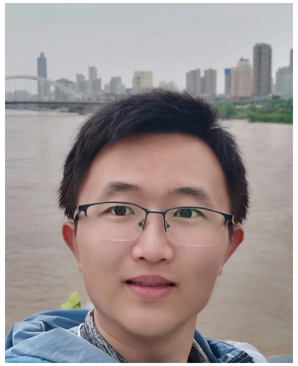
SketchKnitter: Vectorized Sketch Generation with Diffusion Models



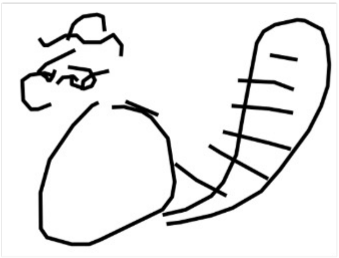
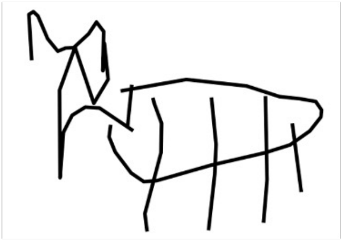


Qiang Wang[1], Haoge Deng[1], <u>Yonggang Qi[1*]</u>, Da Li[2], Yi-Zhe Song[2]

[1]Beijing University of Posts and Telecommunications (BUPT), CN
[2]SketchX, CVSSP, University of Surrey, UK

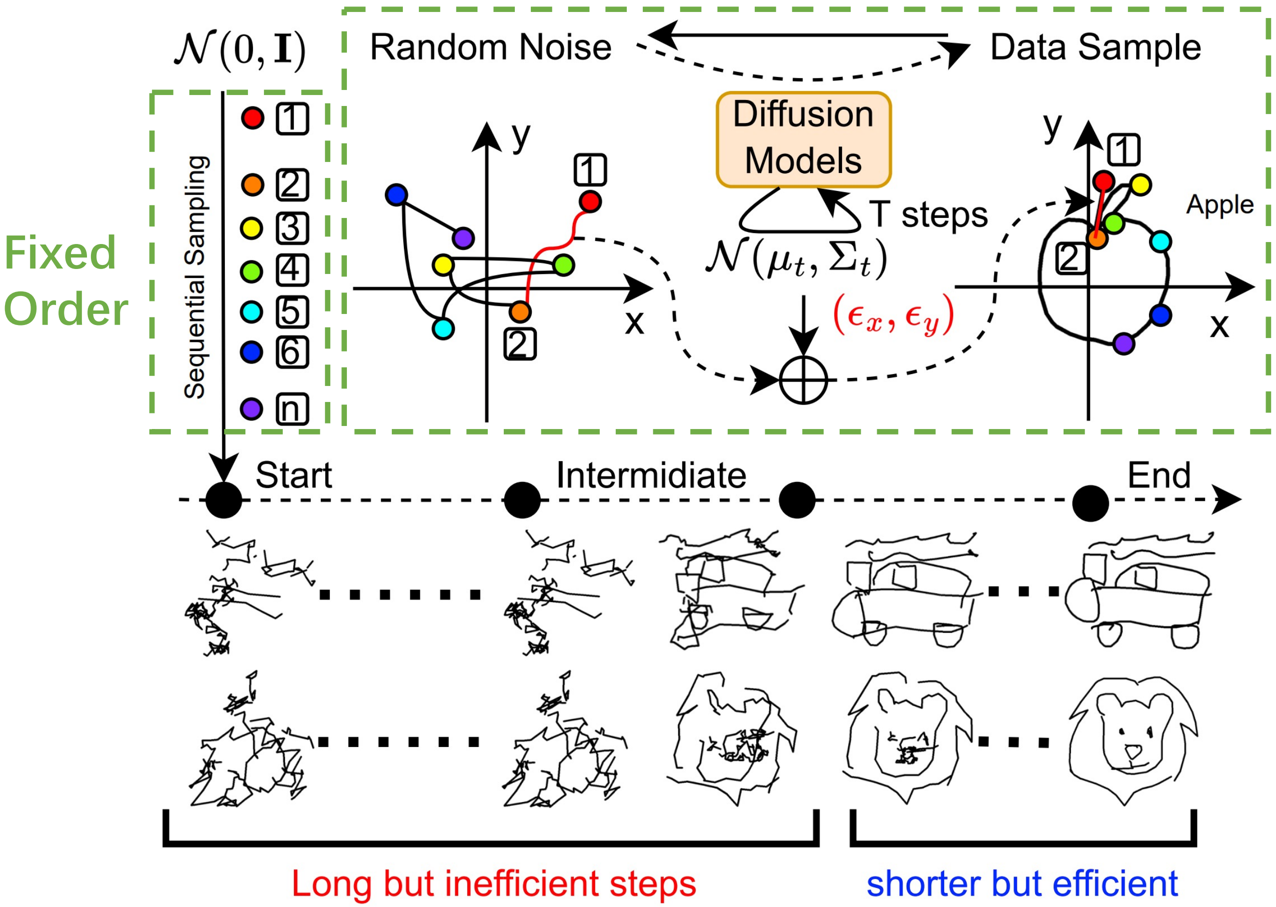
Motivation

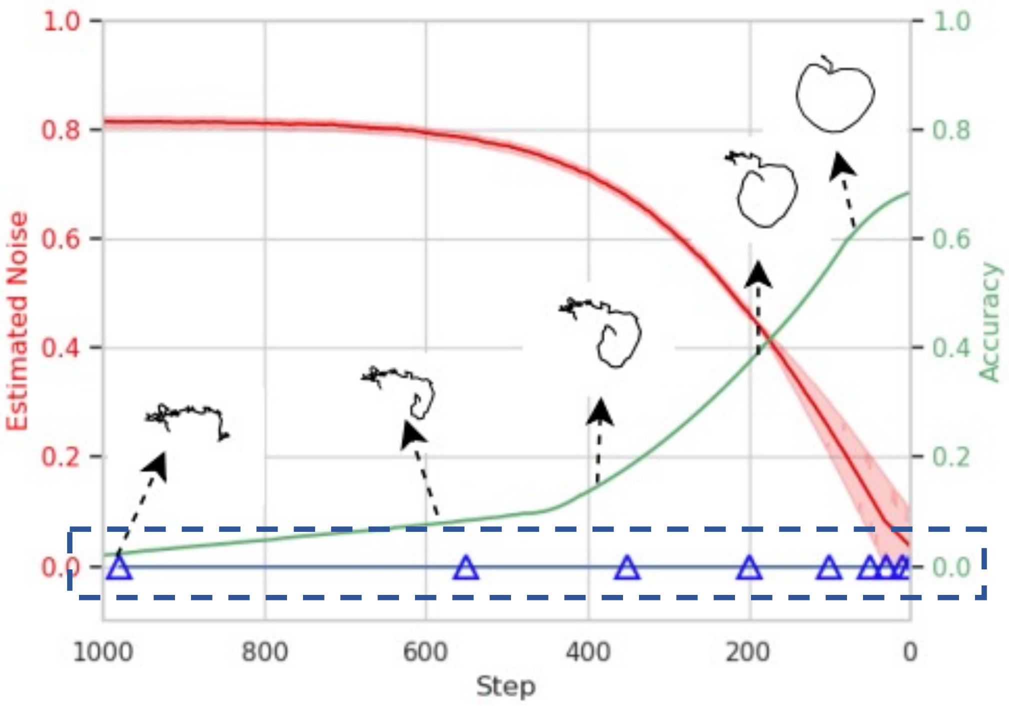


Most approaches for vectorized sketch generation have limited ability to model complex (long stroke points) sketches!

Our work


Fixed Order
$\mathcal{N}(0, \mathbf{I})$
Sequential Sampling
1
2
3
4
5
6
n
Random Noise
Data Sample
Diffusion Models
T steps
$\mathcal{N}(\mu_t, \Sigma_t)$
(ϵ_x, ϵ_y)
Apple
x
y
Start
Intermidiate
End
Long but inefficient steps
shorter but efficient


Estimated Noise
Accuracy
Step
1.0
0.8
0.6
0.4
0.2
0.0
1000
800
600
400
200
0

Our Model

- Vectorization representation $\quad s_0 = \{s^1, s^2, \dots, s^N\} \qquad s^i = (\underline{\Delta x^i, \Delta y^i}, \underline{g^i})$

- Forward Process

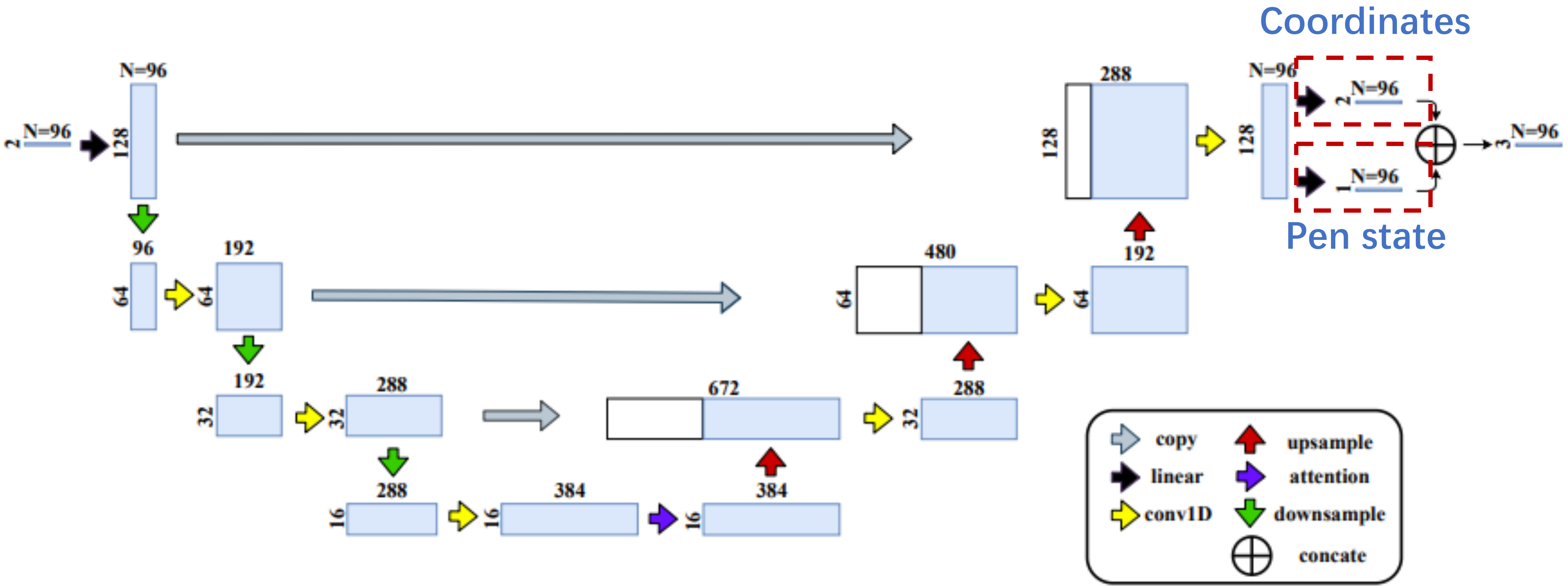
$$q(s_{1:T}|s_0) \coloneqq \prod_{t=1}^{T} q(s_t|s_{t-1}) \qquad q(s_t|s_{t-1}) \coloneqq N\left(s_t; \sqrt{\frac{\alpha_t}{\alpha_{t-1}}} s_{t-1}, \left(1 - \frac{\alpha_t}{\alpha_{t-1}}\right) I\right)$$

- Backward (Generative) process

$$s_{t-1} = \sqrt{\alpha_{t-1}} \left(\frac{s_t - \sqrt{1-\alpha_t \underline{\epsilon_\theta^{(t)}(s_t)}}}{\sqrt{\alpha_t}} \right) + \sqrt{1 - \alpha_{t-1} - \sigma_t^2} \cdot \underline{\epsilon_\theta^{(t)}(s_t)}$$

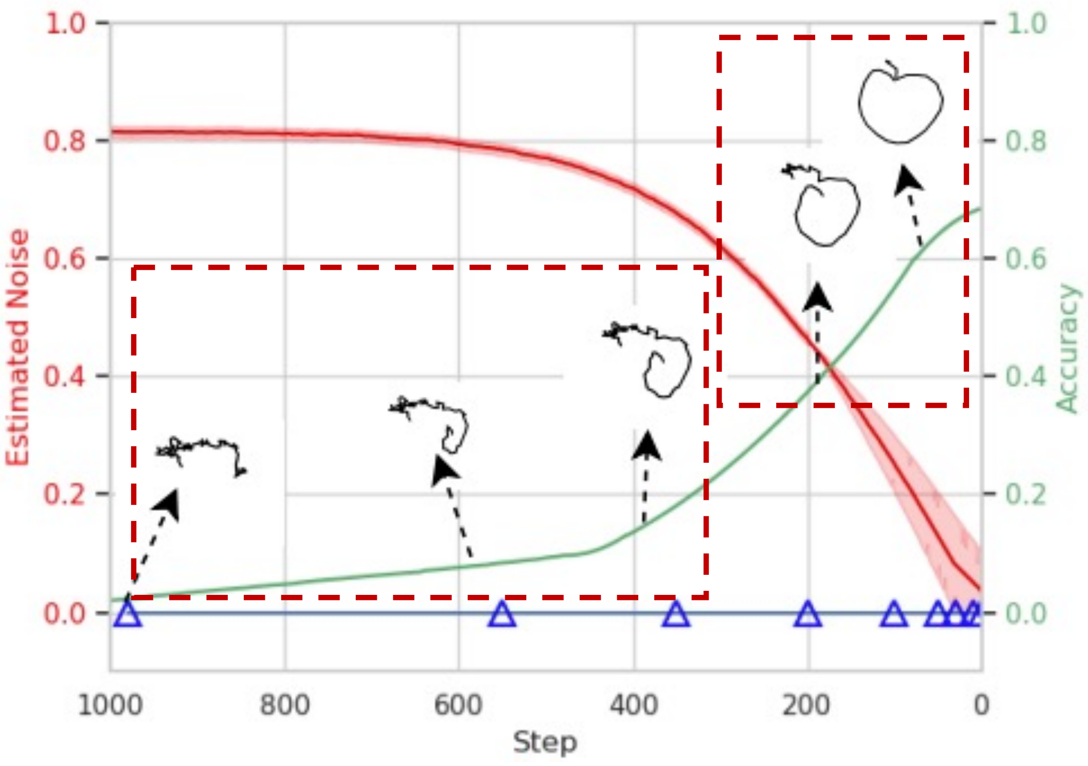
Our Model

- Noise approximator $\epsilon_\theta^{(t)}$



Our Model

- Recognizability based shortcut sampling

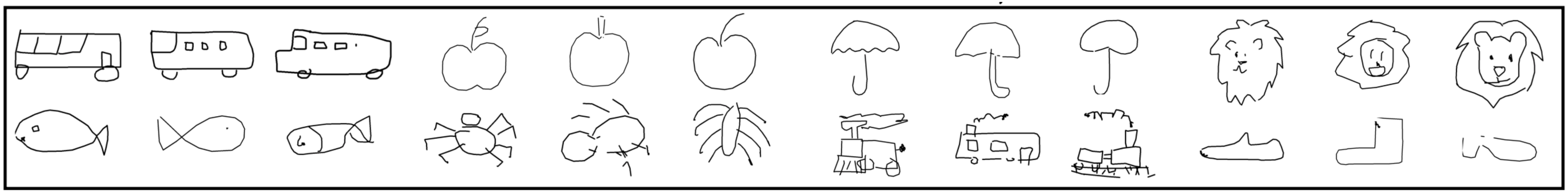


$$\hat{r}_t = h_\phi(\epsilon_\theta^{(t)}, t)$$

$$t_n = \begin{cases} t_c - m, & \text{if} \quad \hat{r}_t < \zeta \\ t_c - 1, & \text{otherwise} \end{cases}$$

Results

Sketch generated from random noise



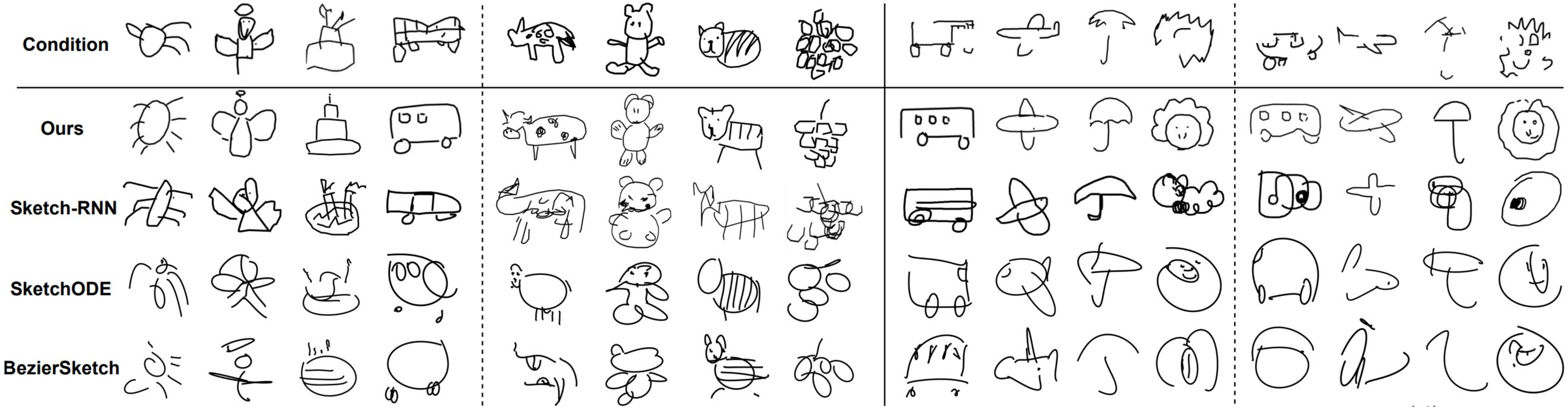
Unconditional generation

Results

Model	Simple				Moderate				Complex				Speed↓
	FID↓	GS↓	Prec↑	Rec↑	FID↓	GS↓	Prec↑	Rec↑	FID↓	GS↓	Prec↑	Rec↑	
SketchPix2seq	13.3	7.0	0.40	0.79	16.4	49.7	0.38	0.75	18.0	73.3	0.36	0.72	0.04
SketchHealer	10.3	5.9	0.45	0.81	12.9	9.8	0.39	0.79	25.9	93.2	0.29	0.63	0.03
SketchRNN	10.8	5.4	0.44	0.82	13.0	11.0	0.42	0.77	21.4	97.6	0.35	0.72	0.03
Diff-HW	13.3	6.8	0.42	0.81	15.9	23.4	0.37	0.76	18.3	64.4	0.23	0.64	0.19
SketchODE	11.5	9.4	0.48	0.74	18.8	29.6	0.31	0.66	33.5	68.1	0.20	0.58	0.03
Ours (full 1000 steps)	**6.9**	**3.4**	**0.52**	**0.88**	**8.4**	**4.7**	**0.45**	**0.87**	**9.4**	**5.2**	**0.42**	**0.85**	1.29
Ours (r-Shortcut, S=30)	7.4	3.9	0.47	0.87	8.9	5.2	0.44	0.85	10.5	6.1	0.39	0.81	0.08
Ours (Linear-DDIMs, S=30)	11.9	6.4	0.38	0.81	13.3	8.8	0.36	0.78	15.1	9.6	0.33	0.72	0.08
Ours (Quadratic-DDIMs, S=30)	12.3	6.6	0.41	0.79	13.8	8.7	0.35	0.76	15.4	9.9	0.34	0.75	0.09
Ours (Abs)	20.7	12.1	0.18	0.55	23.4	64.6	0.13	0.48	29.4	98.9	0.10	0.39	0.20
Ours (Point-Shuffle)	9.5	5.3	0.35	0.72	11.3	7.5	0.31	0.65	12.4	8.1	0.20	0.61	0.18
Ours (Stroke-Shuffle)	8.2	3.8	0.36	0.74	9.6	7.4	0.34	0.66	10.3	7.6	0.25	0.62	0.18

Results

DL	Recognition				Retrieval					
	acc@1(%)		acc@10(%)		mAP		acc@1(%)		acc@10(%)	
–	51.9	52.4 (+0.90)	87.7	90.2 (+2.50)	0.704	0.789 (+0.045)	67.4	73.3 (+5.90)	91.3	96.2 (+4.90)
10%	45.7	48.9 (+3.20)	82.3	82.4 (+0.10)	0.724	0.788 (+0.064)	66.9	73.1 (+6.20)	92.1	96.8 (+4.70)
20%	33.0	47.3 (+14.3)	68.2	81.9 (+13.7)	0.607	0.772 (+0.165)	55.8	72.8 (+17.0)	81.8	94.7 (+12.9)
30%	20.6	48.2 (+27.6)	51.5	81.9 (+30.4)	0.496	0.787 (+0.291)	46.9	72.8 (+25.9)	68.9	95.0 (+26.1)
50%	7.29	50.1 (+42.8)	27.1	84.3 (+57.2)	0.328	0.786 (+0.458)	28.6	74.9 (+46.3)	47.8	96.3 (+48.5)



Conclusion

- We show for the first time sketch generation can be formulated as a process of deformation-based denoising.
- We devise a recognition-based skip function for a more efficient sampling.
- Model trained for unconditional generation could be readily extended for conditional generation by incorporating a perceptual similarity based gradients into the sampling.

For further details and code, please see our paper and visit our GitHub page at: https://github.com/XDUWQ/SketchKnitter

Thank you!