

BézierSketch: A generative model for scalable vector sketches

Das, Yongxin Yang, Timothy Hospedales, Tao Xiang, Yi-Zhe Song

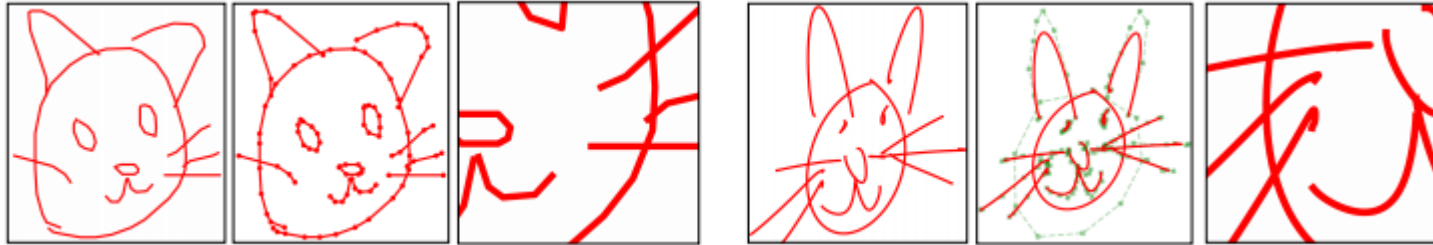
ECCV 2020

Sanghyeon Lee

06 Jan. 2021

Introduction

Task : Generation scalable vector sketches & Translation raster image into vector sketches

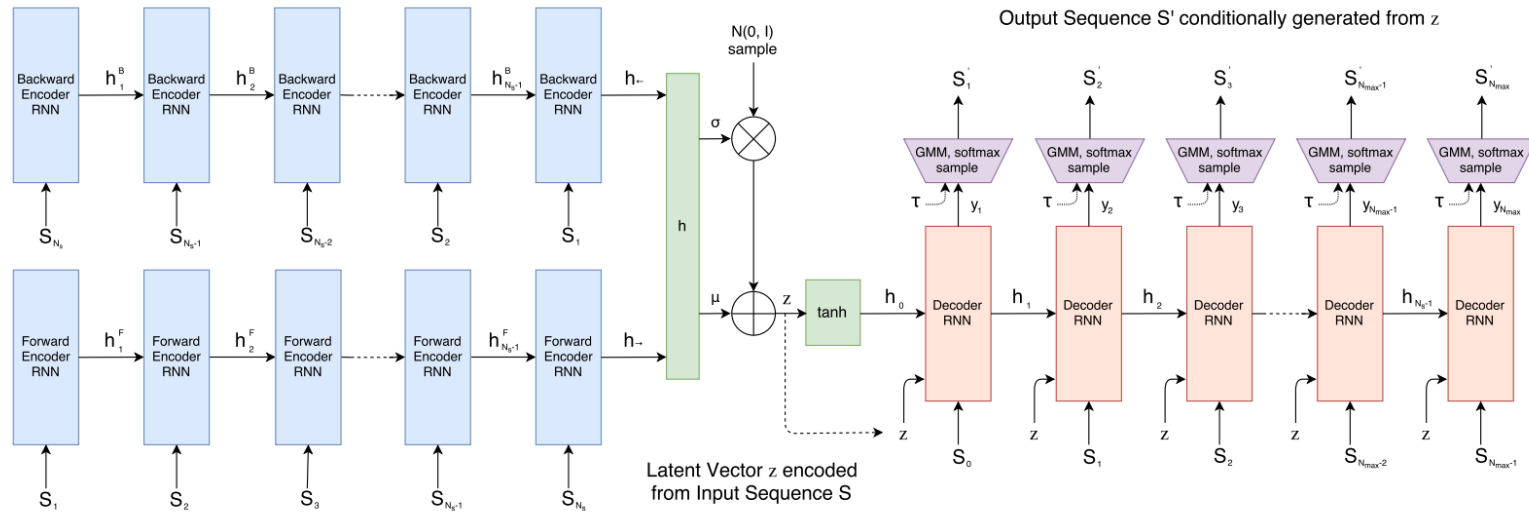


Contribution

- BézierEncoder: Inverse-graphics approach for mapping strokes to parameterized Bézier
- BézierSketch: A sequential generative model for sketches that produces high resolution and low-noise vector graphic samples
- Training model without supervision (Inverse Graphic)

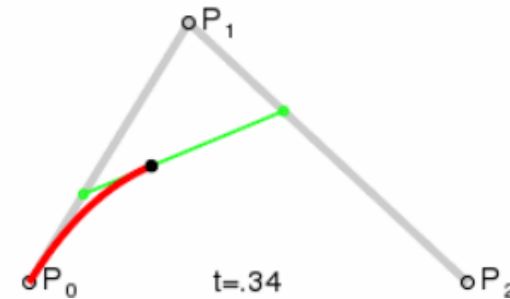
Background

Sketch RNN



Bézier curve

$$C(t; \{P_i\}) = \sum_{i=0}^n B_{i,n}(t) \cdot P_i \quad B_{i,n}(t) \triangleq \binom{n}{i} t^i (1-t)^{n-i}$$



Method

1. Sketch Representation

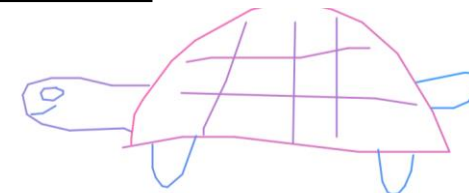
Common format

$$\mathcal{S} = [(\mathbf{X}_i, q_i)]_{i=1}^L \quad \mathbf{X}_i \triangleq [x \ y]_i^T \in \mathbb{R}^2, q_i \in \{\text{PENUP}, \text{PENDOWN}\}$$

Stroke-Level Representation

$$\bar{\mathcal{S}} \triangleq [\mathbf{T}_j]_{j=1}^N, \text{ with } \mathbf{T}_j \triangleq [\mathbf{X}_i^{(j)}]_{i=1}^{N_j}$$

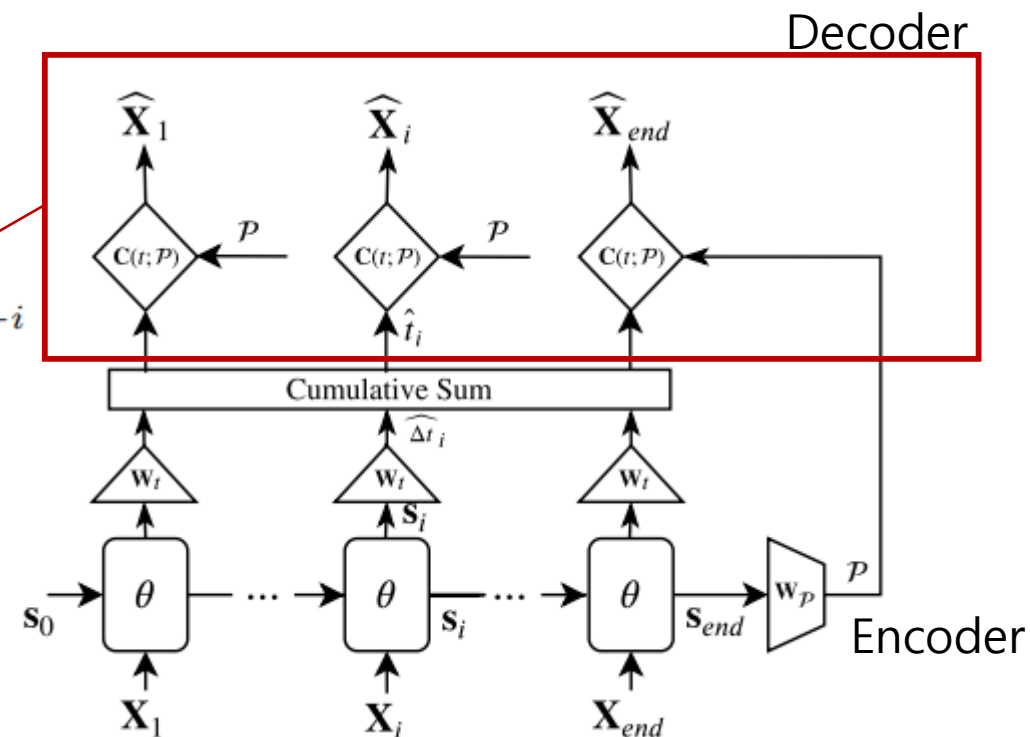
155	-25	1	0	1	141	-117	1	0	1	3	0	1	0	0
124	-5	1	0	0	10	111	0	1	0	6	0	1	0	0
238	-1	1	0	0	129	-98	1	0	1	2	0	1	0	0
229	6	1	0	0	135	0	1	0	0	14	-12	1	0	0
38	-1	1	0	0	130	0	1	0	1	12	-38	1	0	0
133	25	1	0	0	147	0	1	0	0	17	13	1	0	0
33	56	1	0	0	122	6	1	0	0	4	26	1	0	0
144	38	1	0	0	151	10	1	0	0	5	10	1	0	0
-81	2	1	0	0	14	10	1	0	0	4	2	1	0	0
-125	-10	1	0	0	146	8	1	0	1	6	-14	1	0	0
-108	1	1	0	0	149	0	1	0	0	6	-7	1	0	0
-104	18	1	0	0	154	0	1	0	0	2	-14	1	0	0
-105	-45	1	0	0	17	3	1	0	0	2	-15	0	1	0
23	-8	1	0	0	151	-9	1	0	0	22	-44	1	0	0
188	2	1	0	0	122	1	1	0	0	21	0	1	0	0
-15	-9	1	0	0	1	6	1	0	0	14	-4	1	0	0
255	0	1	0	0	14	6	1	0	0	12	-17	1	0	0
-127	43	1	0	0	110	1	1	0	0	-1	-9	1	0	0
142	2	1	0	0	154	-9	1	0	0	-16	-1	1	0	0
136	6	1	0	0	9	-3	1	0	0	-49	0	1	0	0
-110	-71	1	0	0	-9	-2	1	0	0	0	0	1	0	0



2. BézierEncoder: Stroke Embedding

$$C(t; \{\mathbf{P}_i\}) = \sum_{i=0}^n \mathcal{B}_{i,n}(t) \cdot \mathbf{P}_i \quad \mathcal{B}_{i,n}(t) \triangleq \binom{n}{i} t^i (1-t)^{n-i}$$

$$\mathbf{P} \triangleq [P_x \ P_y]^T \in \mathbb{R}^2 : \text{Control points}$$



Method

2. BézierEncoder: Stroke Embedding

$$[\vec{s}_i, \overleftarrow{s}_i] = \text{BiRNN}(\mathbf{X}_{i-1}, \mathbf{s}_{i-1}; \theta)$$

$$\mathcal{P} = \mathbf{W}_{\mathcal{P}} [\vec{s}_{end}; \overleftarrow{s}_{end}]$$

$$\mathbf{W}_{\mathcal{P}} \in \mathbb{R}^{2(n+1) \times 2h}$$

$$\hat{t}_i = \sum_{i'=1}^i \hat{\Delta t}_{i'}, \text{ with } \hat{\Delta t}_i = \text{SOFTMAX}_i(\mathbf{W}_t \cdot [\vec{s}_i; \overleftarrow{s}_i])$$

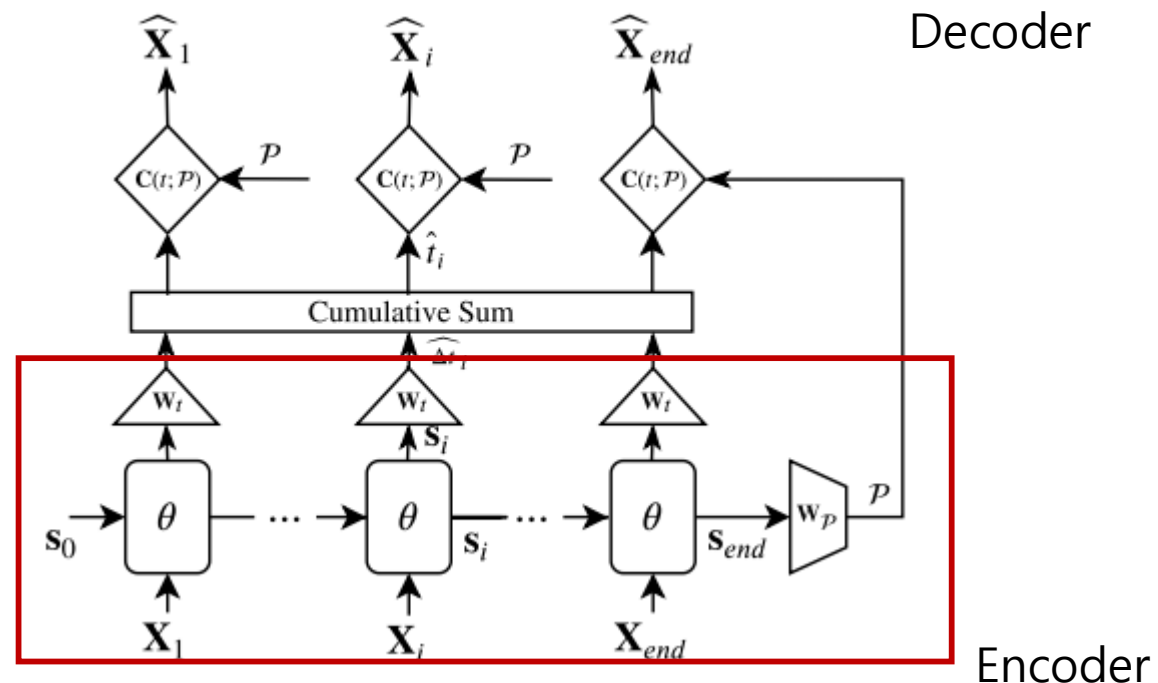
$$\mathcal{L}(\theta, \mathbf{W}_{\mathcal{P}}, \mathbf{W}_t) \triangleq \sum_i \|\mathcal{C}(\hat{t}_i, \mathcal{P}) - \mathbf{X}_i\|^2$$

Multi-Degree Representation Extension

$$\mathcal{L}_{total} \triangleq \sum_{n=n_{min}}^{n_{max}} \mathcal{L}_n, \text{ with } \mathcal{L}_n(\theta, \mathbf{W}_{\mathcal{P}^n}, \mathbf{W}_t^n) \triangleq \sum_i \|\mathcal{C}(\hat{t}_i^n, \mathcal{P}^n) - \mathbf{X}_i\|^2$$

Smoothness Regularizer

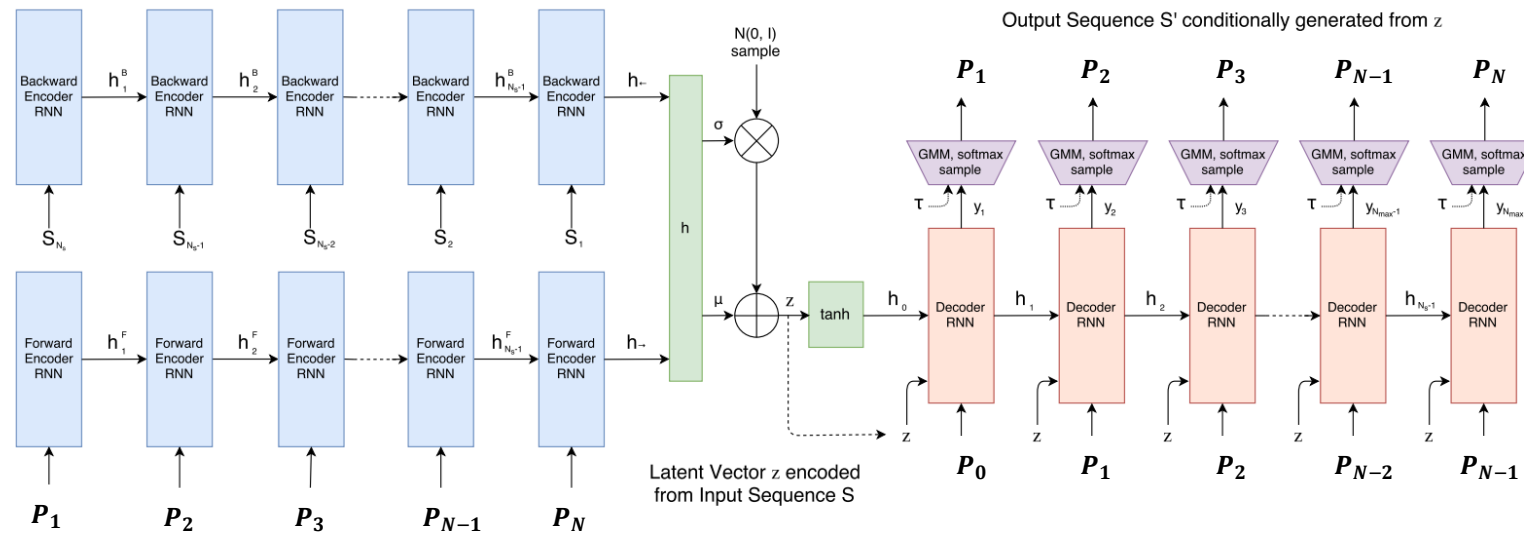
$$\mathcal{R}_n(\mathcal{P}^n) \triangleq \sum_{i=1} \|\mathbf{P}_{i+1} - \mathbf{P}_i\|_2^2$$



Method

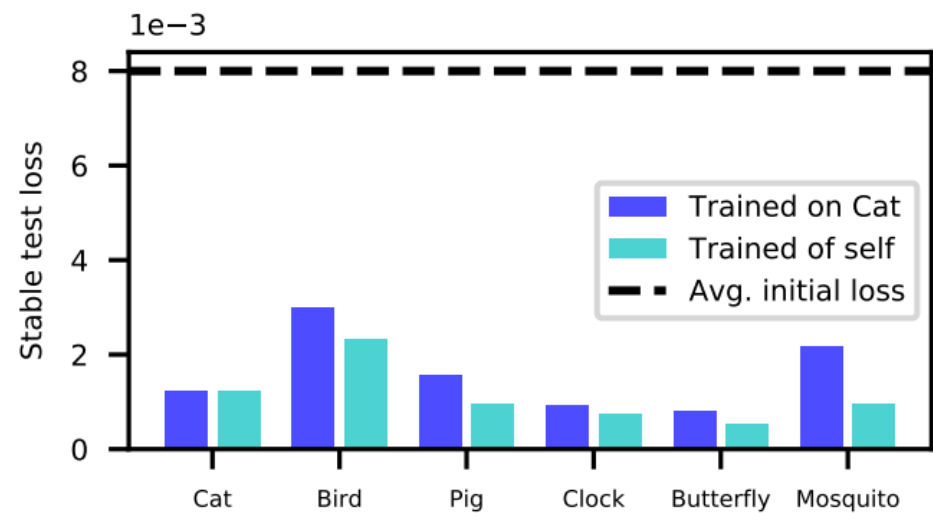
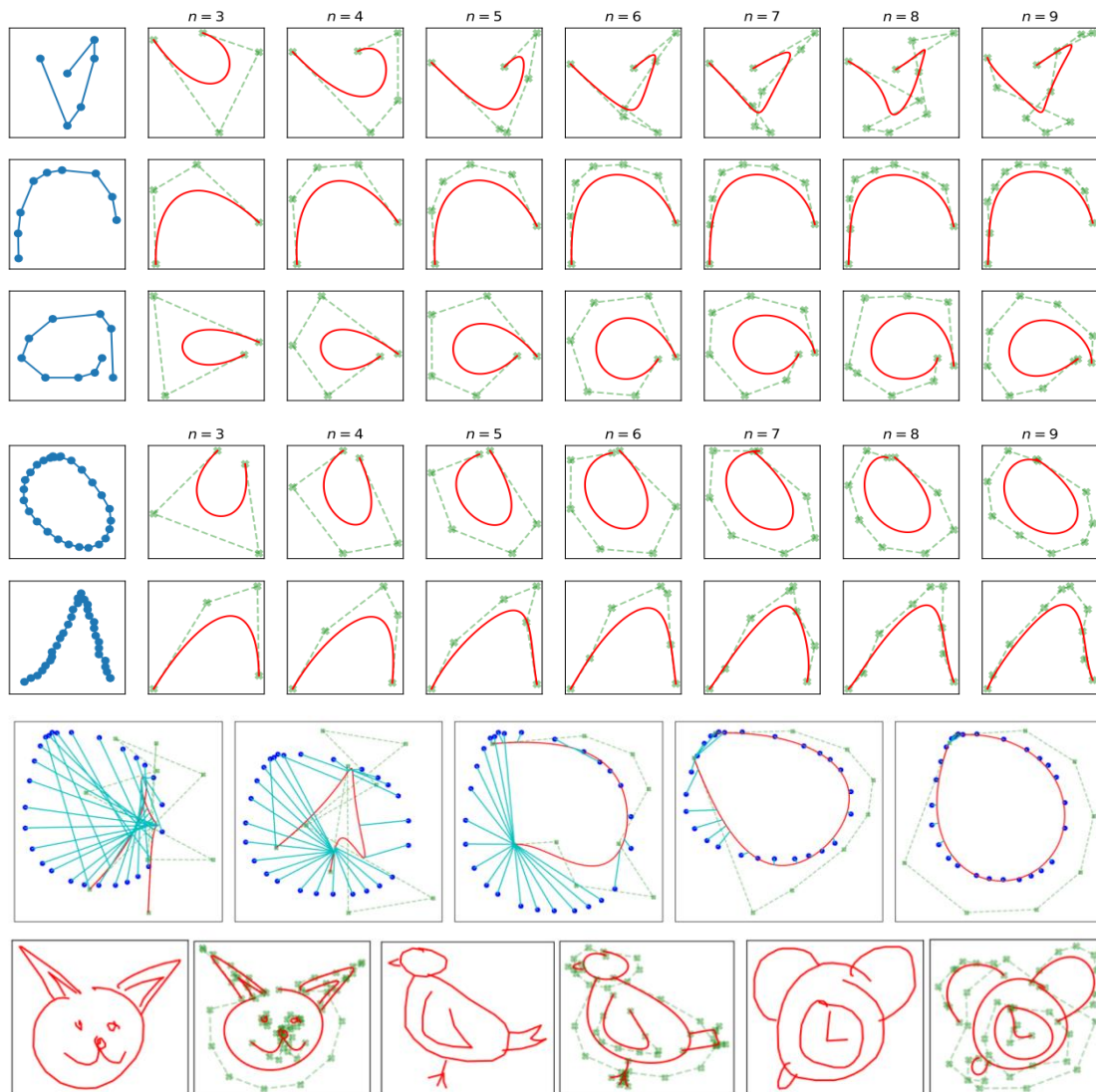
3. BézierSketch: Sketch Generation

- 1) Transform the set of strokes as $S_{st} = \{P_j\}_{j=1}^N$, Where $P_j = e(T_j)$
- 2) Training model via SketchRNN schema



Experiments

1. BézierEncoder



Experiments

2. BézierSketch

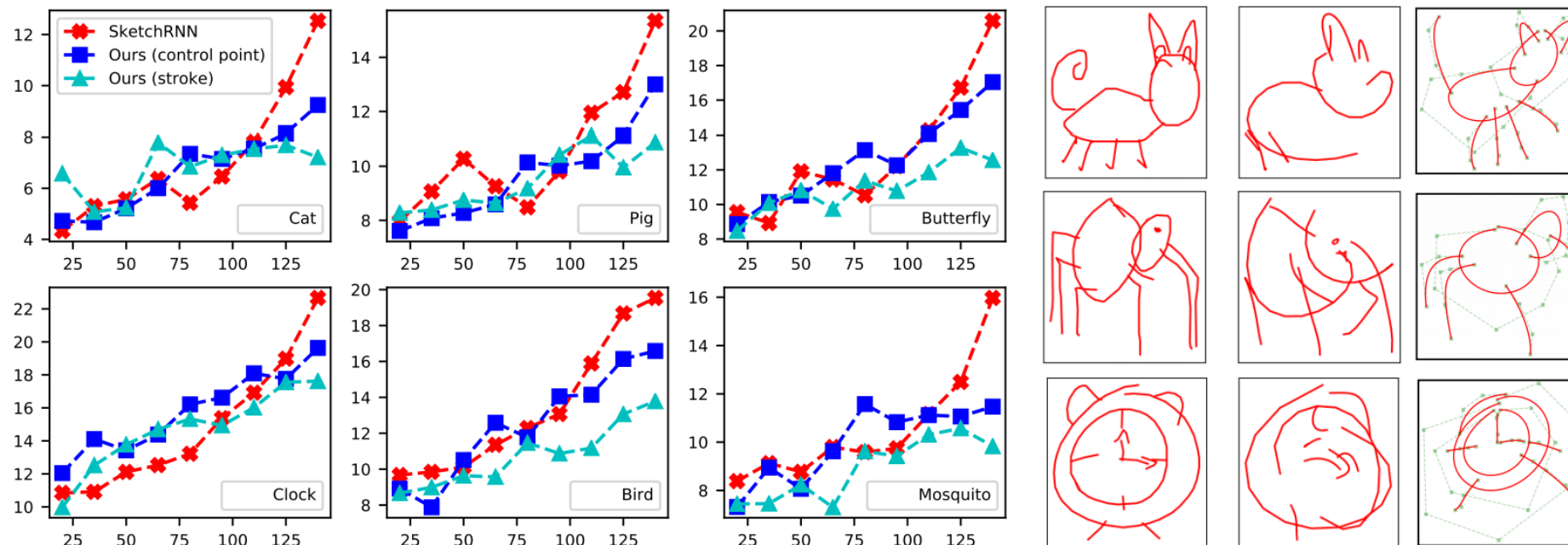
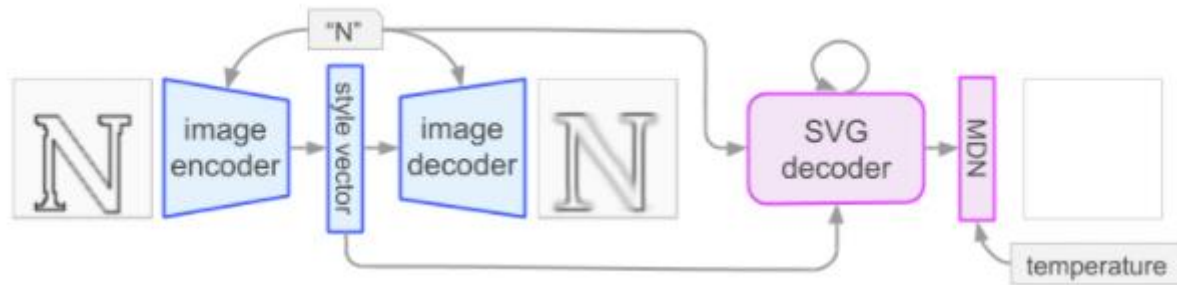


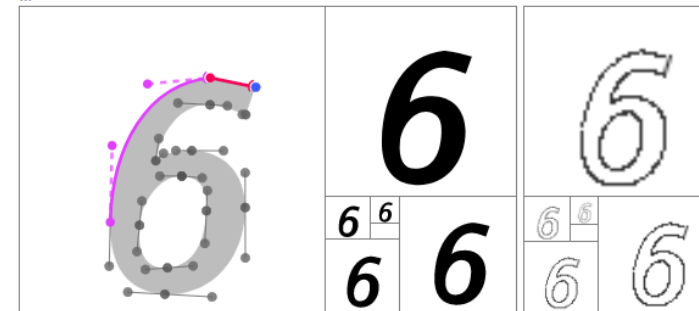
Fig. 7: Left: FID score (\downarrow) vs length of sketch shows the effectiveness of our generative model on longer sketches. Right: Qualitative samples of long sketches. Three columns denote the original sketch, SketchRNN and our BézierSketch.

Other works

A Learned Representation for Scalable Vector Graphics (CVPR2019)



moveTo (15, 25)
lineTo (-2, 0.3)
cubicBezier (-7.4, 0.2) (-14.5, 11.7), (-12.1, 23.4)



Deep Vectorization of Technical Drawings(CVPR2020)

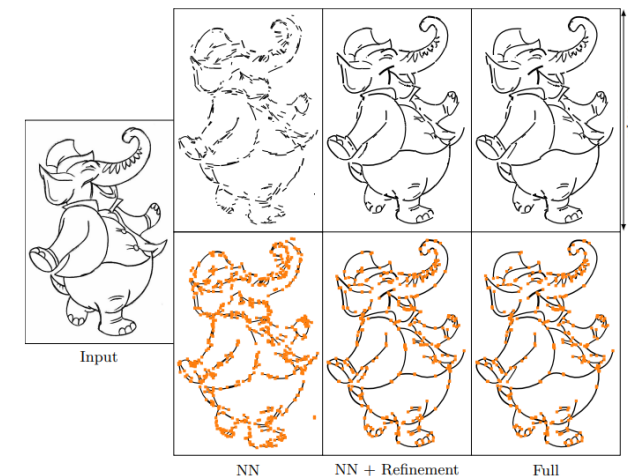
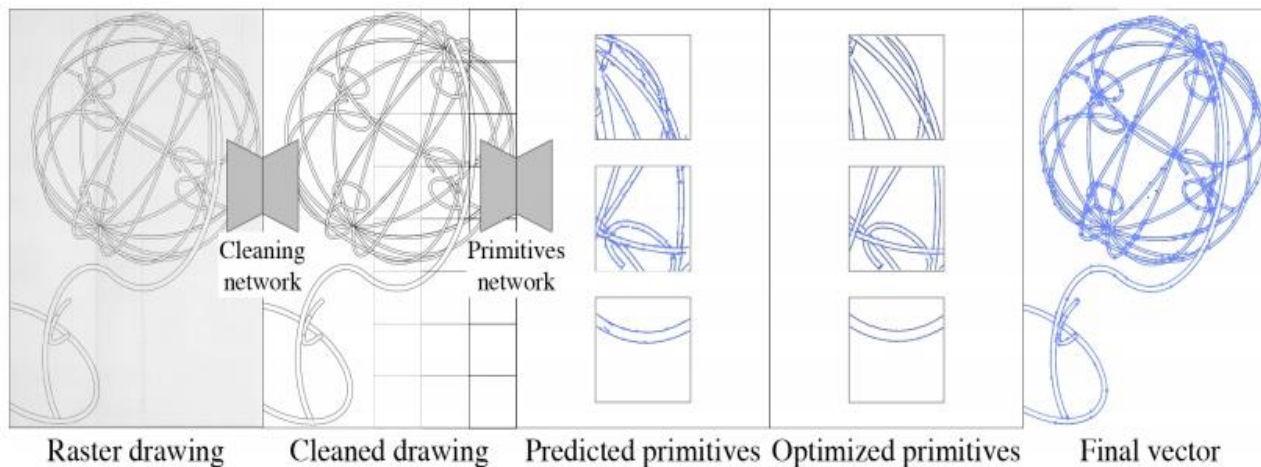


Fig. 16: Qualitative results of our system on clean cartoon drawings. Endpoints of primitives are shown in orange. The input image on the top is copyrighted by David Revoy www.davidrevoy.com under CC-by 4.0 license and on the bottom from www.easy-drawings-and-sketches.com, © Ivan Huska.

Thank you