Supplementary Material: Finding Badly Drawn Bunnies

S.1. Proof for the Lower Bound of Normalised Softmax in Sec. 3.1 of Main Text

Eq. 2 of the main text specifies the following formulation:

$$\tilde{L}_{sm} = -\frac{1}{C}\sum_{i=1}^{C}\log\frac{e^{W_{y_i}^T W_{y_i}||f(x_i)||}}{\sum\limits_{j=1}^{C} e^{W_{y_j}^T W_{y_i}||f(x_i)||}} \tag{1}$$

With $||W_{y_j}||$ set to 1 (*i.e.* $e^{W_{y_i}^T W_{y_i}} = e^1$), we can transform Eq. 1 to:

$$\tilde{L}_{sm} = \frac{1}{C}\sum_{i=1}^{C}\log\left(1+\sum_{j=1,j\neq i}^{C} e^{||f(x_i)||(W_{y_j}^T W_{y_i}-1)}\right) \tag{2}$$

Since e^x is a convex function and by Jensen's inequality ($\frac{1}{n}\sum_{i=1}^{n} e^{x_i} \geq e^{\frac{1}{n}\sum_{i=1}^{n} x_i}$), the following holds:

$$\tilde{L}_{sm} \geq \frac{1}{C}\sum_{i=1}^{C}\log\left(1+(C-1)e^{\frac{||f(x_i)||}{C-1}\sum\limits_{j=1,j\neq i}^{C}(W_{y_j}^T W_{y_i}-1)}\right) \tag{3}$$

With SoftPlus $\log(1+Ce^x)$ further a convex function itself subjected to $C>0$ (*i.e.* $\frac{1}{n}\sum_{i=1}^{n}\log(1+Ce^{x_i}) \geq \log(1+Ce^{\frac{1}{n}\sum_{i=1}^{n} x_i})$), Eq. 3 transforms to:

$$\tilde{L}_{sm} \geq \log\left(1+(C-1)e^{\left(\frac{||f(x_i)||}{C(C-1)}\sum\limits_{i=1}^{C}\sum\limits_{j=1,j\neq i}^{C} W_{y_j}^T W_{y_i}\right)-||f(x_i)||}\right) \tag{4}$$

Note that $||\sum\limits_{i=1}^{C} W_{y_i}||_2^2 = C + \sum\limits_{i=1}^{C}\sum\limits_{j=1,j\neq i}^{C} W_{y_i}^T W_{y_j} \implies \sum\limits_{i=1}^{C}\sum\limits_{j=1,j\neq i}^{C} W_{y_i}^T W_{y_j} \geq -C$, we then get to the lower bound of L_{sm}:

$$\tilde{L}_{sm} \geq \log\left(1+(C-1)e^{-\frac{C}{C-1}||f(x_i)||}\right) \tag{5}$$

S.2. Constraint Proof for the GACL Instantiations in Sec. 3.3 of Main Text

We provide proof here that the proposed four GACL instantiations meet our constraints named after geometry, co-optimisation and convexity. The precise mathematical formulations of these three constraints have been respectively defined in the Sec. 3.2 of main text:

(i) $\frac{\nabla_q A(q_i,\theta_{y_i})}{\nabla_\theta A(q_i,\theta_{y_i})} > 0$

(ii) $\nabla_\theta A(q_i,\theta_{y_i})|_{q_i=q_i'} \leq 0$

(iii) $\nabla_q^2 A(q_i,\theta_{y_i}) \leq 0$

$\boldsymbol{A(q_i,\theta_{y_i}) = (1-q_i)s\cos\theta_{y_i}}$ It is easy to prove that (iii) holds with the first and second derivative of $A(q_i,\theta_{y_i})$ to q_i as:

$$\begin{aligned}\nabla_q\, A(q_i,\theta_{y_i}) &= -s\cos\theta_{y_i}\\ \nabla_q^2\, A(q_i,\theta_{y_i}) &= 0\end{aligned} \tag{6}$$

We then calculate the derivative with respect to θ_{y_i}:

$$\nabla_\theta A(q_i,\theta_{y_i}) = -(1-q_i)s\sin\theta_{y_i} \tag{7}$$

Given our implementation of $q_i \in [0.1, 0.3]$, this means $1-q_i$ remains positive throughout. With θ_{y_i} in the range of $[0,\frac{\pi}{2}]$, it becomes evident that (i)(ii) hold.

$\boldsymbol{A(q_i,\theta_{y_i}) = s\cos(q_i\theta_{y_i})}$ The first and second derivative of $A(q_i,\theta_{y_i})$ to q_i are:

$$\begin{aligned}\nabla_q A(q_i,\theta_{y_i}) &= -s\theta_{y_i}\sin(q_i\theta_{y_i})\\ \nabla_q^2 A(q_i,\theta_{y_i}) &= -s\theta_{y_i}^2\cos(q_i\theta_{y_i})\end{aligned} \tag{8}$$

Since $s\theta_{y_i}^2 > 0$ and our implementation of q_i, θ_{y_i} ensures that $\cos(q_i\theta_{y_i}) > 0$, (iii) always holds. We then calculate the derivative with respect to θ_{y_i}:

$$\nabla_\theta A(q_i,\theta_{y_i}) = -sq_i\sin(q_i\theta_{y_i}) \tag{9}$$

Since $\nabla_\theta A(q_i,\theta_{y_i})$ remains negative throughout, it is to derive that (i)(ii) hold.

We omit the proof for $\boldsymbol{A(q_i,\theta_{y_i}) = s\cos(\theta_{y_i}+q_i)}$ and $\boldsymbol{A(q_i,\theta_{y_i}) = s\cos\theta_{y_i} - q_i}$, where similar analysis can be conducted.

S.3. q_i for Quality-Guided Sketch Generation

In this section, we show that q_i can be re-purposed as a plug-and-play quality critic into existing sketch generative models for quality-guided sketch generation – this produces the results in the Sec. 4.4 of main text. To our best knowledge, either conditional or unconditional sketch generative models [2–4, 7] are currently quality unattended. Without loss of generality, we take SketchRNN [4], the pioneering sketch generative method that paves the base for many subsequent works as our model choice. SketchRNN takes the form of a variational auto-encoder [6], with a bidirectional LSTM as encoder that projects a sequence of sketch points s into latent embedding $z = E(s)$, and a LSTM decoder $D(\cdot)$ conditioned on z to reconstruct s. We refer the readers to the SketchRNN paper for more details.

We portray the problem of quality-guided sketch generation as an iterative process of latent feature discovery. This means given s and its initial latent representation z_0, we aim to traverse in the latent space to a target z that is not too far to z_0 but with a significantly higher quality score under $q(\cdot)$, which is formulated as:

$$\begin{aligned} L_{latent} &= (q_{max} - q(D(z))) + \alpha(z - z_0)^2 \\ z &:= z - \lambda \bigtriangledown_z L_{latent} \end{aligned} \tag{10}$$

where q_{max} corresponds to different u_q values under different instantiations, α and λ are two hyper-parameters controlling relative importance of identity preservation and gradient descent step size.

Non-differentiable point sampling. Eq. 10 requires gradients flowing from $q(\cdot)$ back to $D(\cdot)$, which is potentially problematic in practice as $D(\cdot)$ involves non-differentiable operation during the sampling of Gaussian Mixture Model (GMM)[1] for sketch point generation. Putting formally, suppose the GMM is instantiated with M normal distributions, this means we need to sample from a categorical vector Π of length M that represents the mixture weights, resulting in backpropagation discontinuity. We get around this issue by: (i) Gumbel-Softmax [5], a differentiable approximate sampling mechanism for categorical variables via reparametrisation trick. (ii) straight-through gradient estimator [1] for discrete actions in argmax. Combining both turns the once indifferentiable $y = \text{one\_hot}(\underset{i}{\text{argmax}}(\Pi_i))$ to:

$$\begin{aligned} &y_{soft} = (\Pi'_1, \Pi'_2, ...\Pi'_M) \;\; \Pi'_i = \frac{exp((\Pi_i + g_i)/\tau)}{\sum_{j=1}^{M} exp((\Pi_j + g_j)/\tau)} \\ &y_{hard} = \text{one\_hot}(\underset{i}{\text{argmax}}(y_{soft})) \\ &y_{new} = \text{stop\_gradient}(y_{hard} - y_{soft}) + y_{soft} \end{aligned} \tag{11}$$

$g_1, g_2, ..., g_M$ are i.i.d samples drawn from $\text{Gumbel}(0, 1)$[2], τ is the softmax temperature that interpolates between discrete one-hot-encoded categorical distributions and continuous categorical densities. By replacing y with y_{new}, we can now proceed Eq. 10 in an end-to-end manner.

[1]Modelling each sketch point as a Gaussian Mixture Model is observed in most existing sketch generations works [4,7,8]. This is in contrast to the single-modal normal distribution that corresponds to common L_2 regression loss for maximum likelihood estimation.

[2]$Gumbel(0, 1)$ is sampled by first drawing $u \sim \text{Uniform}(0, 1)$ and compute $g_i = -\log(-\log(u))$.

References

[1] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013. 2

[2] Ayan Das, Yongxin Yang, Timothy M Hospedales, Tao Xiang, and Yi-Zhe Song. Cloud2curve: Generation and vectorization of parametric sketches. In *CVPR*, 2021. 2

[3] Songwei Ge, Vedanuj Goswami, C Lawrence Zitnick, and Devi Parikh. Creative sketch generation. In *ICLR*, 2020. 2

[4] David Ha and Douglas Eck. A neural representation of sketch drawings. In *ICLR*, 2018. 2

[5] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *ICLR*, 2017. 2

[6] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2

[7] Jifei Song, Kaiyue Pang, Yi-Zhe Song, Tao Xiang, and Timothy M Hospedales. Learning to sketch with shortcut cycle consistency. In *CVPR*, 2018. 2

[8] Guoyao Su, Yonggang Qi, Kaiyue Pang, Jie Yang, and Yi-Zhe Song. Sketchhealer: A graph-to-sequence network for recreating partial human sketches. In *BMVC*, 2020. 2