pineapple exhibit strokes with rapid fluctuations. Such variations can result in suboptimal embeddings, potentially causing misinterpretations for our segmentation Transformer. Consequently, certain strokes might be inaccurately classified as part of the body.

8. Implementation Detail

All experiments were conducted on a single Nvidia RTX3090 GPU. We implement our networks using Tensorflow, and detailed network structures are shown in Fig. 13.

The embedding network has an encoder-decoder structure, accepting the grayscale sketch input augmented with x and y coordinate channels. Specifically, the encoder comprises 10 layers grouped into four segments, each characterized by distinct feature dimensions (i.e., 64, 128, 256, and 512), resulting in a stroke embedding of $256d$. Both decoder branches share an identical encoder structure, working symmetrically to transform the stroke embedding into sketch reconstruction and the distance map.

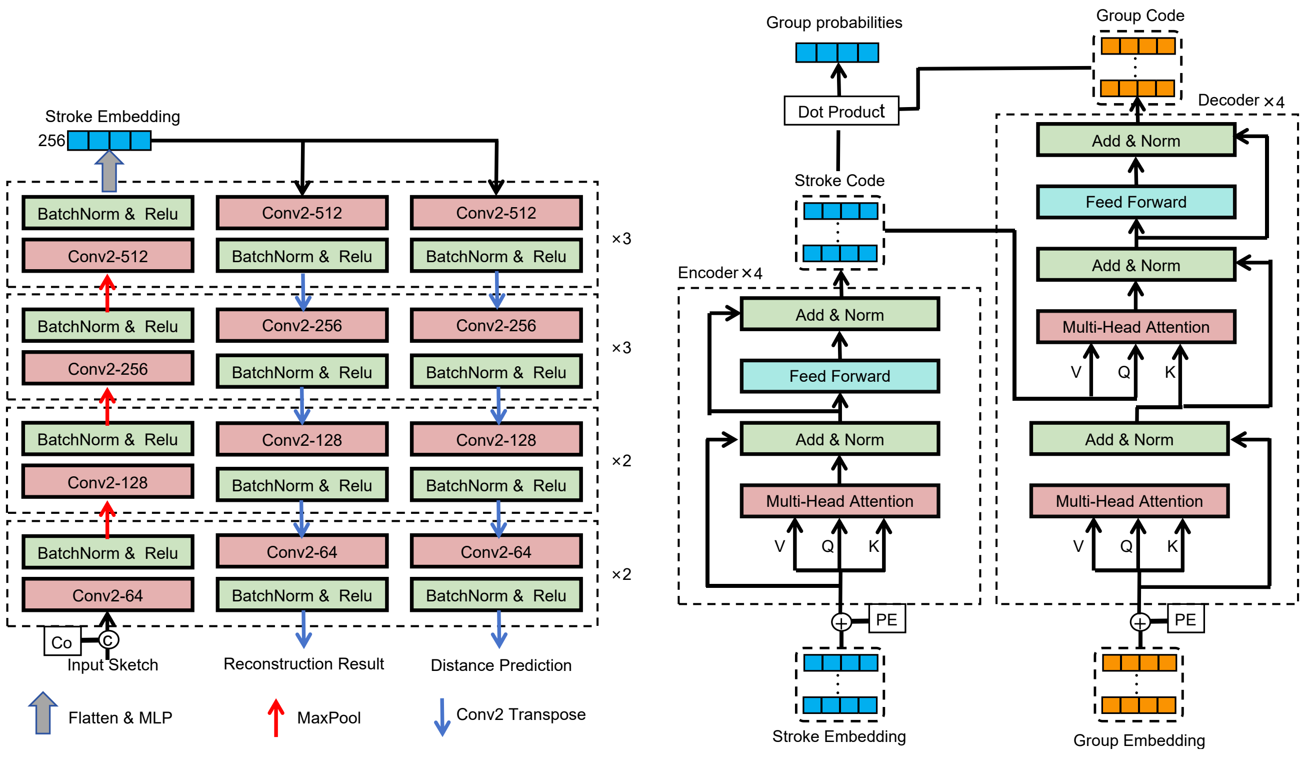
The segmentation Transformer has 4 attention layers in both the encoder and decoder, each layer has 4 attention heads and the dropout rate is 0.4.

Network Training. We first train the embedding network until convergence, which takes around 15 hours with a batch size of 64. Then, the segmentation Transformer is trained until convergence taking around 10 hours with a batch size of 8. Adam [12] optimizer was used in both network training with a fixed learning rate 10^{-4} and other default parameters.

Teacher Forcing Gap. Teacher-forcing is widely used for Transformer training. However, it introduces the exposure bias issue by feeding the ground truth context to the decoder at training time while exploiting the inferior prediction at testing time. To overcome the teacher-forcing gap, in our case, we follow [20] to forward the decoder twice to mix the predicted group of strokes with the ground truth group of strokes. The ratio of the ground truth strokes gradually

Table 6. Statistical comparison of the invariance tests on four representative categories.

	Angle/Distance	**SketchGNN** [31]				**Sketch-Segformer** [34]				Ours			
		Airplane	Calculator	Face	Ice cream	Airplane	Calculator	Face	Ice cream	Airplane	Calculator	Face	Ice cream
Rotation Test	$-45°$	87.7	92.4	90.7	88.4	88.2	93.1	90.2	87.5	**90.1**	**94.9**	**91.7**	**90.8**
	$-30°$	89.4	94.7	93.2	91.3	89.9	95.4	92.1	90.5	**91.0**	**97.2**	**94.1**	**92.1**
	$-15°$	90.4	97.6	95.7	93.4	91.3	98.1	94.8	92.1	**92.6**	**98.5**	**95.9**	**94.3**
	0	91.1	98.4	96.2	94.5	92.4	98.5	97.2	94.4	**93.2**	**99.2**	**96.4**	**95.9**
	$+15°$	90.1	97.1	95.2	92.9	91.1	98.3	94.1	92.7	**92.9**	**98.2**	**95.5**	**94.1**
	$+30°$	89.2	94.3	93.1	90.8	90.0	95.9	92.7	90.1	**91.3**	**97.4**	**93.9**	**92.3**
	$+45°$	87.2	92.1	90.1	87.8	88.5	93.7	89.6	87.1	**89.5**	**95.3**	**91.3**	**90.4**
	Average	89.3	95.2	93.5	91.3	90.2	96.1	93.0	90.6	**91.5**	**97.2**	**94.1**	**92.8**
	Standard Deviation	**1.4**	2.5	2.4	2.5	1.5	2.2	2.7	2.7	**1.4**	**1.6**	**2.0**	**2.0**
Offset Test	0	91.1	98.4	96.2	94.5	92.4	98.5	97.2	94.4	**93.2**	**99.2**	**96.4**	**95.9**
	0.05σ	90.3	96.5	92.4	91.4	90.5	96.7	92.7	91.3	**91.1**	**97.4**	**93.5**	**92.1**
	0.1σ	88.1	93.0	90.4	89.2	88.7	93.4	91.0	88.4	**89.1**	**94.1**	**91.9**	**90.1**
	0.15σ	85.5	89.4	87.1	86.1	85.2	88.7	87.6	85.3	**86.4**	**91.4**	**88.4**	**87.8**
	0.20σ	79.9	85.4	81.0	82.5	78.6	84.2	82.1	81.9	**82.4**	**87.5**	**83.7**	**84.5**
	Average	87.0	92.5	89.4	88.7	87.1	92.3	90.1	88.3	**88.4**	**93.9**	**90.8**	**90.1**
	Standard Deviation	4.5	5.3	5.7	4.6	5.4	5.9	5.7	4.9	**4.2**	**4.7**	**4.9**	**4.3**



(a) Embedding Network. (b) Segmentation Transformer.

Figure 13. Detailed structure of our embedding and segmentation networks.

decreases from 100% to 20% along with the training process.

9. Dataset Details

Evaluation Metrics. The three metrics we used in the paper are defined as:

- **Stroke Accuracy (SAcc)** calculates the percentage of correctly labeled strokes. For a point-based stroke representation, if a minimum of 75% of its points are correctly labeled, then the stroke is correctly labeled.

- **Grouping Accuracy (GAcc)** measures the accuracy of the group-based classification task in our segmentation Transformer. Suppose the ground truth classification labels are stored in a binary matrix $M^{\mathbb{S}\times\mathbb{C}}$, where $\mathbb{S}$ is the total number of strokes and $\mathbb{C}$ is the total number of groups/categories. $M_{i,j} = 1$ if and only if stroke s_i belongs to g_j. The Transformer predicts M' given a sketch s_i, we thus calculate the grouping accuracy as:

$$GAcc = \frac{1}{\mathbb{S}\times\mathbb{C}} \left| M - M' \right|. \tag{7}$$

- **Component Accuracy (CAcc)** measures the percentage of correctly labeled categories. A category is deemed accurately labeled if a minimum of 75% of its strokes receive the correct labels.

Data Augmentation. To enrich the diversity of the dataset and improve the robustness of the trained networks, we apply both stroke-level and sketch-level data augmentations. For the former, we rotate, scale, and add a positional perturbation to one or more strokes within a sketch. While, for the latter, we rotate and scale the sketch image, and randomly discard strokes from sketch images as well.