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:
\[
\tilde{L}_{sm} = -\frac{1}{C} \sum_{i=1}^{C} \log \left( \sum_{j=1}^{C} e^{W^T y_i ||f(x_i)||} \right)
\]  

(1)

With \( ||W_{y_i}|| \) set to 1 (i.e. \( e^{W^T y_i} = e^1 \)), we can transform Eq. 1 to:
\[
\tilde{L}_{sm} = \frac{1}{C} \sum_{i=1}^{C} \log \left( 1 + \sum_{j=1,j\neq i}^{C} e^{||f(x_i)||} (W^T y_i - 1) \right)
\]  

(2)

With SoftPlus \( \log(1+Ce^x) \) further a convex function itself subjected to \( C > 0 \) (i.e. \( \frac{1}{C} \sum_{i=1}^{C} \log(1 + C e^{x_i}) \geq \log(1 + C e^{\frac{1}{C} \sum_{i=1}^{C} x_i}) \)), the following holds:
\[
\tilde{L}_{sm} \geq \frac{1}{C} \sum_{i=1}^{C} \log \left( 1 + (C - 1)e^{||f(x_i)||} \sum_{j=1,j\neq i}^{C} (W^T y_i - 1) \right)
\]  

(3)

With \( \sum_{i=1}^{C} W^T y_i \) further a convex function itself subjected to \( \sum_{i=1}^{C} W^T y_i = C \) (i.e. \( \frac{1}{C} \sum_{i=1}^{C} \sum_{j=1,j\neq i}^{C} W^T y_i \geq \sum_{i=1}^{C} W^T y_i \)), Eq. 3 transforms to:
\[
\tilde{L}_{sm} \geq \log \left( 1 + (C - 1)e^{\sum_{i=1}^{C} \sum_{j=1,j\neq i}^{C} W^T y_i} \right)
\]  

(4)

Note that \( ||\sum_{i=1}^{C} W_{y_i}||^2 = C + \sum_{i=1}^{C} \sum_{j=1,j\neq i}^{C} W^T y_i W_{y_j} \Rightarrow \sum_{i=1}^{C} \sum_{j=1,j\neq i}^{C} W^T y_i W_{y_j} \geq -C \), we then get to the lower bound of \( \tilde{L}_{sm} \):
\[
\tilde{L}_{sm} \geq \log \left( 1 + (C - 1)e^{-\frac{C}{|f(x_i)|}} \right)
\]  

(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, optimisation and convexity. The precise mathematical formulations of these three constraints have been respectively defined in the Sec. 3.2 of main text:

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

(ii) \( \nabla_{q^2} A(q_i, \theta_{y_i}||q_i - q_j| \leq 0 \)

(iii) \( \nabla_{q^2} A(q_i, \theta_{y_i}) \equiv 0 \)

\( A(q_i, \theta_{y_i}) = (1 - q_i) \cos \theta_{y_i} \) 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:
\[
\nabla_q A(q_i, \theta_{y_i}) = -s \cos \theta_{y_i}
\]
\[
\nabla_{q^2} A(q_i, \theta_{y_i}) = 0
\]  

(6)

We then calculate the derivative with respect to \( \theta_{y_i} \):
\[
\nabla_{\theta} A(q_i, \theta_{y_i}) = -(1 - q_i) s \sin \theta_{y_i}
\]  

(7)

Given our implementation of \( q_i \in [0, 1, 0.3] \), this means \( 1 - q_i \) remains positive throughout. With \( \theta_{y_i} \), in the range of \( [0, \frac{\pi}{2}] \), it becomes evident that (i)(ii) hold.
\( A(q_i, \theta_{y_i}) = s \cos(q_i \theta_{y_i}) \) The first and second derivative of \( A(q_i, \theta_{y_i}) \) to \( q_i \) are:
\[
\nabla_q A(q_i, \theta_{y_i}) = -s \theta_{y_i} \sin(q_i \theta_{y_i})
\]
\[
\nabla_{q^2} A(q_i, \theta_{y_i}) = -s \theta_{y_i}^2 \cos(q_i \theta_{y_i})
\]  

(8)

Since \( s \theta_{y_i}^2 > 0 \) and our implementation of \( q_i, \theta_{y_i} \), ensures that \( \cos(q_i \theta_{y_i}) > 0 \), (iii) always holds. We then calculate the derivative with respect to \( \theta_{y_i} \):
\[
\nabla_{\theta} A(q_i, \theta_{y_i}) = -s q_i \sin(q_i \theta_{y_i})
\]  

(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 \( A(q_i, \theta_{y_i}) = s \cos(q_i \theta_{y_i} + q_i) \) and \( A(q_i, \theta_{y_i}) = s \cos \theta_{y_i} - q_i \), 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:

\[
L_{\text{latent}} = (q_{\text{max}} - q(D(z))) + \alpha(z - z_0)^2 \tag{10}
\]

where \( q_{\text{max}} \) corresponds to different \( u_q \) values under different instantiations, \( \alpha \) and \( \lambda \) 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 \( \Pi \) 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 \( y = \text{one.hot}(\text{argmax}(\Pi_i)) \) to:

\[
y_{\text{soft}} = (\Pi'_1, \Pi'_2, \ldots, \Pi'_M), \quad \Pi'_i = \frac{\exp((\Pi_i + g_i)/\tau)}{\sum_j^M \exp((\Pi_j + g_j)/\tau)}
\]

\[
y_{\text{hard}} = \text{one.hot}(\text{argmax}(y_{\text{soft}}))
\]

\[
y_{\text{new}} = \text{stop.gradient}(y_{\text{hard}} - y_{\text{soft}}) + y_{\text{soft}} \tag{11}
\]

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

References


\(^1\)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 \( g_i = -\log(-\log(u)) \).