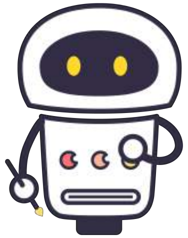
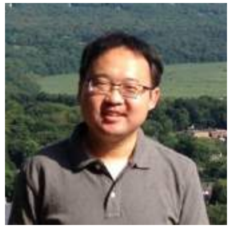
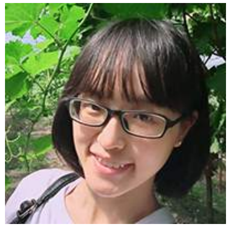


AI-Sketcher: A Deep Generative Model for Generating High Quality Sketches

Nan Cao, **Xin Yan**, Yang Shi, Chaoran Chen

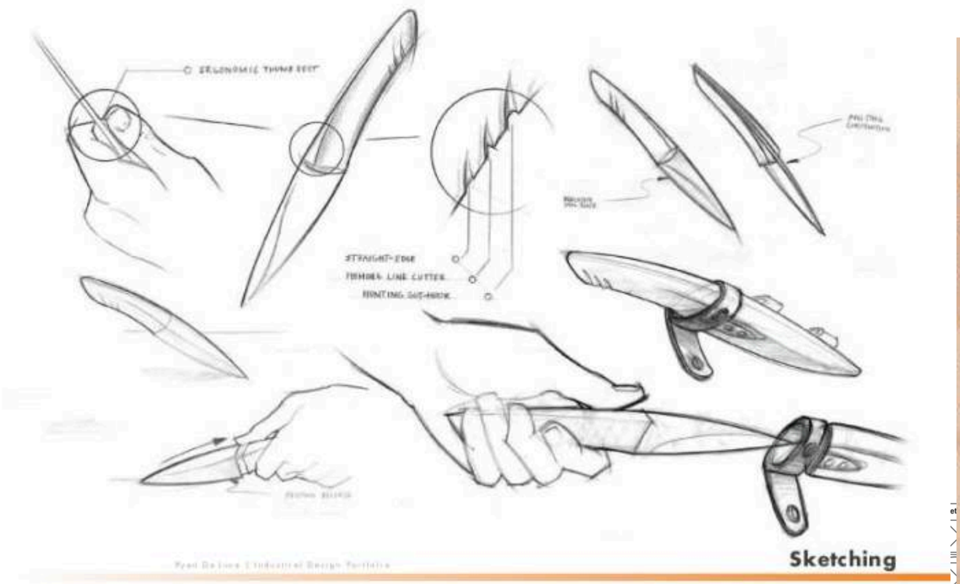
Intelligent Big Data Visualization Lab

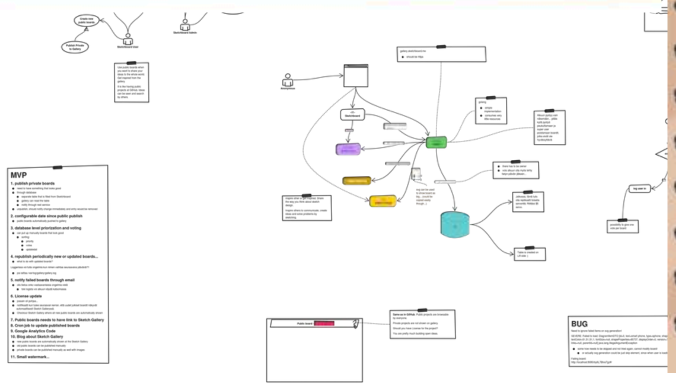
Tongji University

BACKGROUND - Cave Painting

BACKGROUND - Children's Drawings

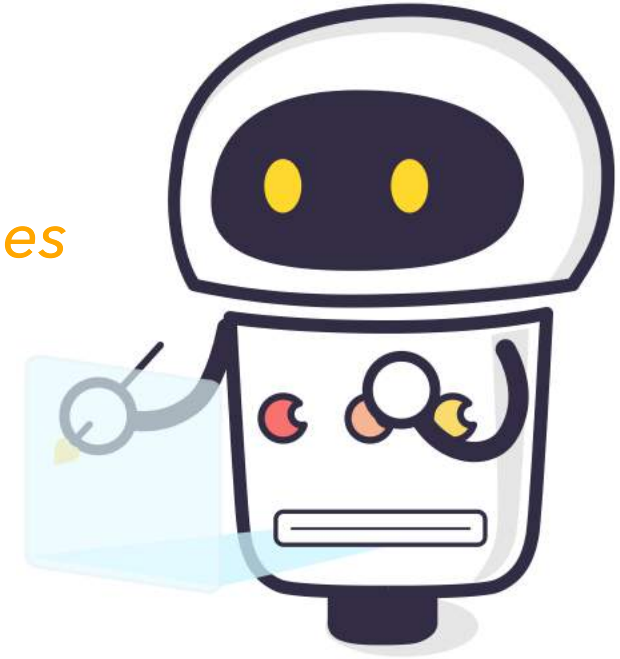
BACKGROUND - Design Sketches

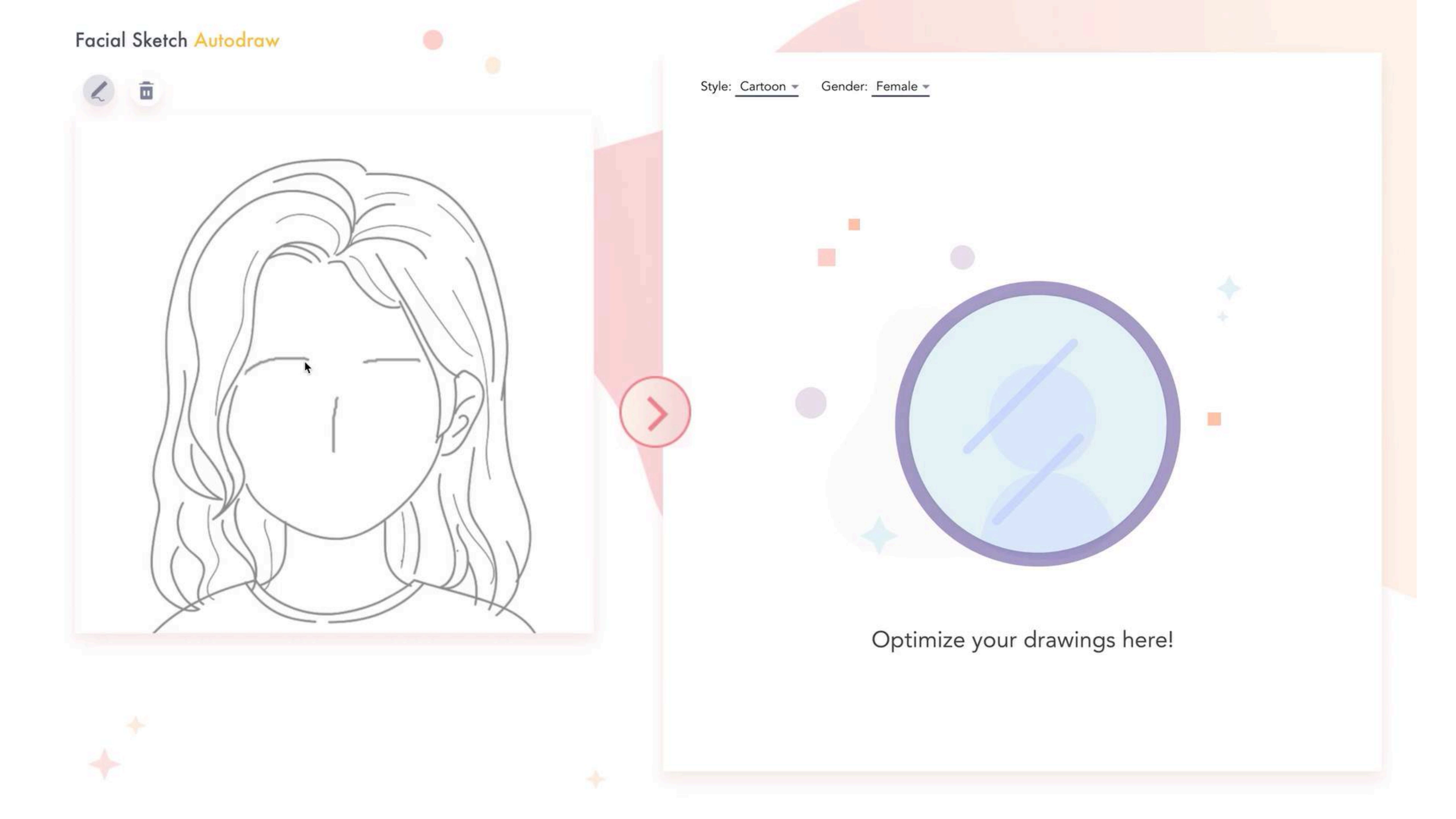




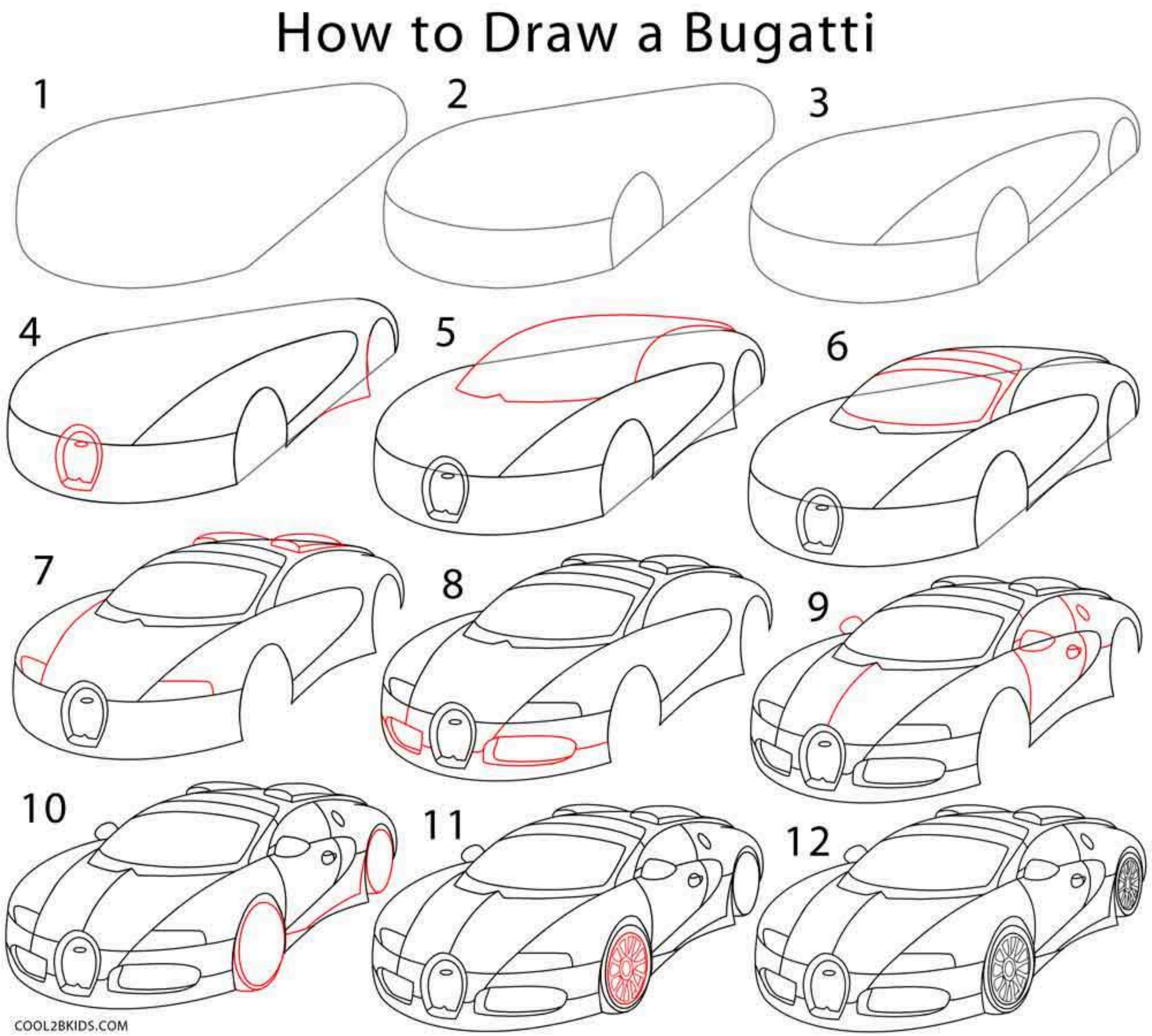
BACKGROUND - Design Sketches

*Can **AI** help designers create high quality sketches and boost their productivity and creativity?*




Facial Sketch Autodraw
Style: Cartoon
Gender: Female
Optimize your drawings here!

CHARACTERISTICS OF SKETCHING



A. Multiple steps

B. Stroke by stroke

C. Only line matters

Reference Pipeline for AI-Supported Sketching

Stroke Encoding

Stroke Encoding

Learning

Generating

continuous

one-hot

$$[\triangle x, \quad \triangle y, \quad p_1, \quad p_2, \quad p_3]$$

- $\triangle x, \triangle y$: Start coordinates relative to the end of last stroke
- p_1: continuous drawing from the last point,
- p_2: end of drawing a stroke
- p_3: end of drawing a sketch

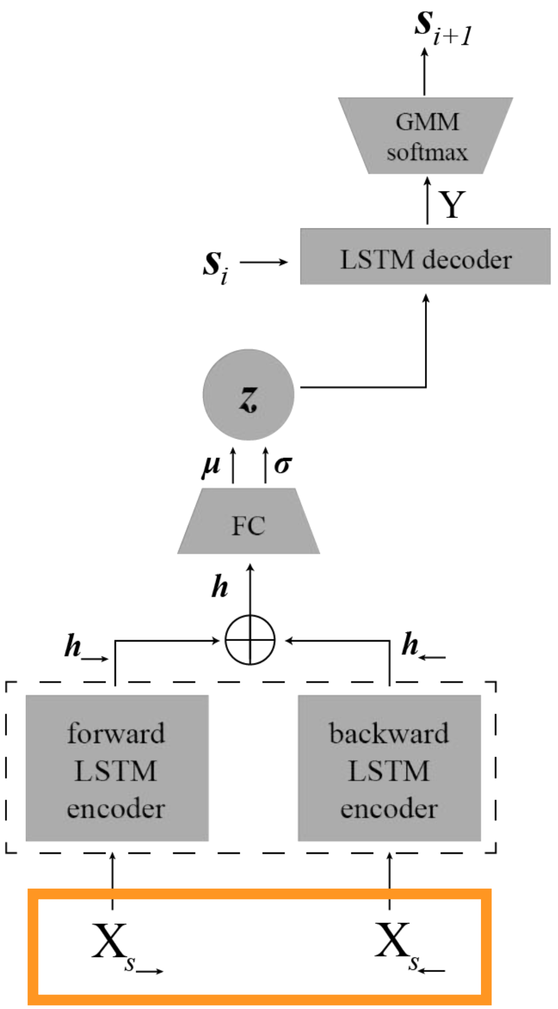
——"A Neural Representation of Sketch Drawings"

Learning (Sketch-RNN)

Stroke Encoding

Learning

Generating



Current Stroke

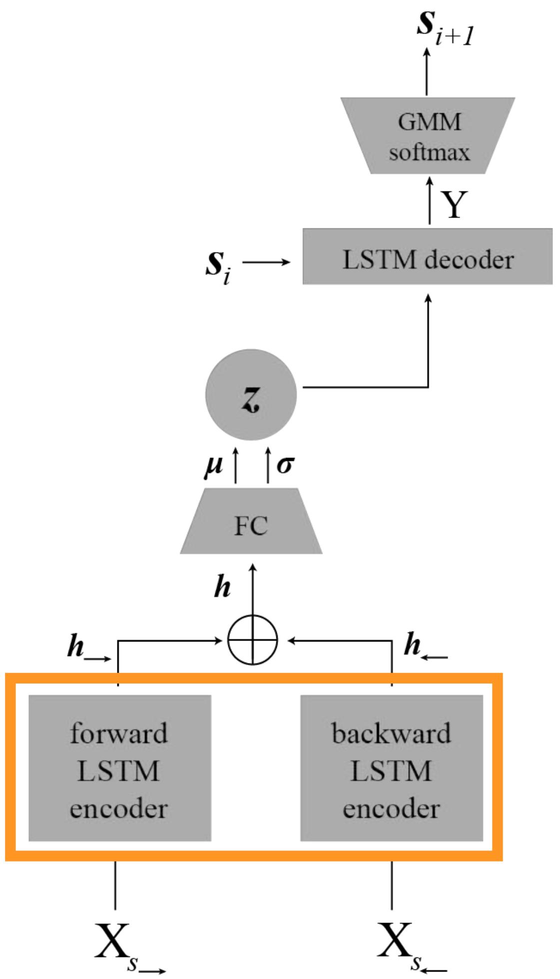
——"A Neural Representation of Sketch Drawings"

Learning (Sketch-RNN)

Stroke Encoding

Learning

Generating



Encoder: Bidirectional RNN (BRNN)

Current Stroke

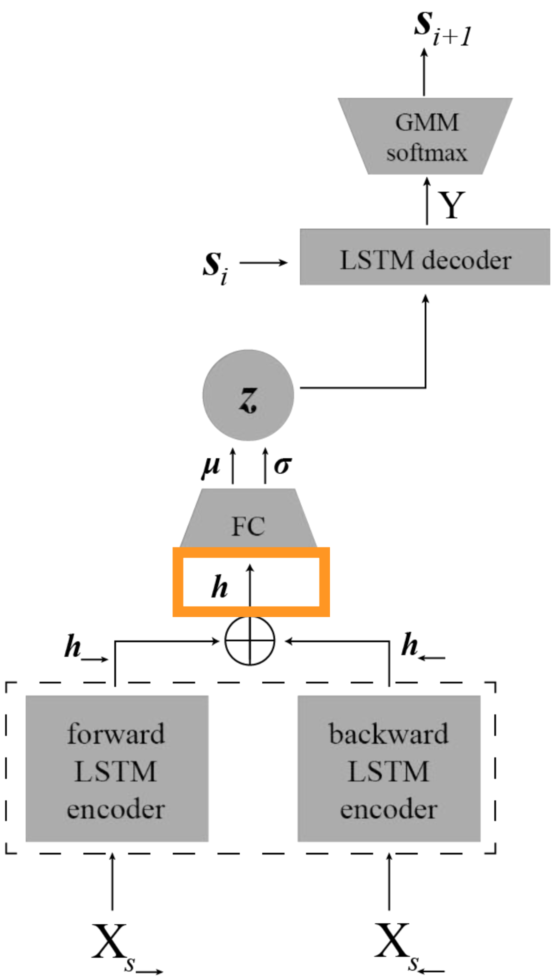
——"A Neural Representation of Sketch Drawings"

Learning (Sketch-RNN)

Stroke Encoding

Learning

Generating



Latent Vector Capturing the drawing behaviors

Encoder: Bidirectional RNN (BRNN)

Current Stroke

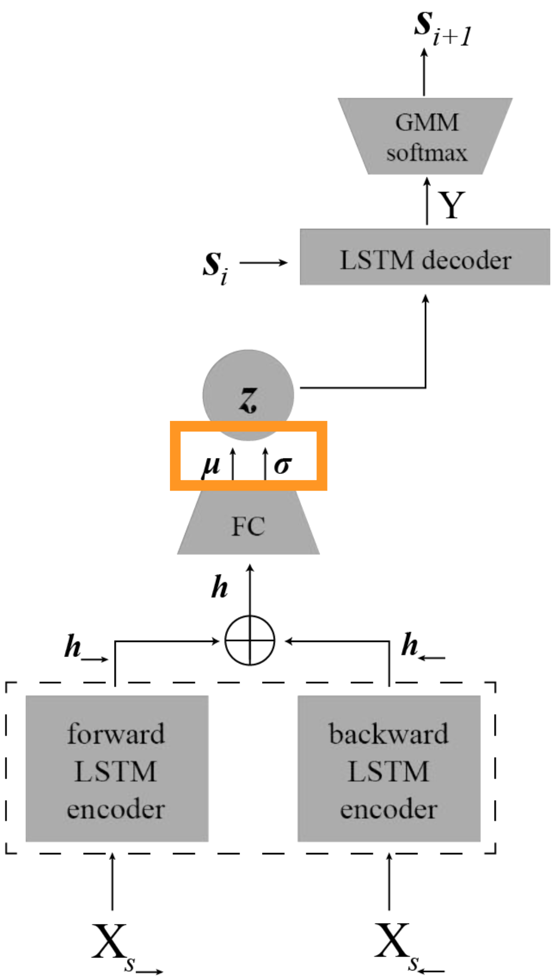
——"A Neural Representation of Sketch Drawings"

Learning (Sketch-RNN)

Stroke Encoding

Learning

Generating



Distribution of strokes

Latent Vector Capturing the drawing behaviors

Encoder: Bidirectional RNN (BRNN)

Current Stroke

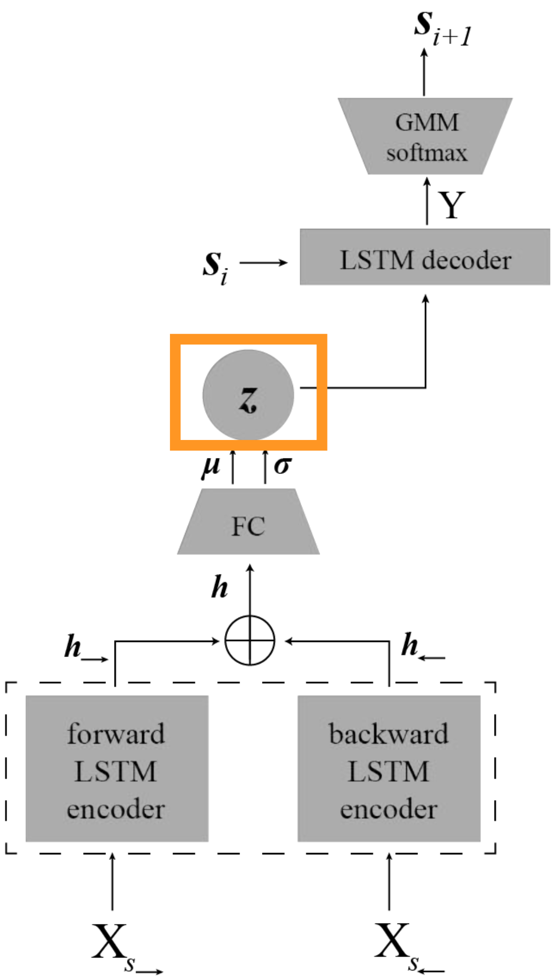
——"A Neural Representation of Sketch Drawings"

Learning (Sketch-RNN)

Stroke Encoding

Learning

Generating



Random Sampling for Stroke Generation

Distribution of strokes

Latent Vector Capturing the drawing behaviors

Encoder: Bidirectional RNN (BRNN)

Current Stroke

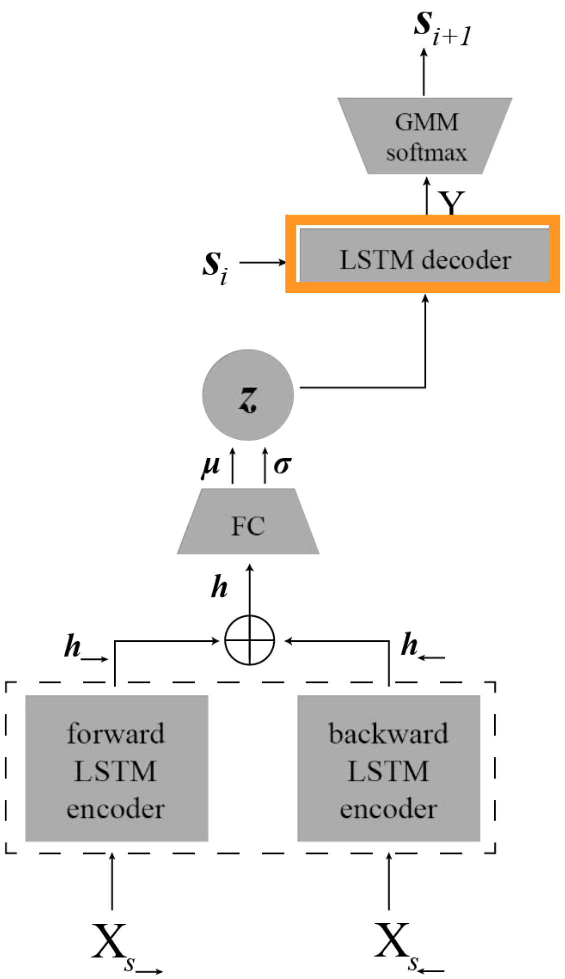
——"A Neural Representation of Sketch Drawings"

Learning (Sketch-RNN)

Stroke Encoding

Learning

Generating



Decoder

Random Sampling for Stroke Generation

Distribution of strokes

Latent Vector Capturing the drawing behaviors

Encoder: Bidirectional RNN (BRNN)

Current Stroke

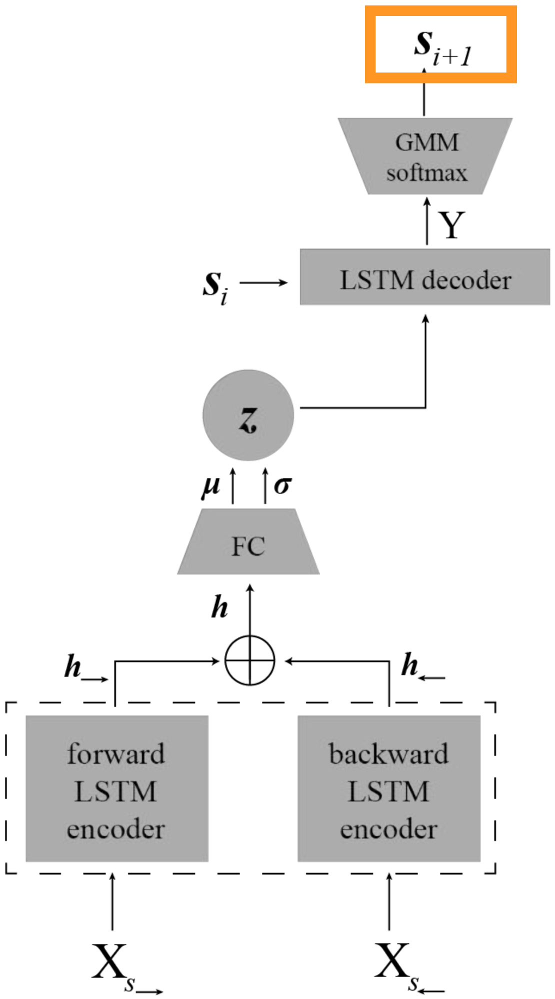
——"A Neural Representation of Sketch Drawings"

Learning (Sketch-RNN)

Stroke Encoding

Learning

Generating



Vector of the next Stroke

Decoder

Random Sampling for Stroke Generation

Distribution of strokes

Latent Vector Capturing the drawing behaviors

Encoder: Bidirectional RNN (BRNN)

Current Stroke

——"A Neural Representation of Sketch Drawings"

Learning (Sketch-RNN)

Stroke Encoding

Learning

Generating

LOSS Function : $\sum_{i=1}^{N} l_i$

——"A Neural Representation of Sketch Drawings"

Learning (Sketch-RNN)

Stroke Encoding

Learning

Generating

LOSS Function : $\sum_{i=1}^{N} l_i$

$$l_i(\theta, \phi) = -E_{z \sim q_\theta(z|x_i)}[\log p_\phi(x_i|z)] + KL(q_\theta(z|x_i)||p(z))$$

——"A Neural Representation of Sketch Drawings"

Learning (Sketch-RNN)

Stroke Encoding

Learning

Generating

LOSS Function : $\sum_{i=1}^{N} l_i$

$$l_i(\theta, \phi) = \underbrace{-E_{z \sim q_\theta(z|x_i)}[\log p_\phi(x_i|z)]}_{} + KL(q_\theta(z|x_i)||p(z))$$

Reconstruction Loss

Expected negative log-likelihood of the *i*-th image

Encourages the decoder to learn to reconstruct the data

——"A Neural Representation of Sketch Drawings"

Learning (Sketch-RNN)

Stroke Encoding

Learning

Generating

LOSS Function : $\sum_{i=1}^{N} l_i$

$$l_i(\theta, \phi) = -E_{z \sim q_\theta(z|x_i)}[\log p_\phi(x_i|z)] + \underbrace{KL(q_\theta(z|x_i)||p(z))}_{\textbf{KL Loss}}$$

Measure of how close q is to p = N(0,1)

——"A Neural Representation of Sketch Drawings"

Generating (Sketch-RNN)

Stroke Encoding

Learning

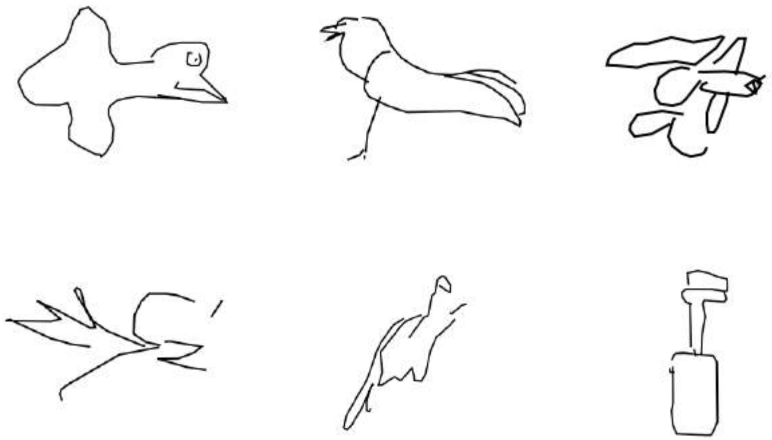
Generating

——"A Neural Representation of Sketch Drawings"

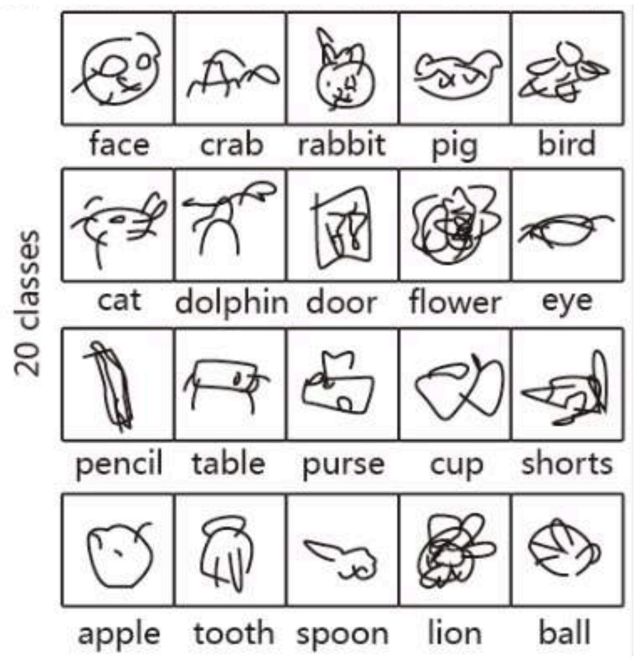
LIMITATION OF SKETCH-RNN

Low quality

1. generating sketches in one category (bird).



2. dealing with multi-class situations

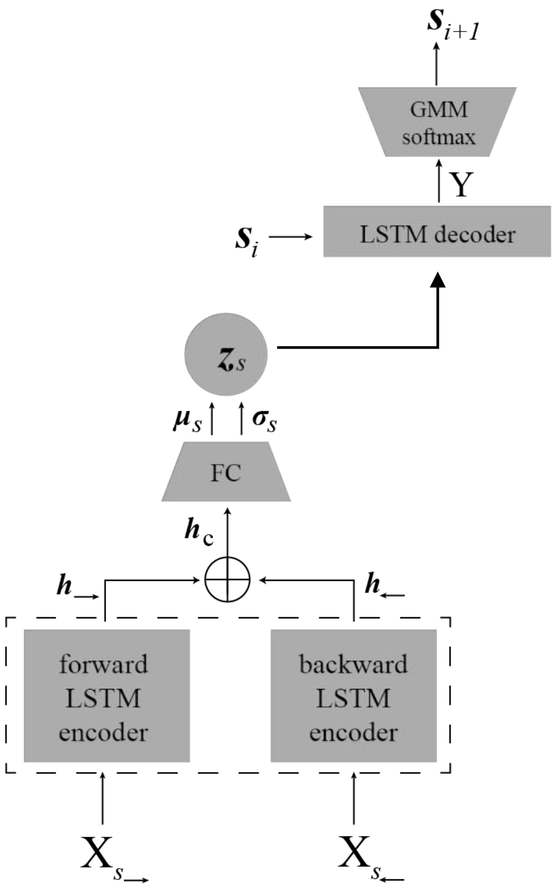


——"A Neural Representation of Sketch Drawings"

AI-Sketcher

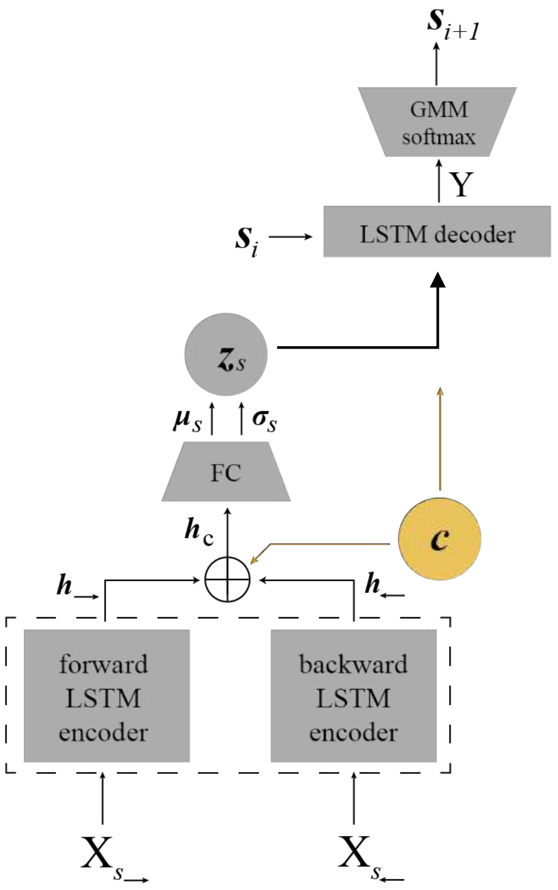
a hybrid deep learning model to automatically generate high quality sketch drawings by learning sequences of strokes

AI-SKETCHER



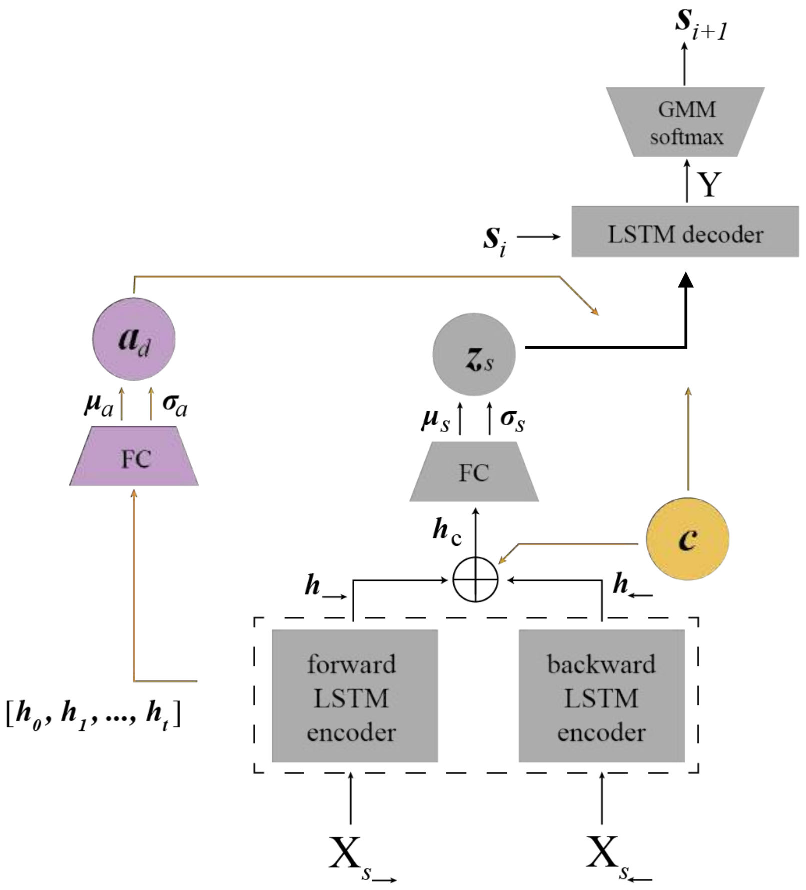
s_{i+1}
GMM softmax
Y
s_i
LSTM decoder
z_s
μ_s
σ_s
FC
h_c
$h_{\rightarrow}$
$h_{\leftarrow}$
forward LSTM encoder
backward LSTM encoder
$X_{s_{\rightarrow}}$
$X_{s_{\leftarrow}}$

AI-SKETCHER



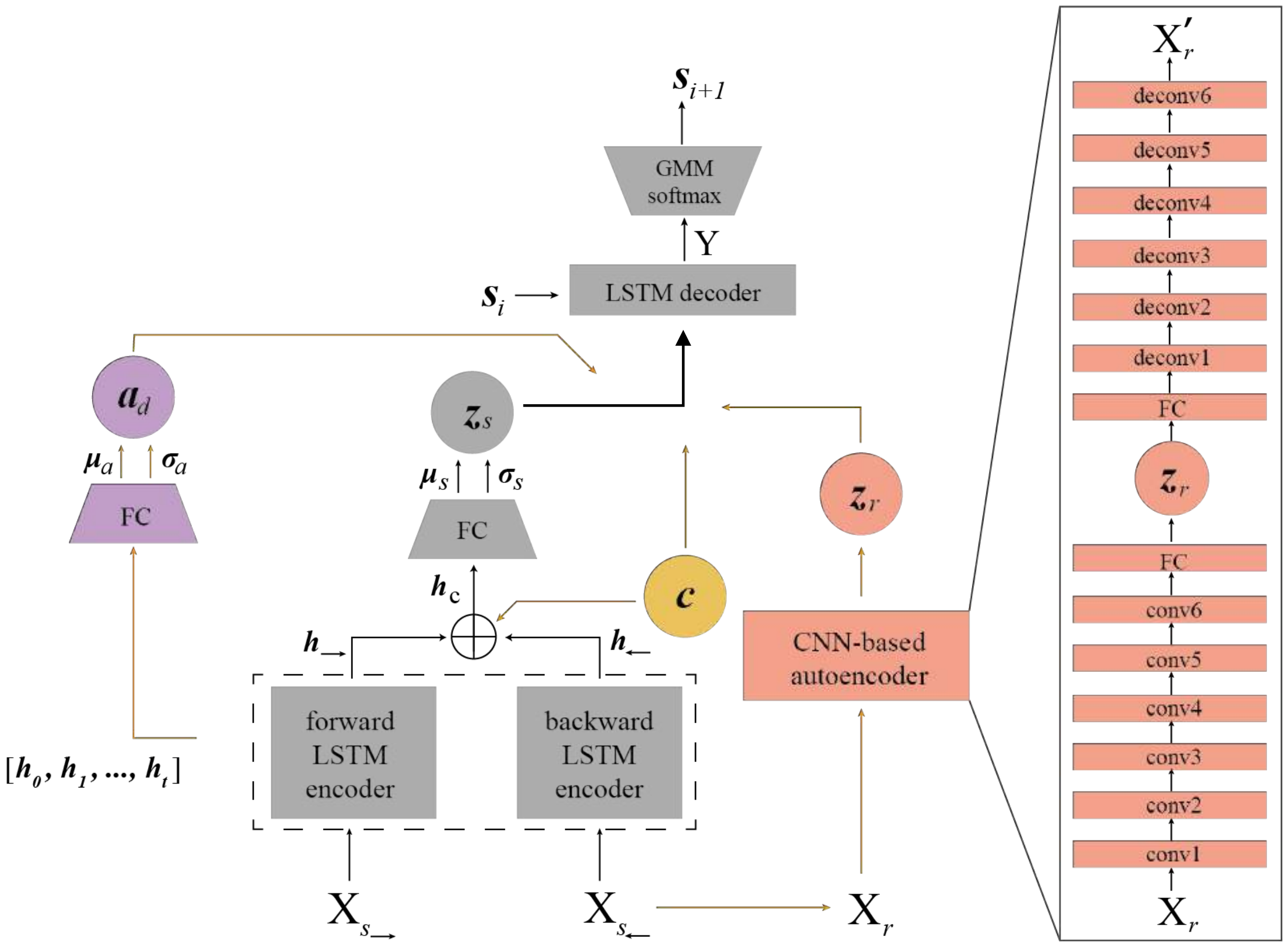
- A conditional vector is used to ensure a high quality generation of sketches from multiple categories.

AI-SKETCHER



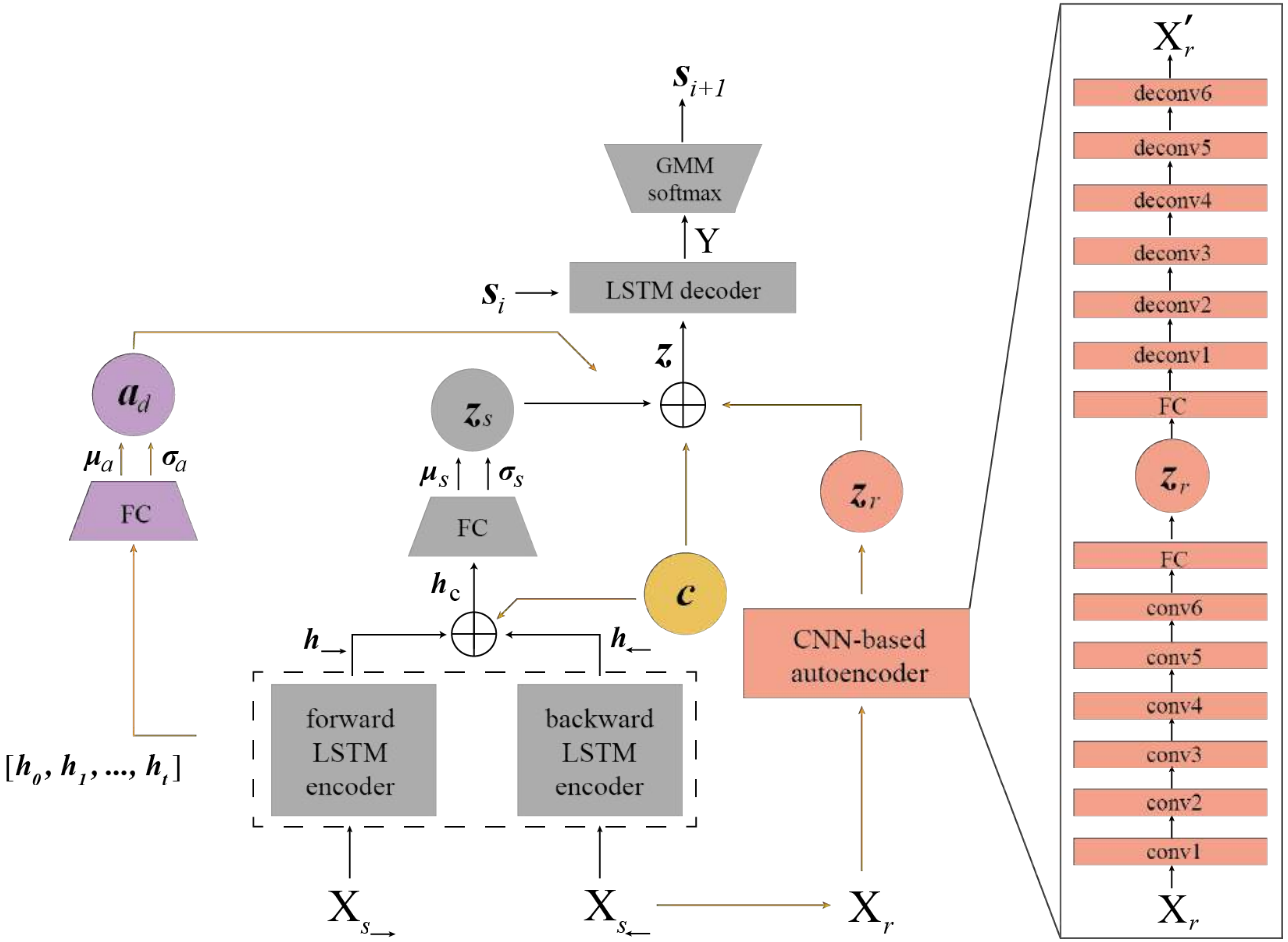
- **A conditional vector** is used to ensure a high quality generation of sketches from multiple categories.
- **A fully-connected layer is introduced to estimate how the previous strokes will influence on the next stroke.**

AI-SKETCHER



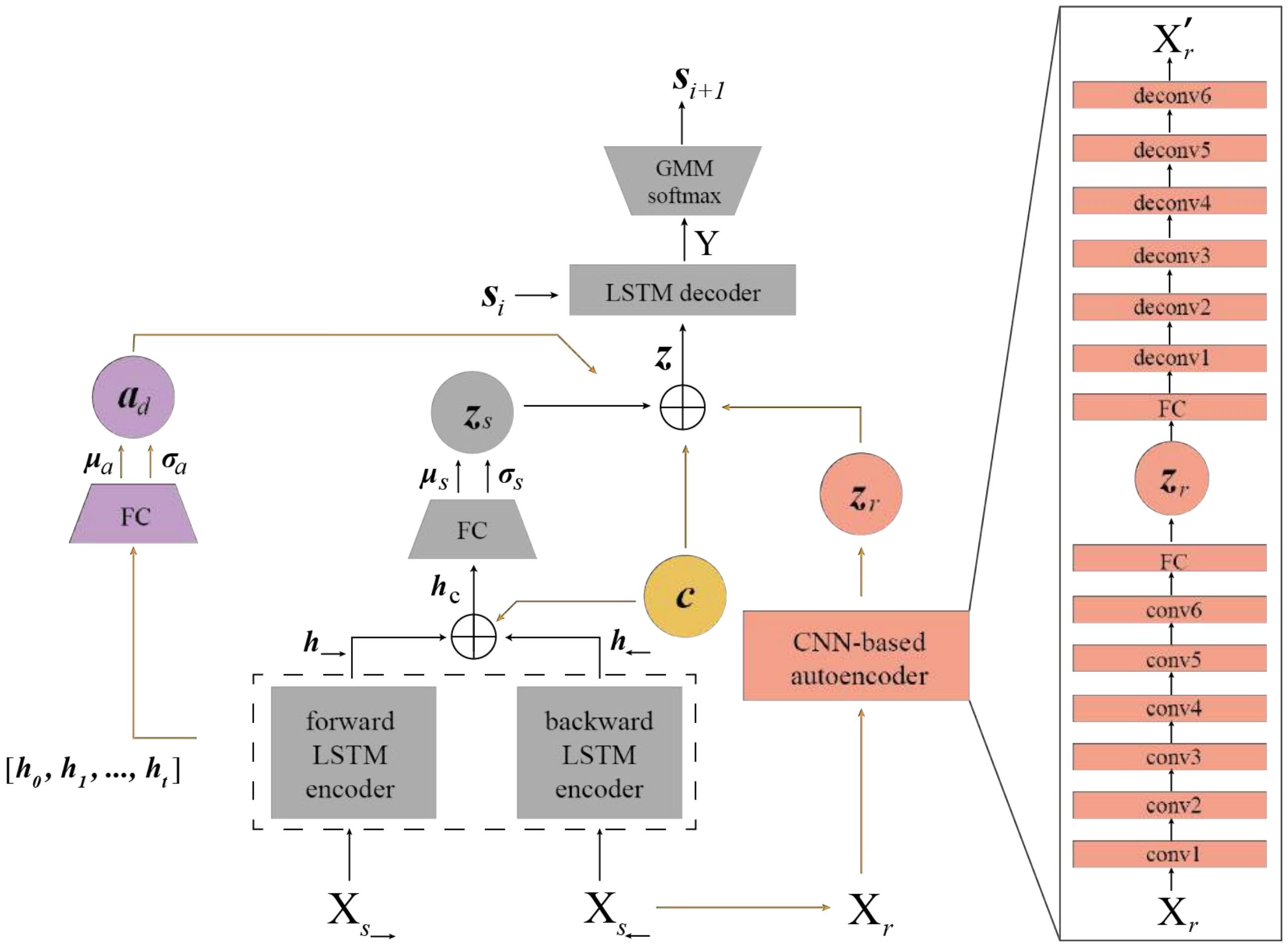
- **A conditional vector** is used to ensure a high quality generation of sketches from multiple categories.
- **A fully-connected layer** is introduced to estimate how the previous strokes will influence on the next stroke.
- **A CNN-based autoencoder is employed to capture the spatial information of a training set.**

AI-SKETCHER



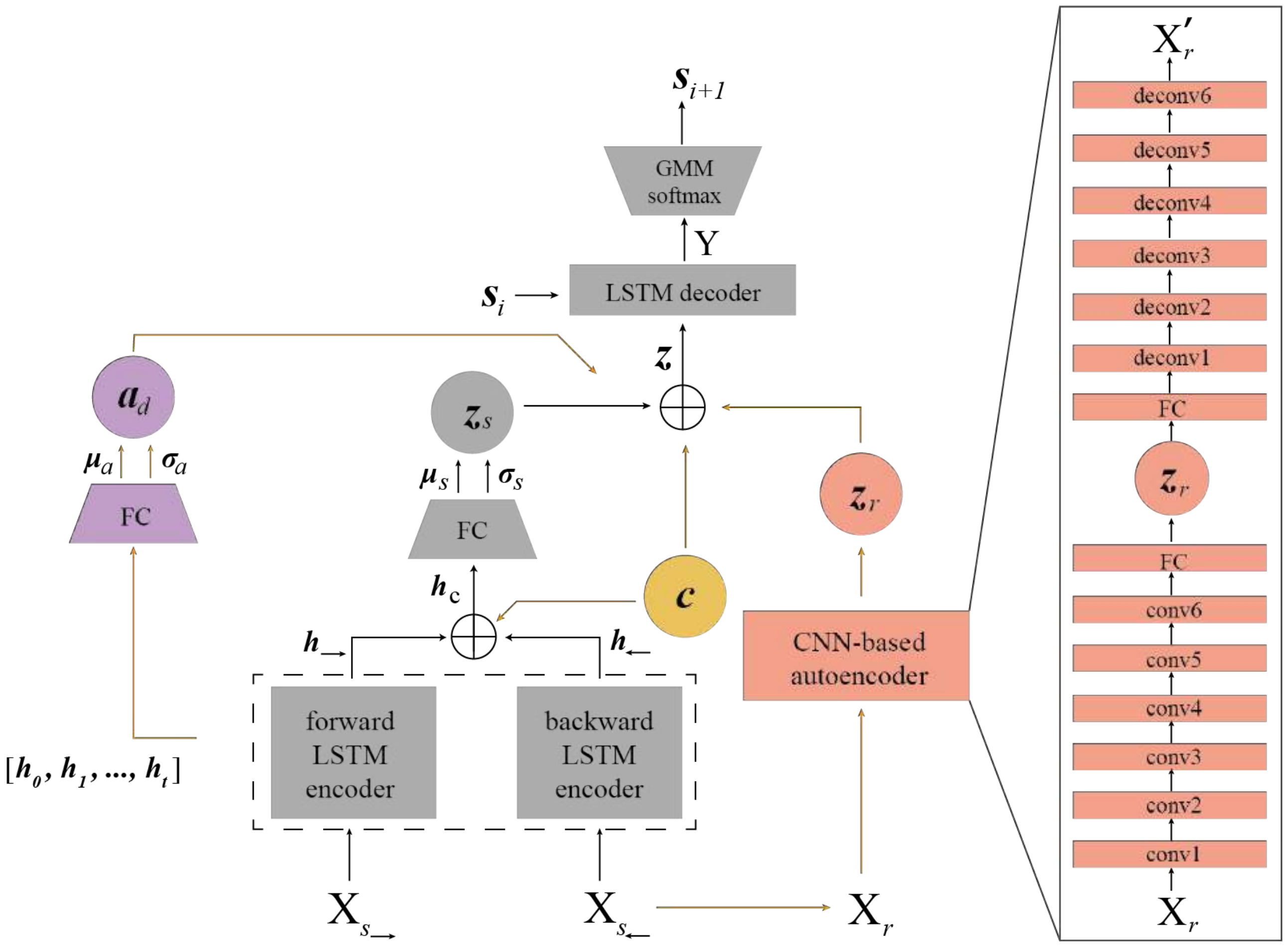
- **A conditional vector** is used to ensure a high quality generation of sketches from multiple categories.
- **A fully-connected layer** is introduced to estimate how the previous strokes will influence on the next stroke.
- **A CNN-based autoencoder** is employed to capture the spatial information of a training set.
- **Loss function is modified.**

AI-SKETCHER



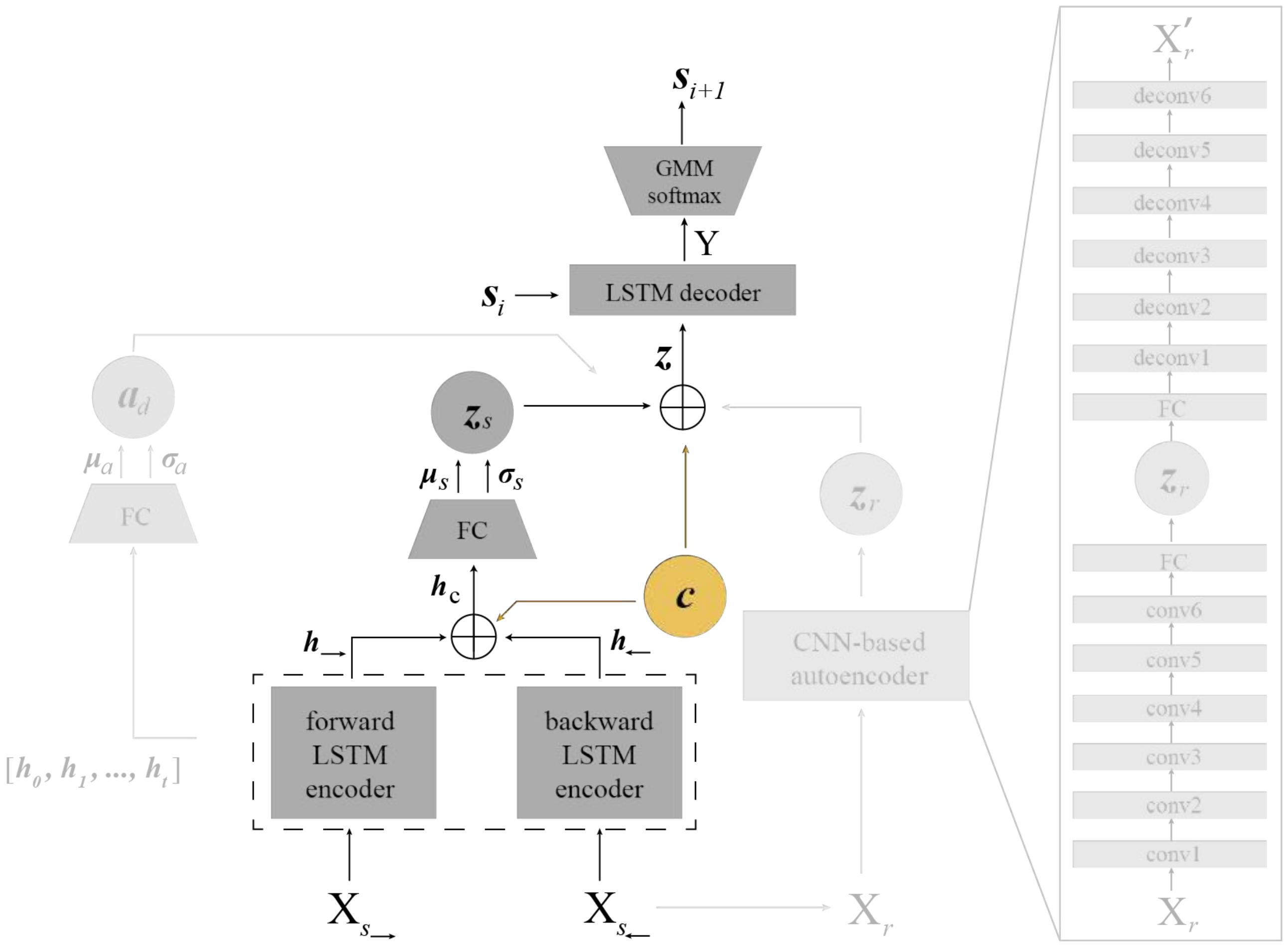
- **A conditional vector** is used to ensure a high quality generation of sketches from multiple categories.
- **A fully-connected layer** is introduced to estimate how the previous strokes will influence on the next stroke.
- **A CNN-based autoencoder** is employed to capture the spatial information of a training set.
- **Loss function** is modified.

AI-SKETCHER



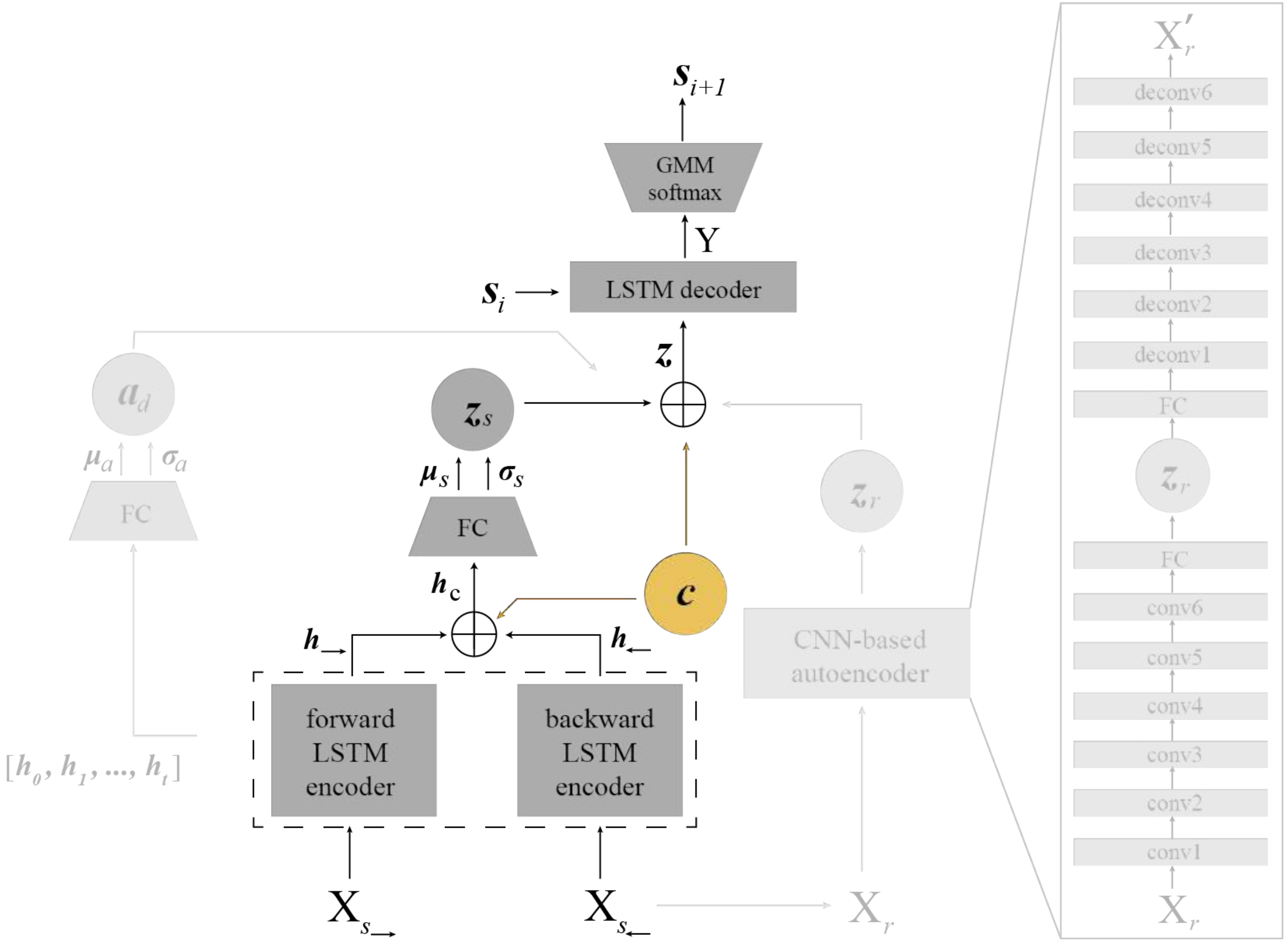
- A conditional vector is used to ensure a high quality generation of sketches from multiple categories.
- **A fully-connected layer** is introduced to estimate how the previous strokes will influence on the next stroke.
- **A CNN-based autoencoder** is employed to capture the spatial information of a training set.
- **Loss function** is modified.

AI-SKETCHER - Conditional Sequence-to-Sequence VAE



- **A conditional vector is used to ensure a high quality generation of sketches from multiple categories.**

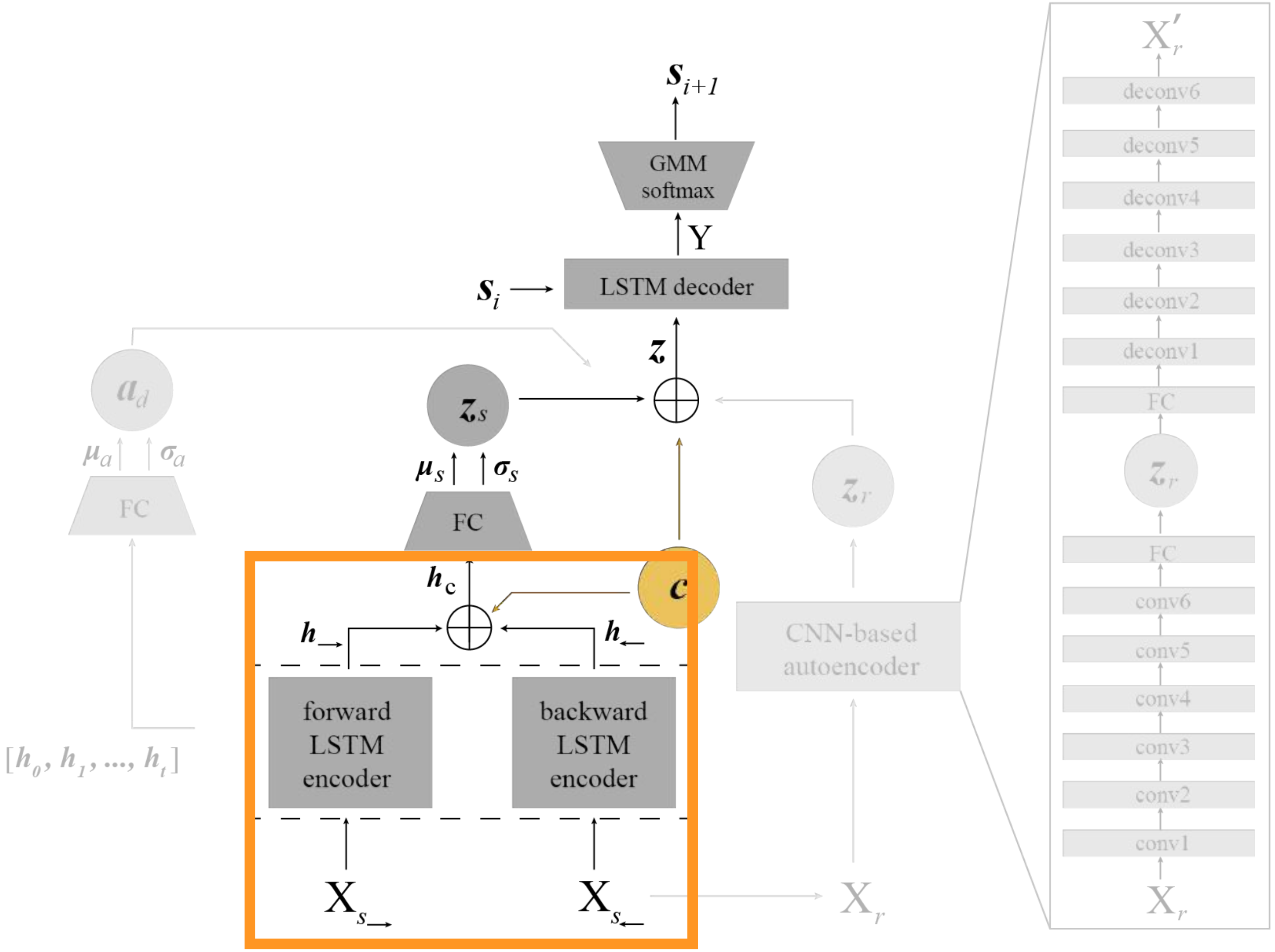
AI-SKETCHER - Conditional Sequence-to-Sequence VAE



- **A conditional vector is used to ensure a high quality generation of sketches from multiple categories.**

Encoder: Bidirectional RNN

AI-SKETCHER - Conditional Sequence-to-Sequence VAE

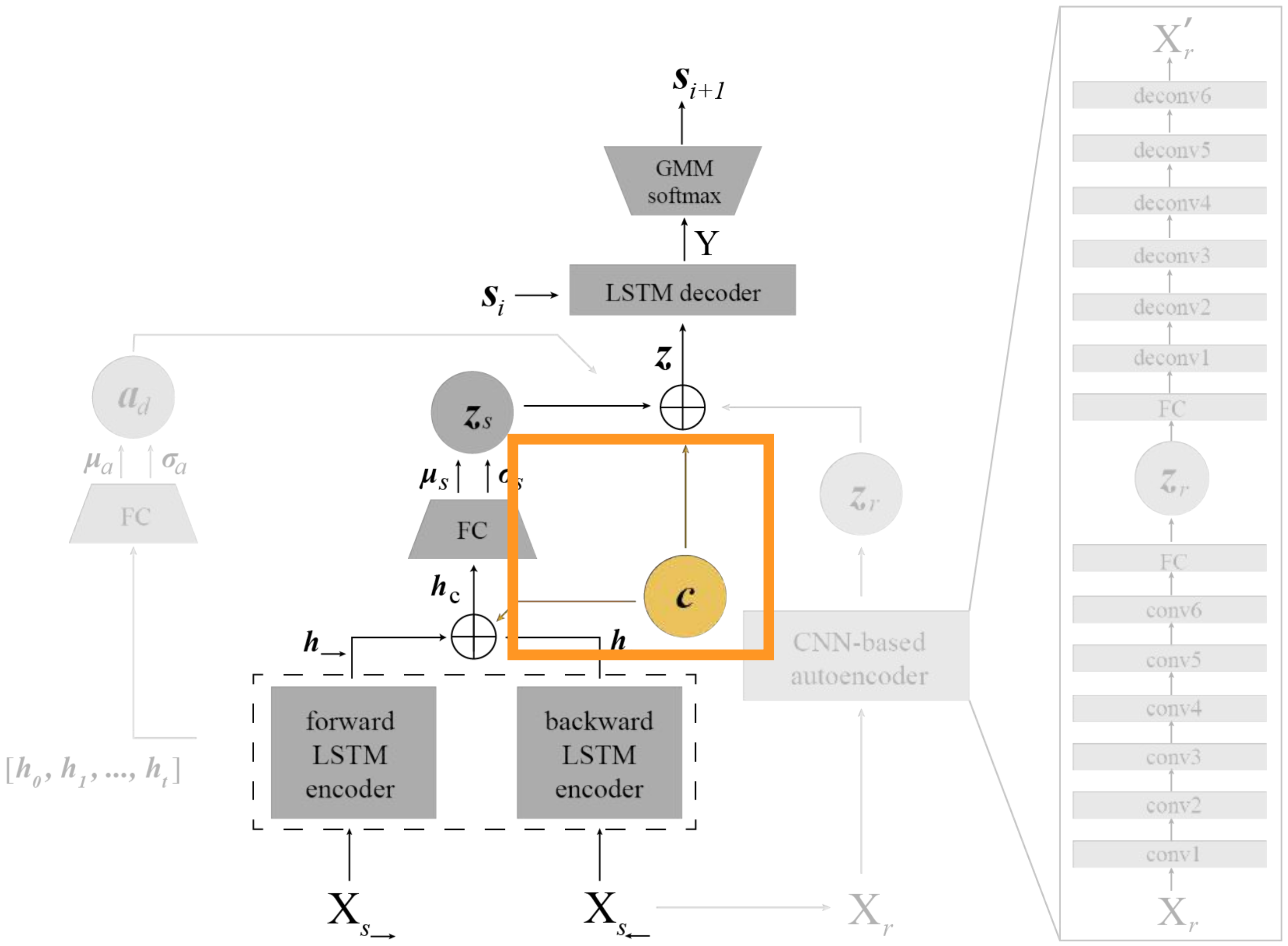


- **A conditional vector is used to ensure a high quality generation of sketches from multiple categories.**

$\boldsymbol{X_s}$: the sequences of strokes.

$$\boldsymbol{h}^{enc} = encode(X_s)$$

AI-SKETCHER - Conditional Sequence-to-Sequence VAE

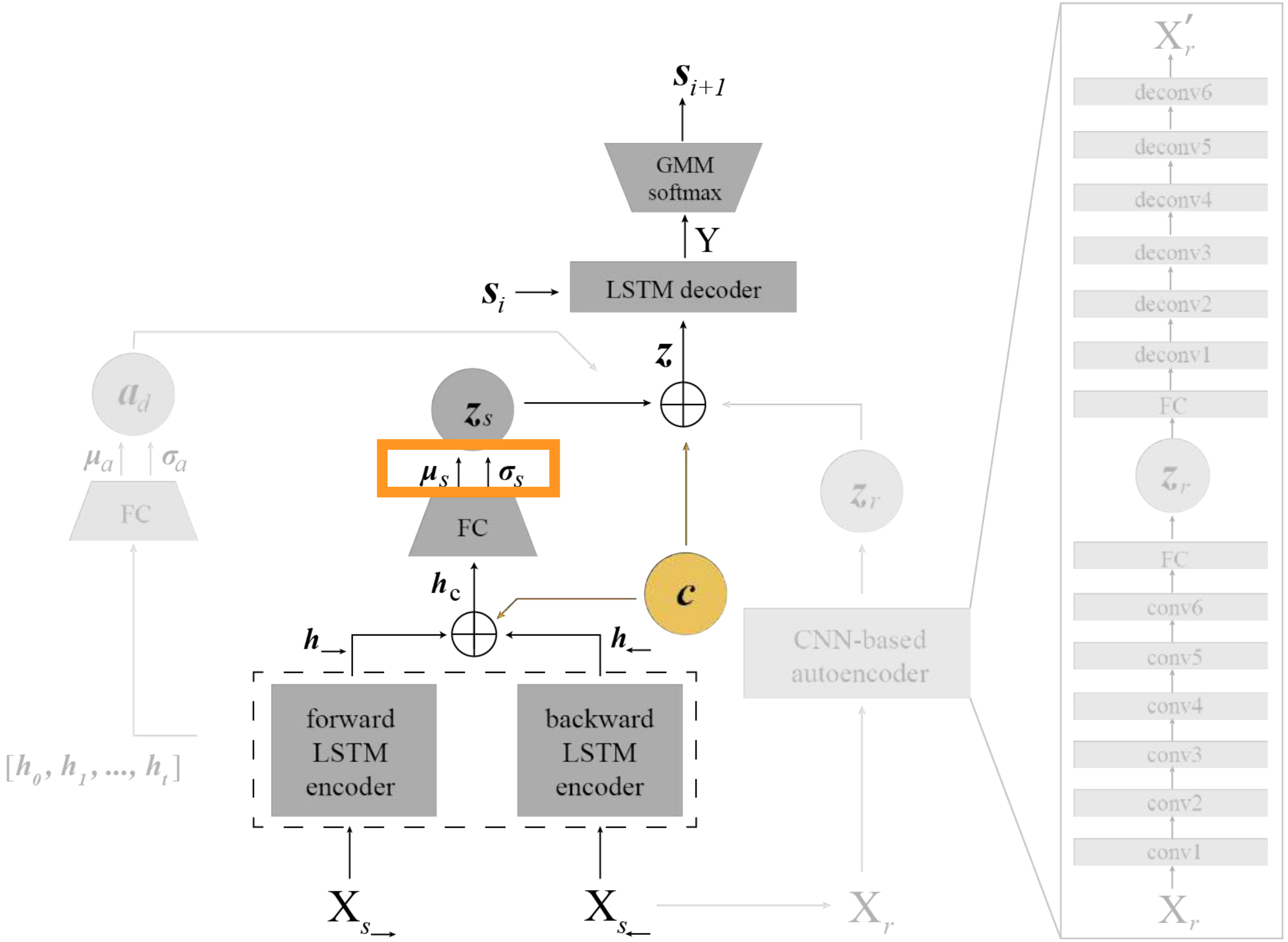


- **A conditional vector is used to ensure a high quality generation of sketches from multiple categories.**

$$\boldsymbol{h}_c = [\boldsymbol{h}^{enc}; \boldsymbol{c}]$$

$\boldsymbol{c}$ is a k-dimensional one-hot conditional vector with k indicates the number of conditions.

AI-SKETCHER - Conditional Sequence-to-Sequence VAE



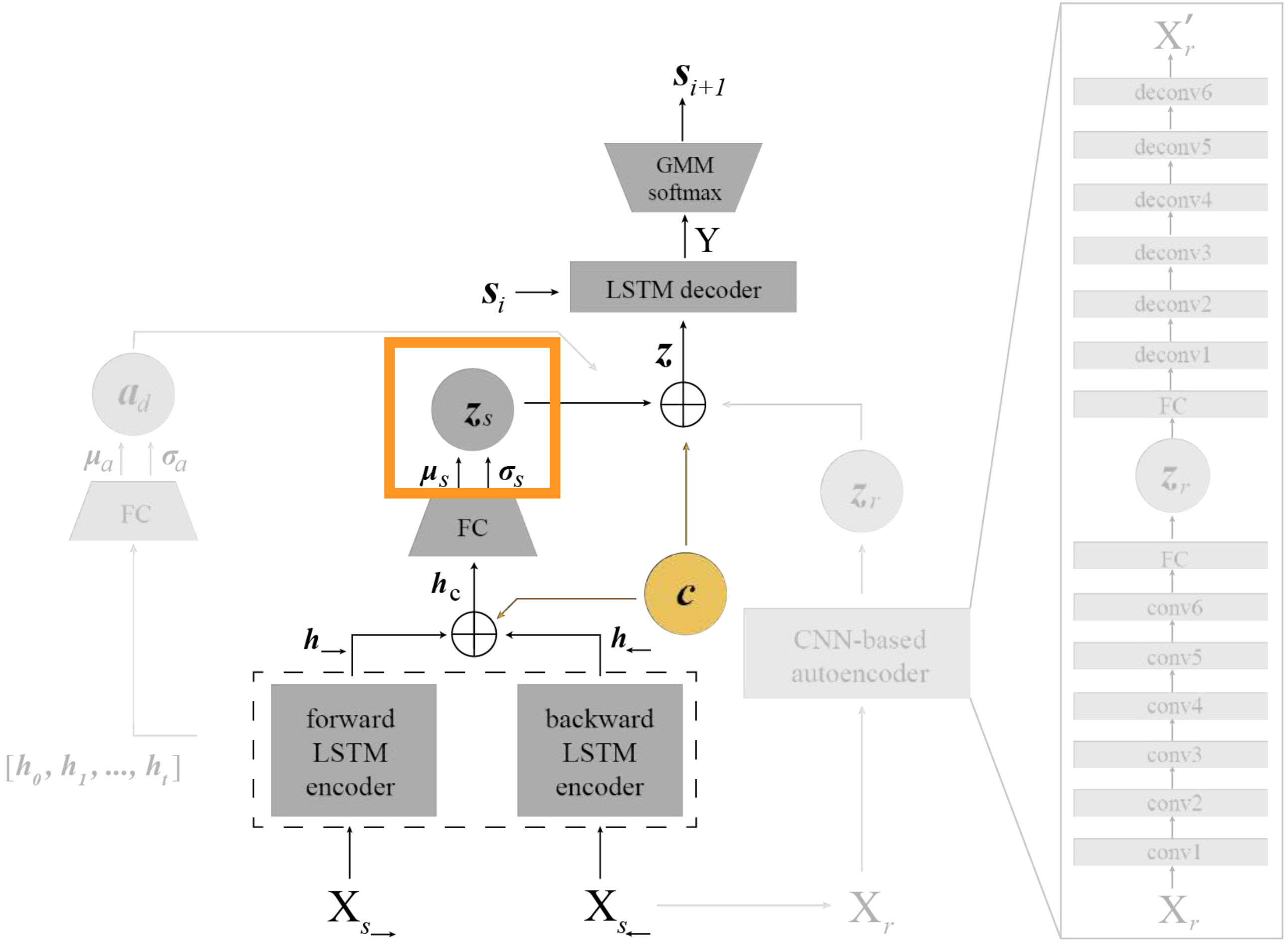
- **A conditional vector is used to ensure a high quality generation of sketches from multiple categories.**

$\boldsymbol{h}_c$ is further transformed into two vectors to capture the distributions of the training strokes:

$$\boldsymbol{\mu}_s = W_\mu \boldsymbol{h}_c + \boldsymbol{b}_\mu$$

$$\boldsymbol{\sigma}_s = exp\left(\frac{W_\sigma \boldsymbol{h}_c + \boldsymbol{b}_\sigma}{2}\right)$$

AI-SKETCHER - Conditional Sequence-to-Sequence VAE

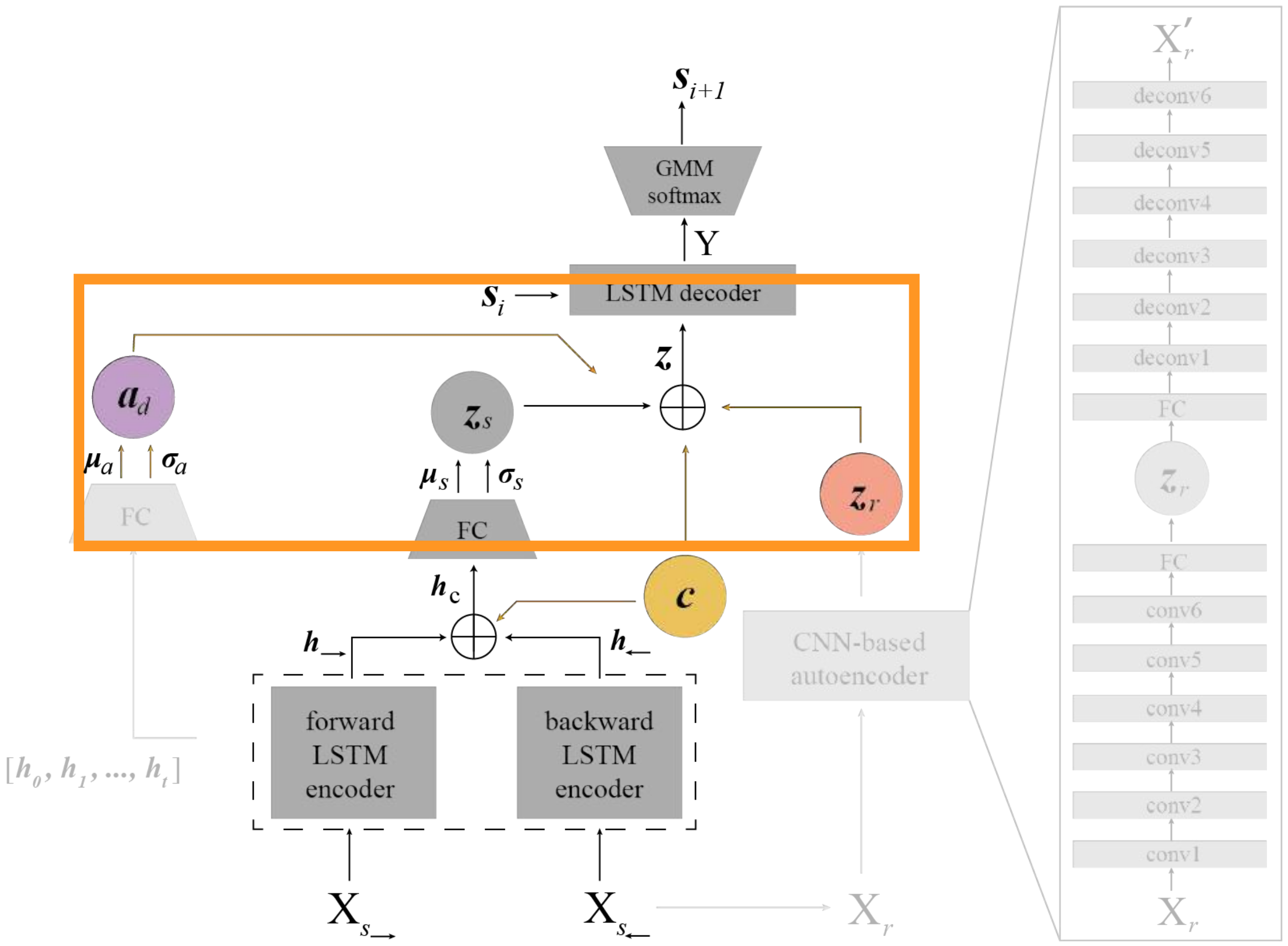


- **A conditional vector is used to ensure a high quality generation of sketches from multiple categories.**

A latent vector has been randomly sampled from the distributions for generating the next strokes:

$$z_s = \mu_s + \sigma_s \cdot \lambda$$

AI-SKETCHER - Conditional Sequence-to-Sequence VAE



- **A conditional vector is used to ensure a high quality generation of sketches from multiple categories.**

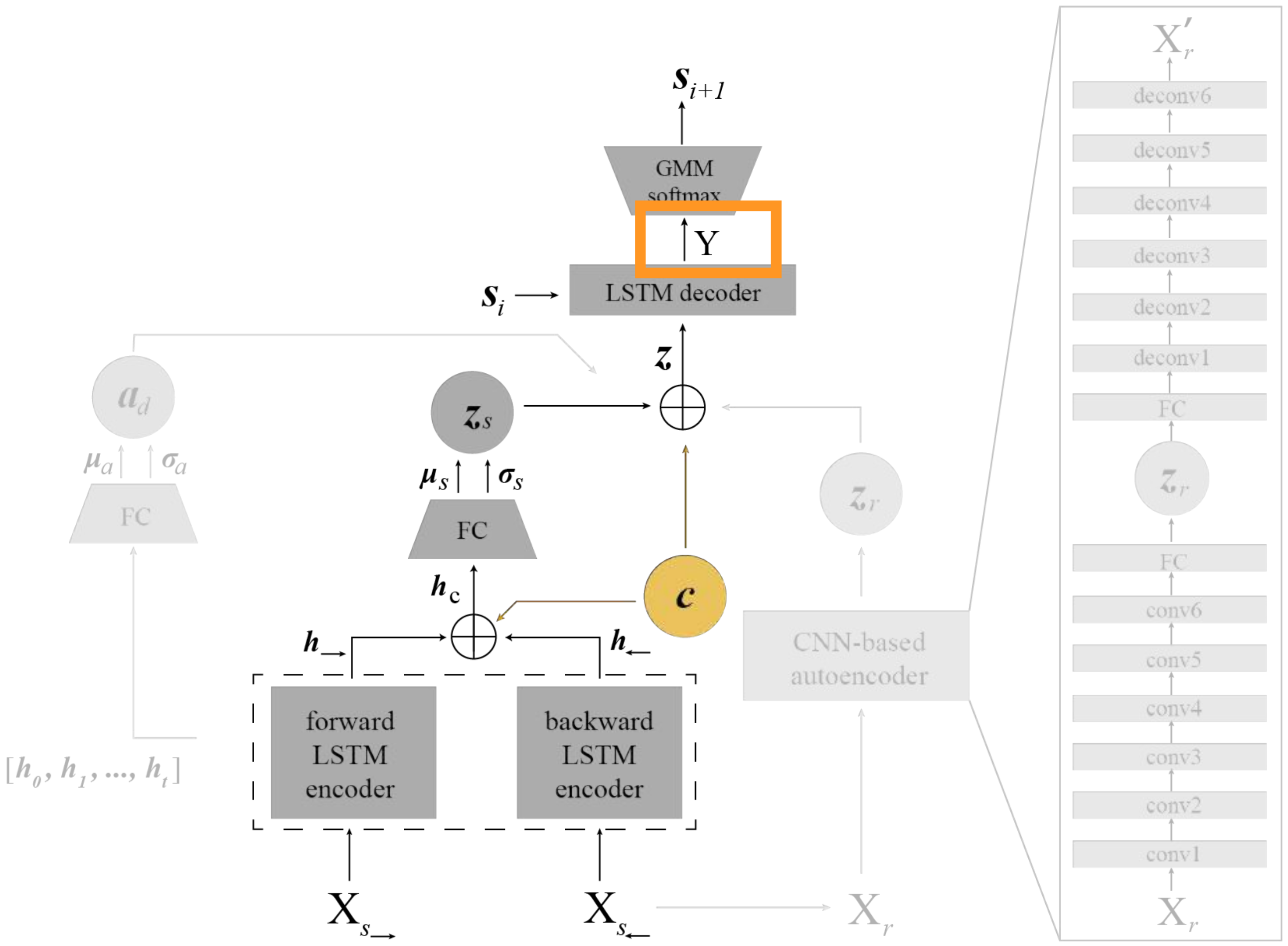
For decoding

z_s is concatenated together with the image feature vector z_r, the latent influence vector a_d, the conditional vector c and the last stroke vector S_i:

$$z = [z_s; z_r; \ a_d; \ c; \ s_i]$$

$$h^{dec} = decode(z)$$

AI-SKETCHER - Conditional Sequence-to-Sequence VAE

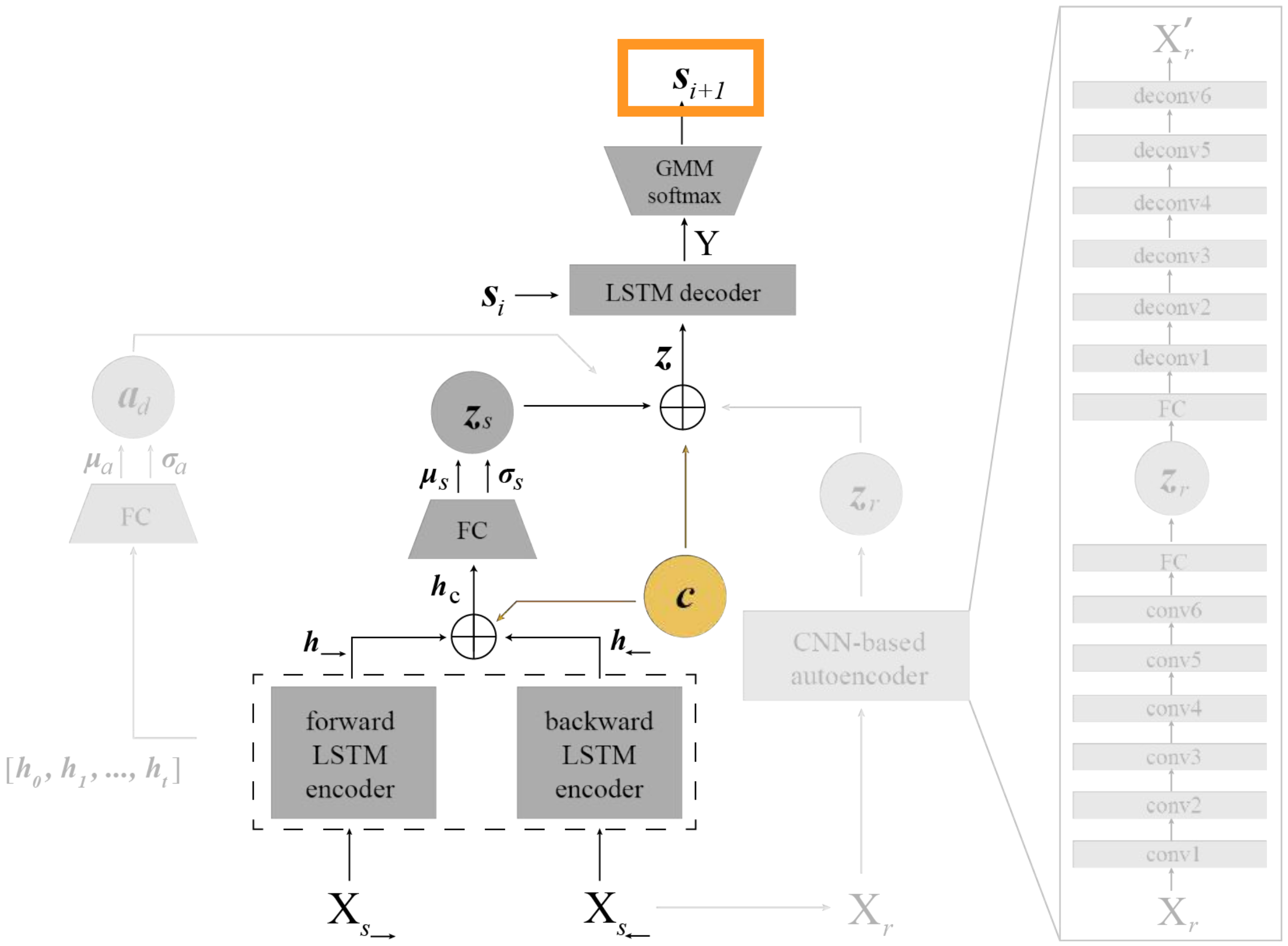


- **A conditional vector is used to ensure a high quality generation of sketches from multiple categories.**

Y: the parameters of a Gaussian mixture model (GMM) employed for **predicting the next stroke.**

$$Y = W_y \boldsymbol{h}^{dec} + \boldsymbol{b}_y$$

AI-SKETCHER - Conditional Sequence-to-Sequence VAE

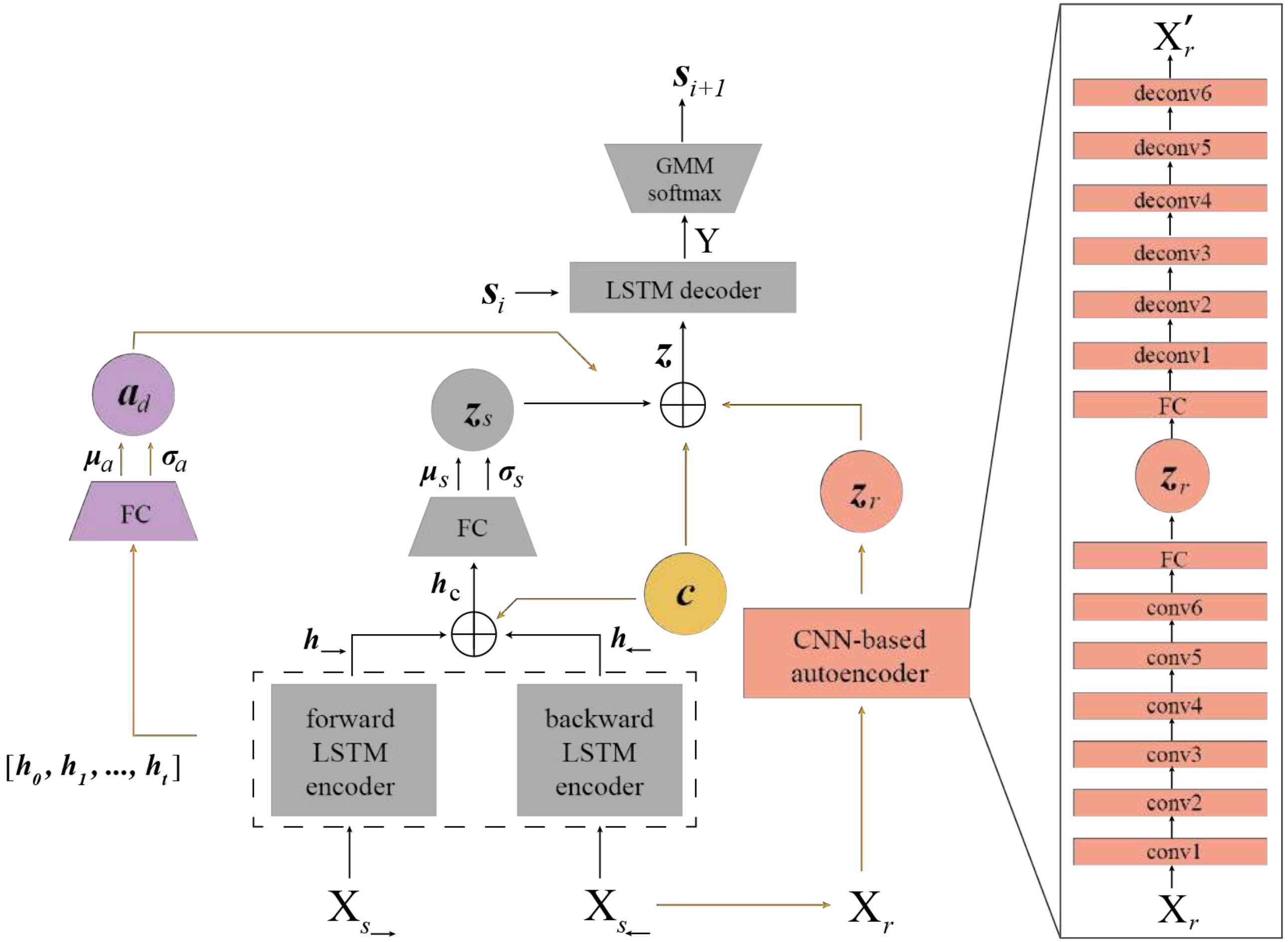


- A conditional vector is used to ensure a high quality generation of sketches from multiple categories.

predict the probabilities of the relative position of the next drawing point:

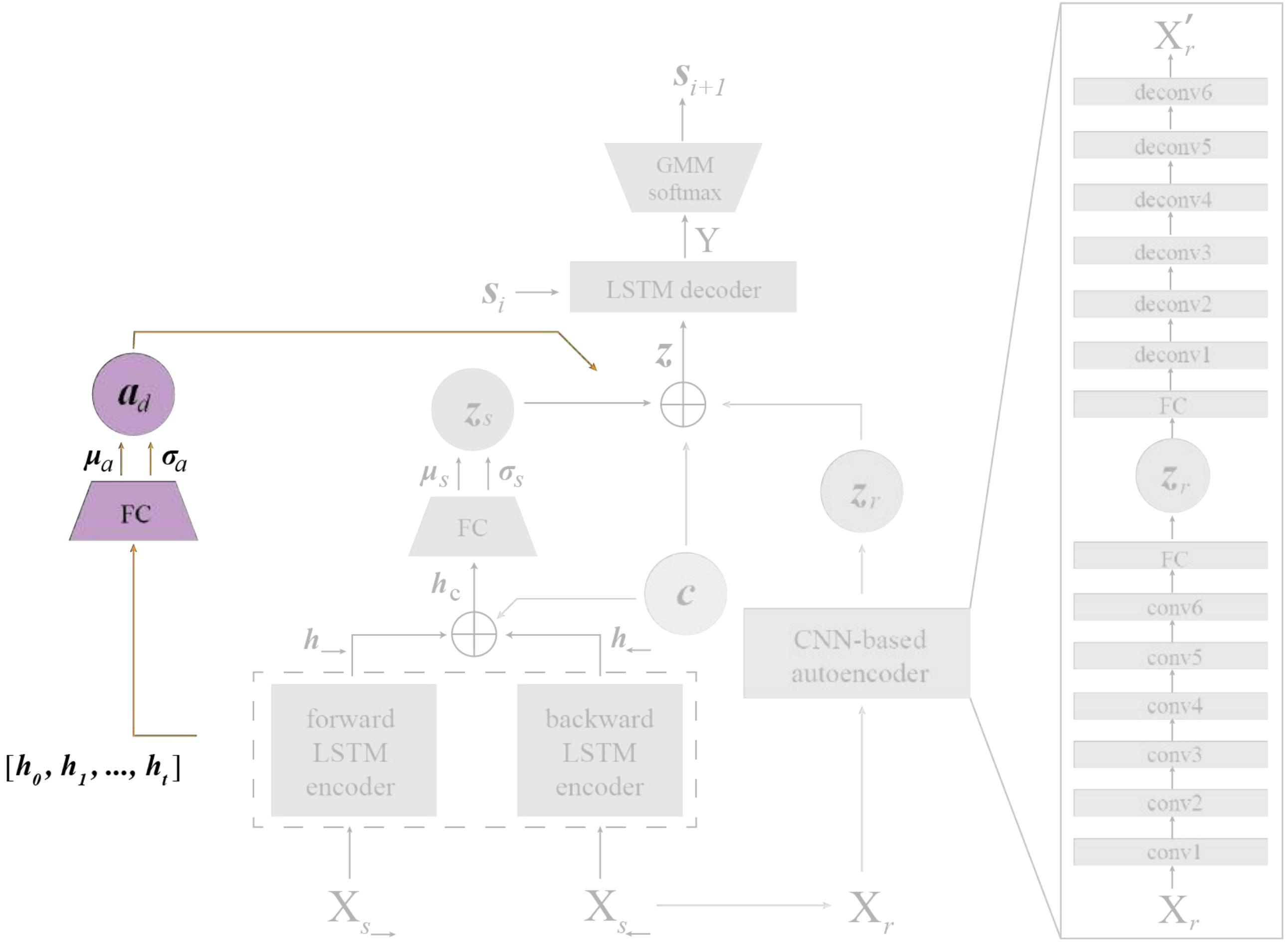
$$p(\Delta x_{i+1}, \Delta y_{i+1})$$

AI-SKETCHER



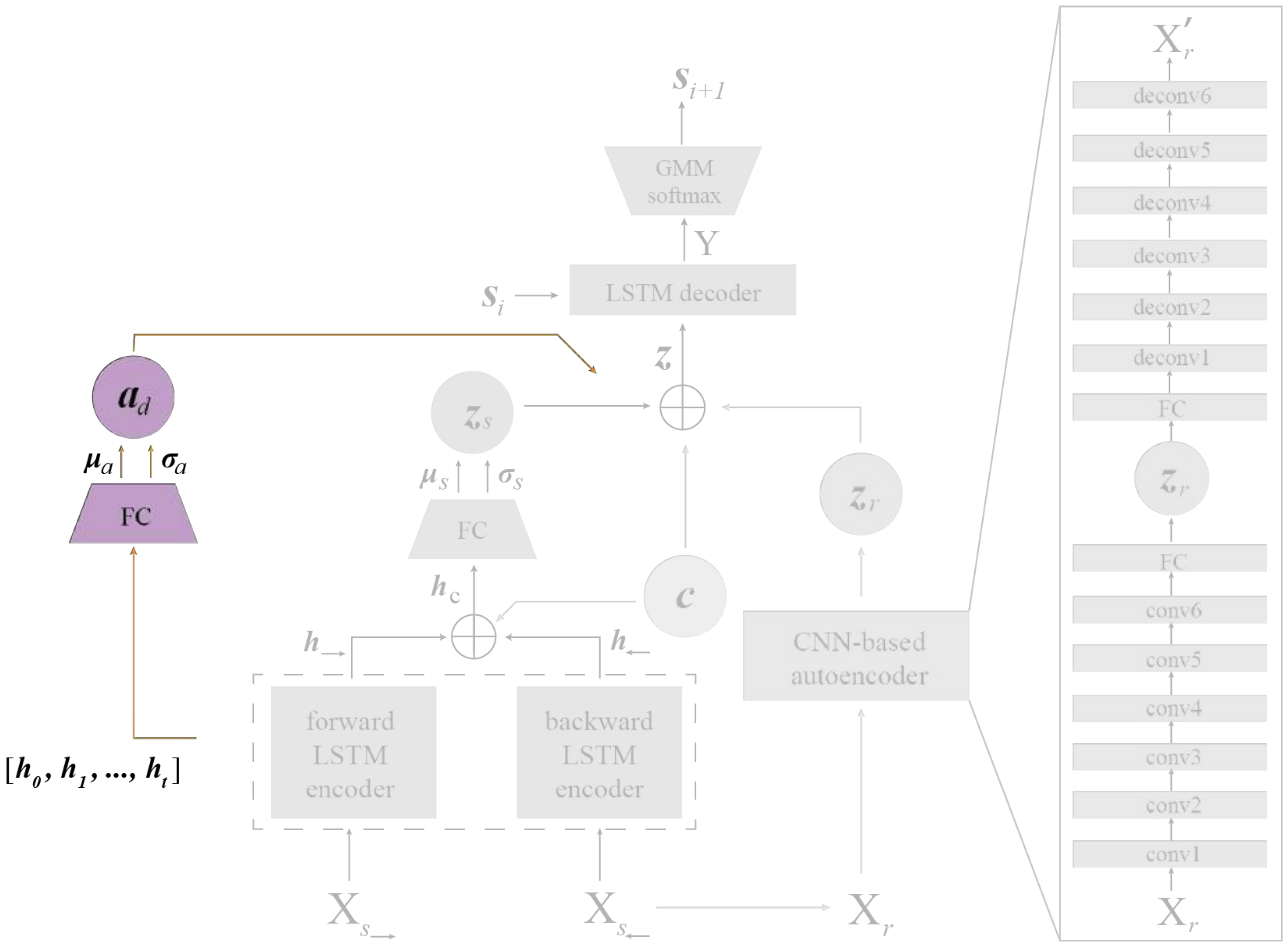
- **A conditional vector** is used to ensure a high quality generation of sketches from multiple categories.
- **A fully-connected layer** is introduced to estimate how the previous strokes will influence on the next stroke.
- **A CNN-based autoencoder** is employed to capture the spatial information of a training set.
- **Loss function** is modified.

AI-SKETCHER



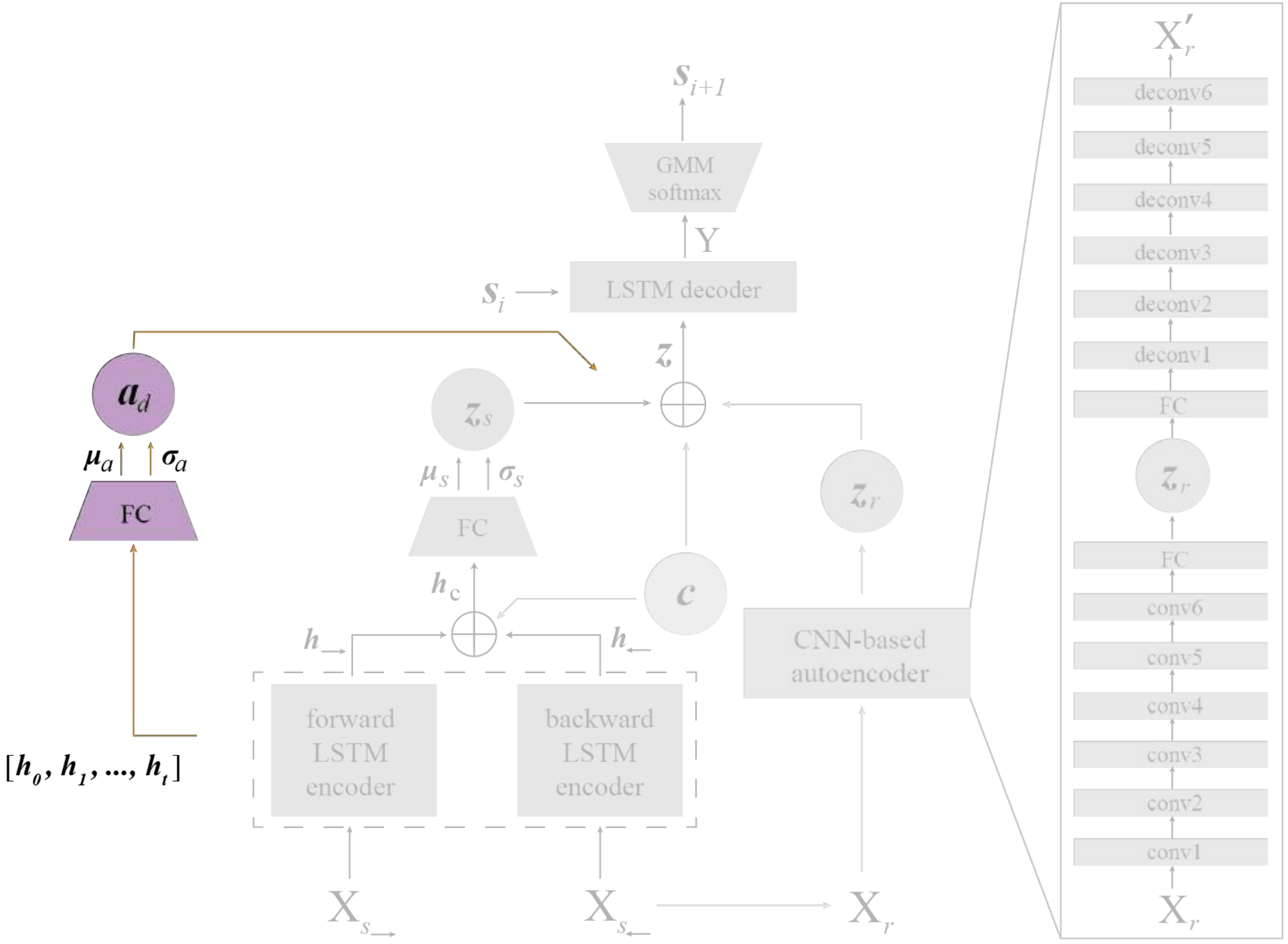
- **A conditional vector** is used to ensure a high quality generation of sketches from multiple categories.
- **A fully-connected layer is introduced to estimate how the previous strokes will influence on the next stroke.**
- **A CNN-based autoencoder** is employed to capture the spatial information of a training set.
- **Loss function** is modified.

AI-SKETCHER - Influence Layer



- A fully-connected layer is introduced to estimate how the previous strokes will influence on the next stroke.

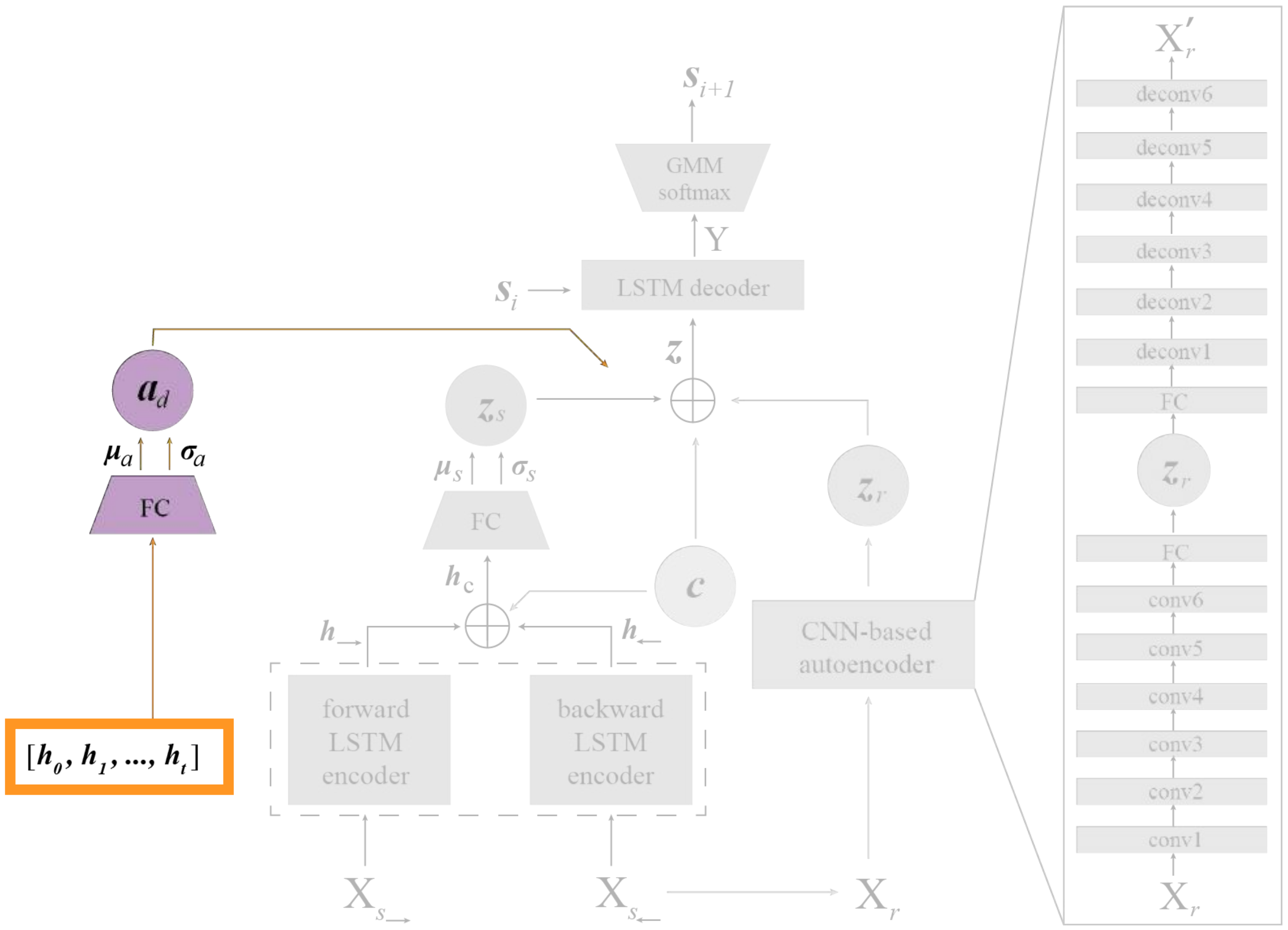
AI-SKETCHER - Influence Layer



- **A fully-connected layer is introduced to estimate how the previous strokes will influence on the next stroke.**

The influence layer is applied to enhance the influence of the input training data on the decoding process.

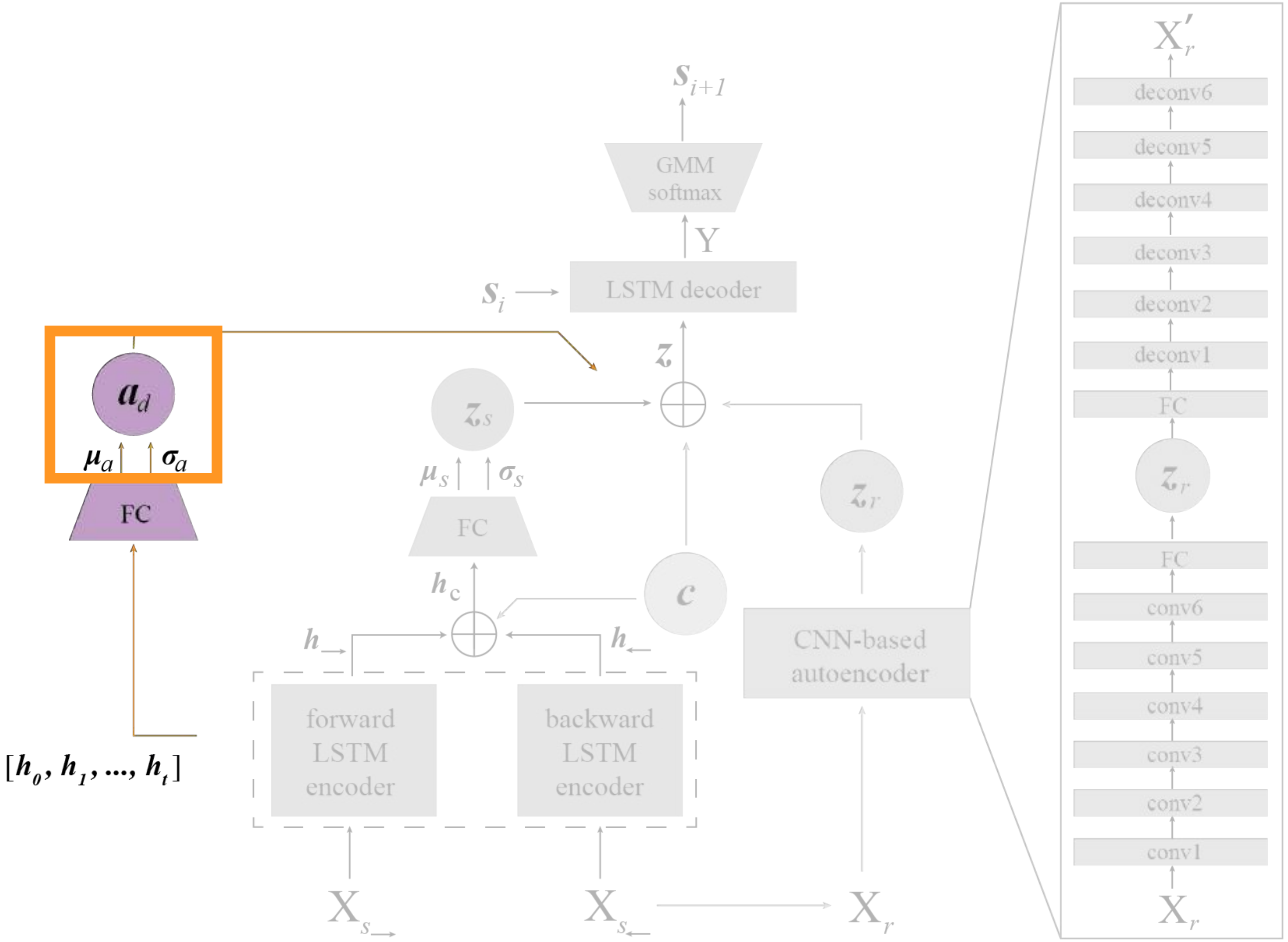
AI-SKETCHER - Influence Layer



- **A fully-connected layer is introduced to estimate how the previous strokes will influence on the next stroke.**

It considers **all the previous hidden node values** until the latest drawing step in the RNN encoder.

AI-SKETCHER - Influence Layer

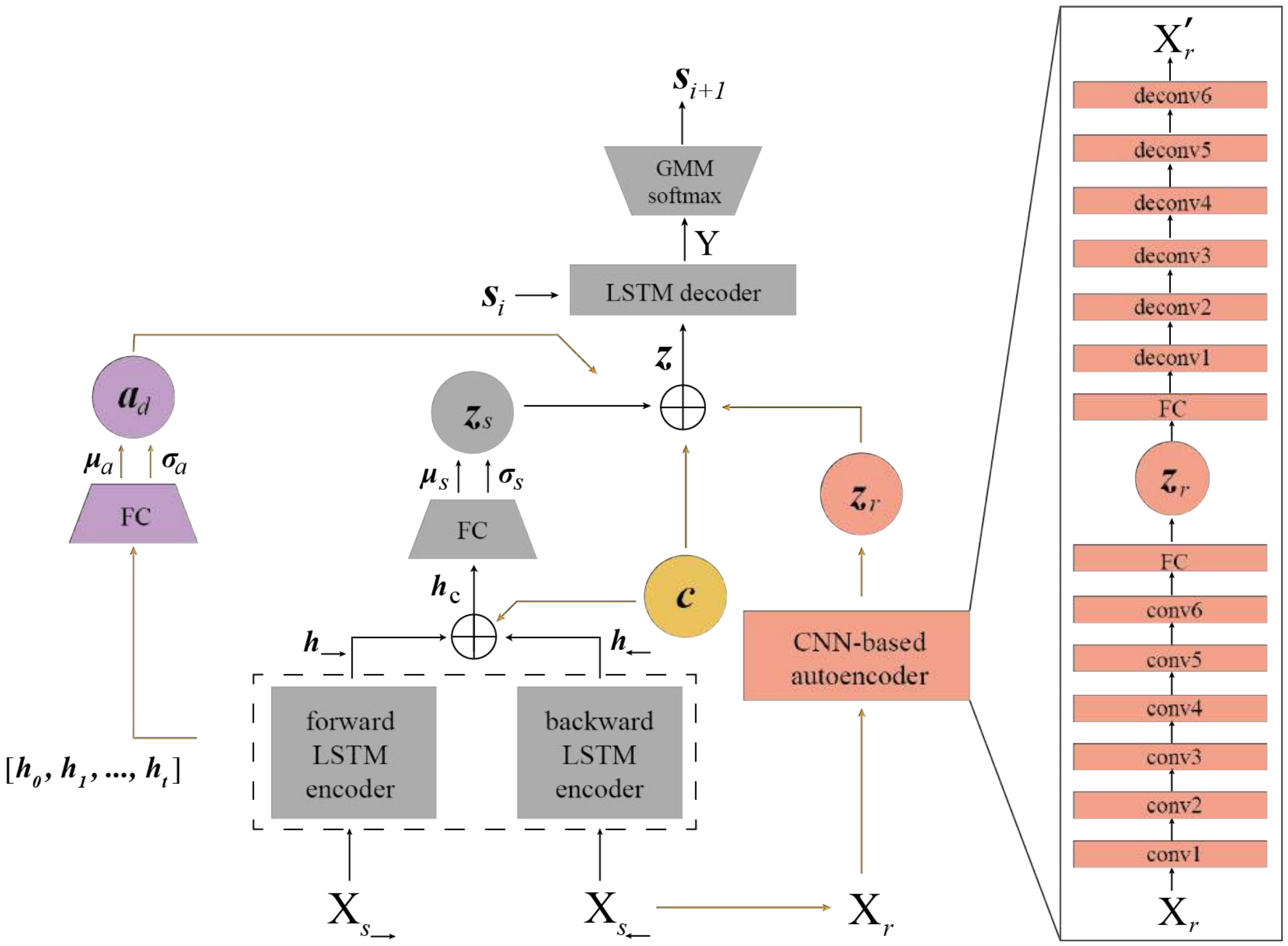


- **A fully-connected layer is introduced to estimate how the previous strokes will influence on the next stroke.**

The influence vector a_d is a latent vector whose fields are sampled from these normal distributions:

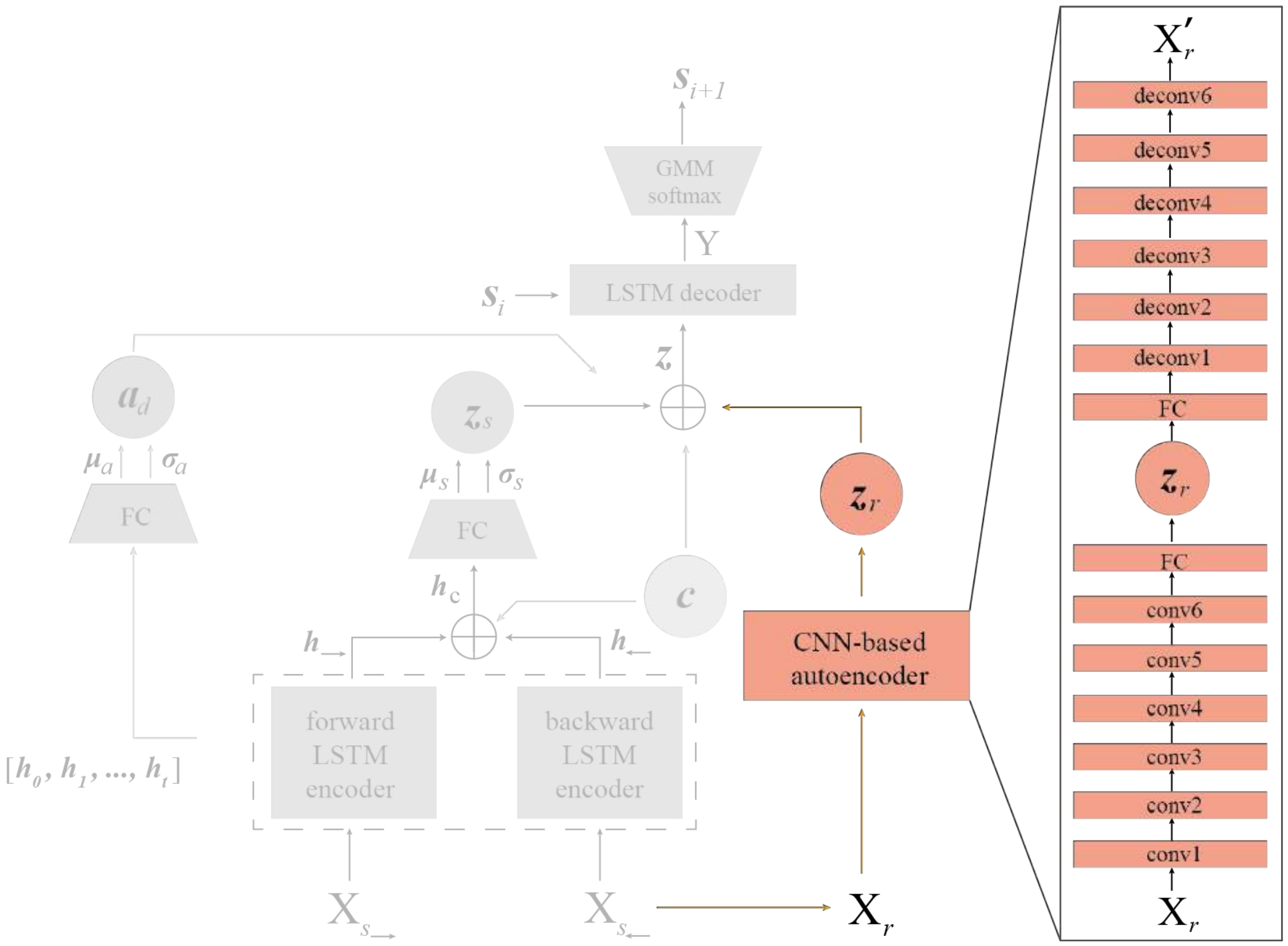
$$a_d = \mu_a + \sigma_a \cdot \lambda_a$$

AI-SKETCHER



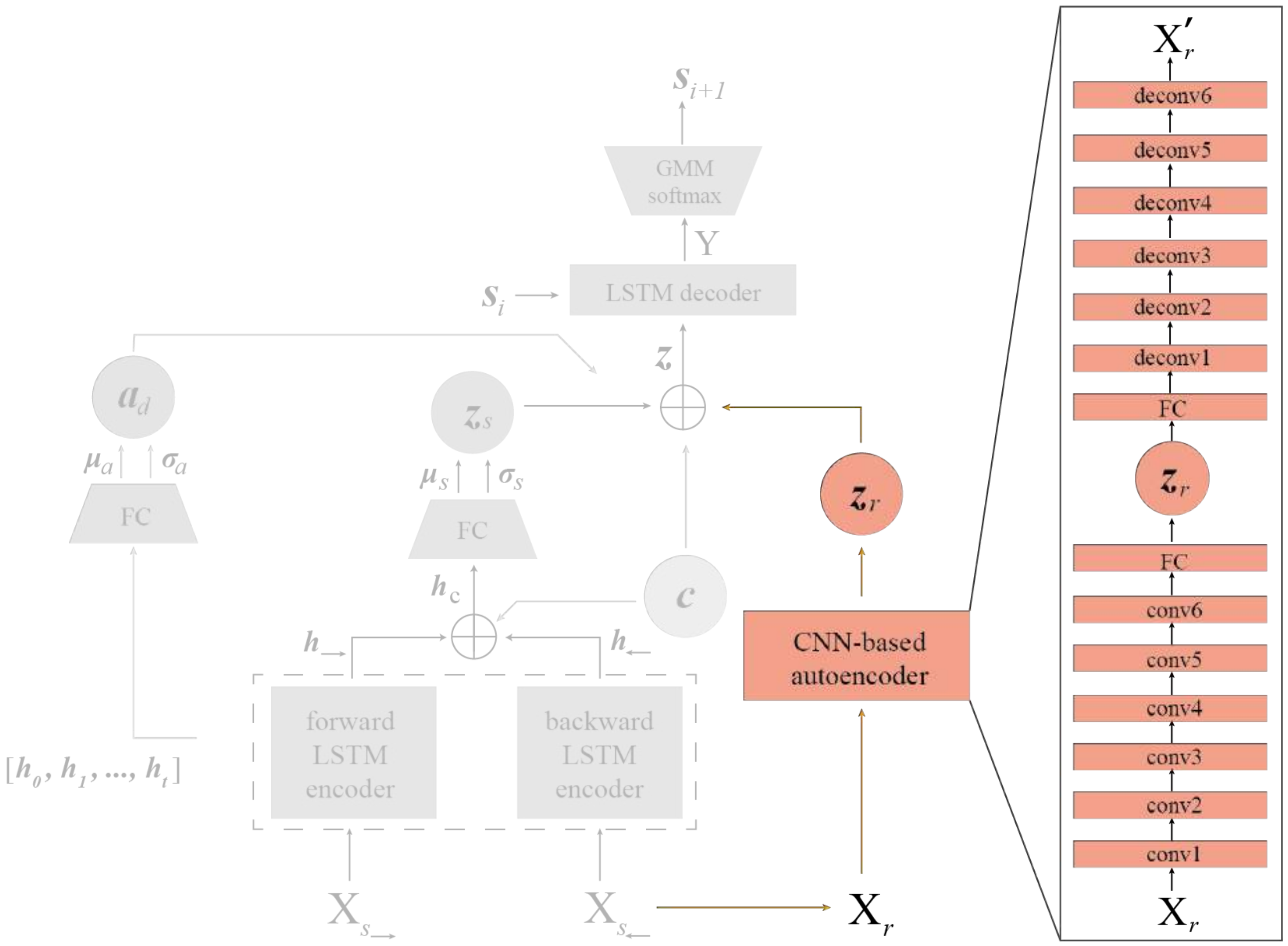
- **A conditional vector** is used to ensure a high quality generation of sketches from multiple categories.
- **A fully-connected layer** is introduced to estimate how the previous strokes will influence on the next stroke.
- **A CNN-based autoencoder** is employed to capture the spatial information of a training set.
- **Loss function** is modified.

AI-SKETCHER



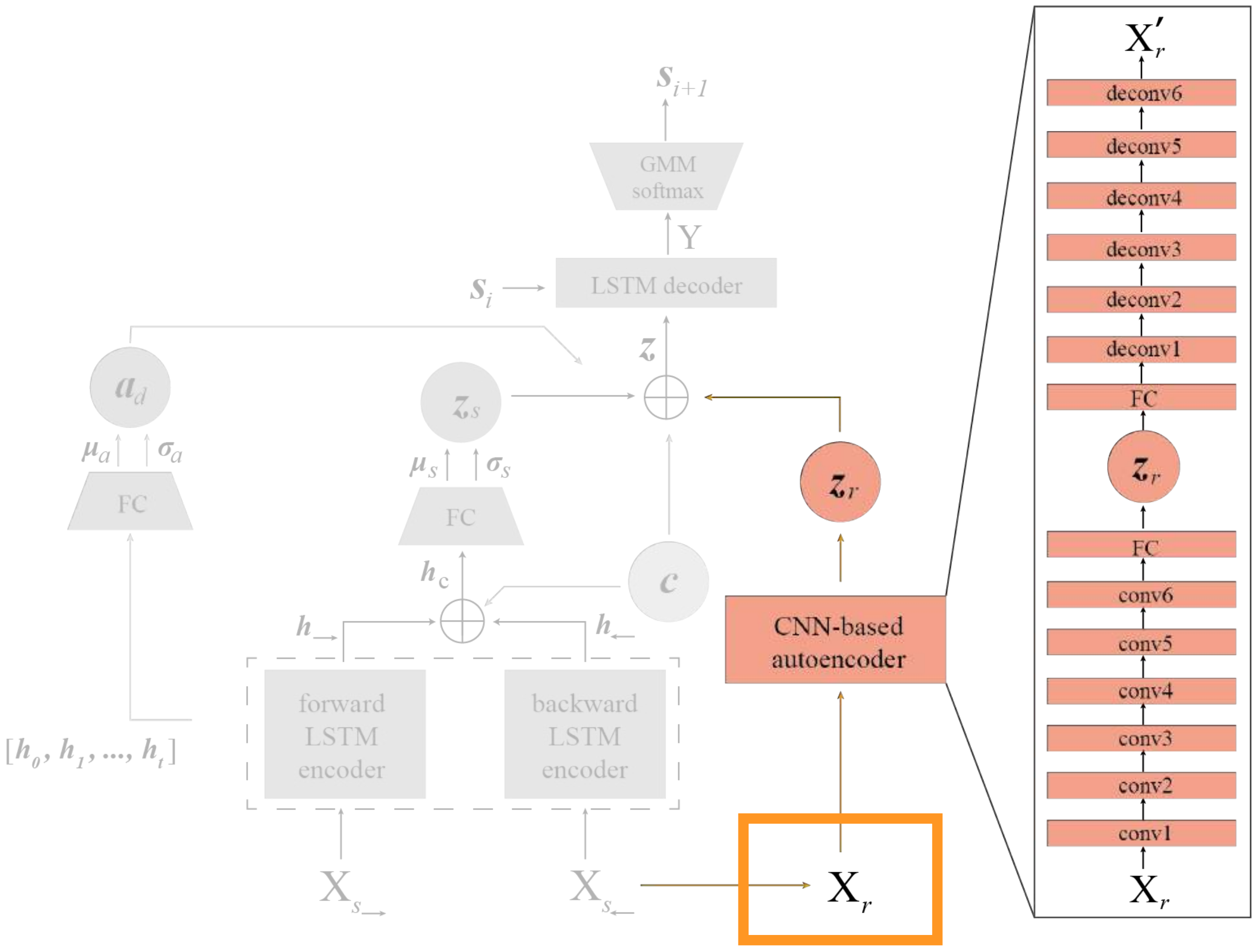
- **A conditional vector** is used to ensure a high quality generation of sketches from multiple categories.
- **A fully-connected layer** is introduced to estimate how the previous strokes will influence on the next stroke.
- **A CNN-based autoencoder is employed to capture the spatial information of a training set.**
- **Loss function** is modified.

AI-SKETCHER - CNN-based Autoencoder



- A CNN-based autoencoder is employed to capture the spatial information of a training set.

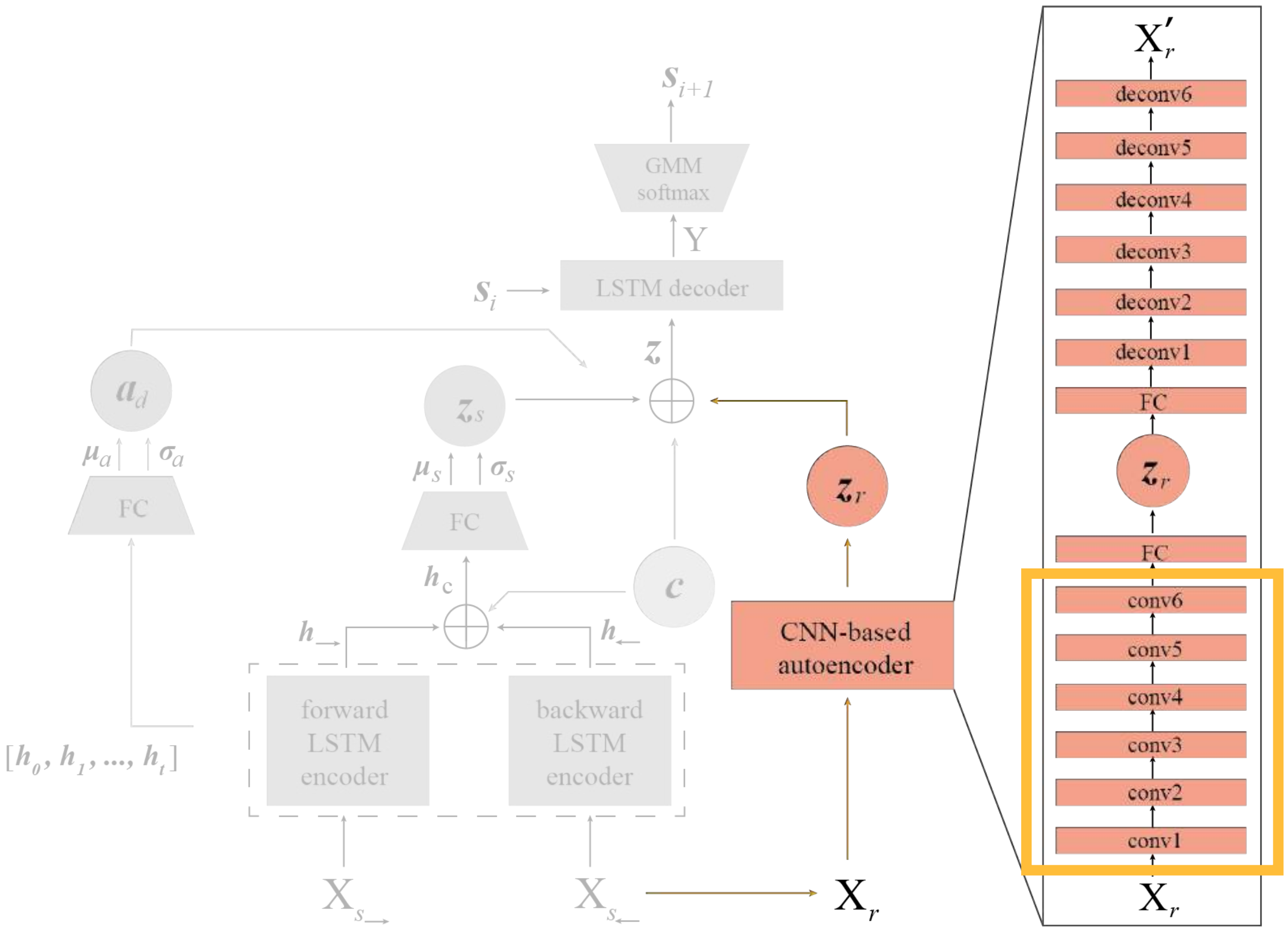
AI-SKETCHER - CNN-based Autoencoder



- **A CNN-based autoencoder is employed to capture the spatial information of a training set.**

X_r: the input raster image matrix.

AI-SKETCHER - CNN-based Autoencoder

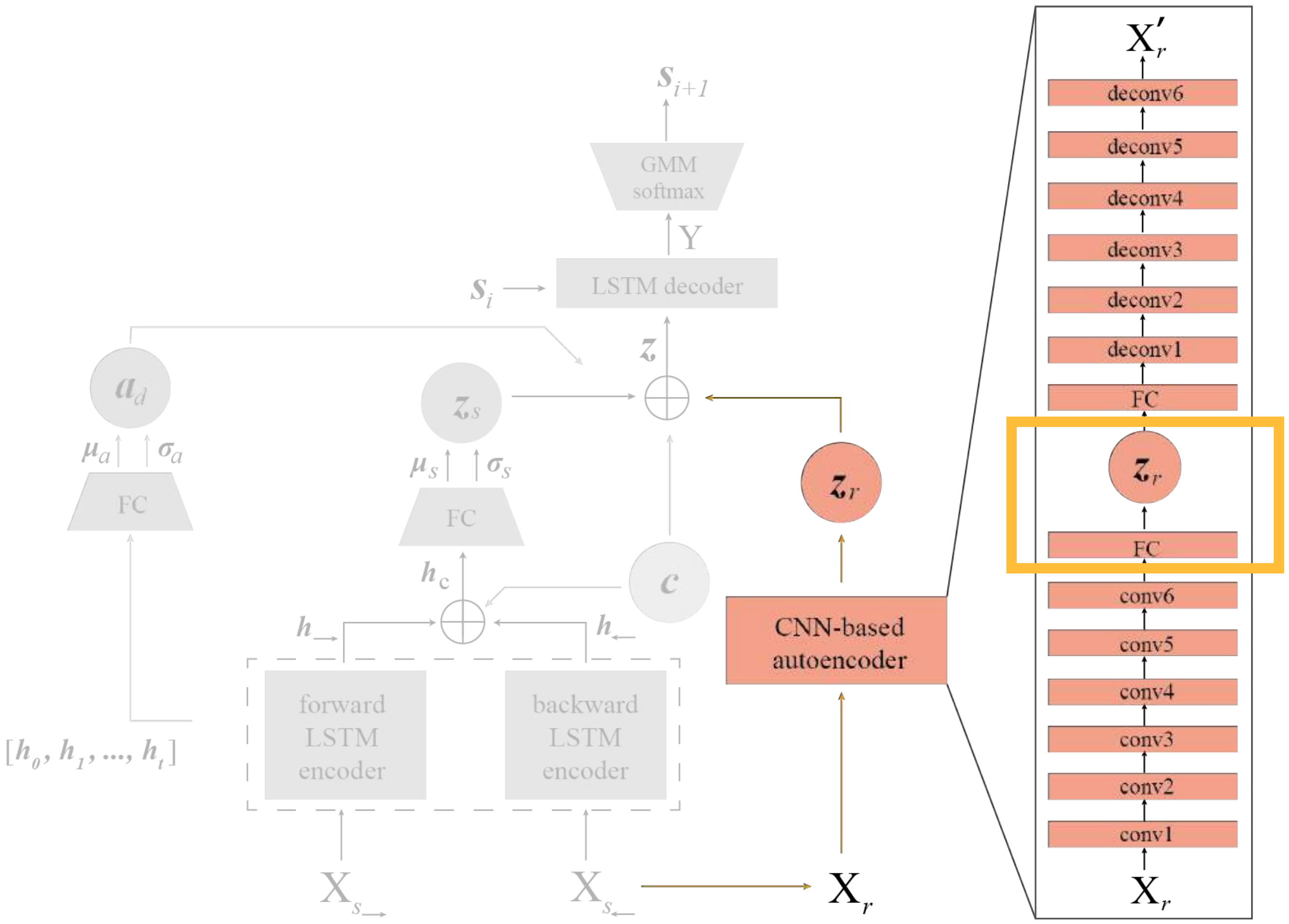


- **A CNN-based autoencoder is employed to capture the spatial information of a training set.**

Encoder:

- three convolutional layers with the stride size as 2.
- the other three layers with the stride size as 1.

AI-SKETCHER - CNN-based Autoencoder

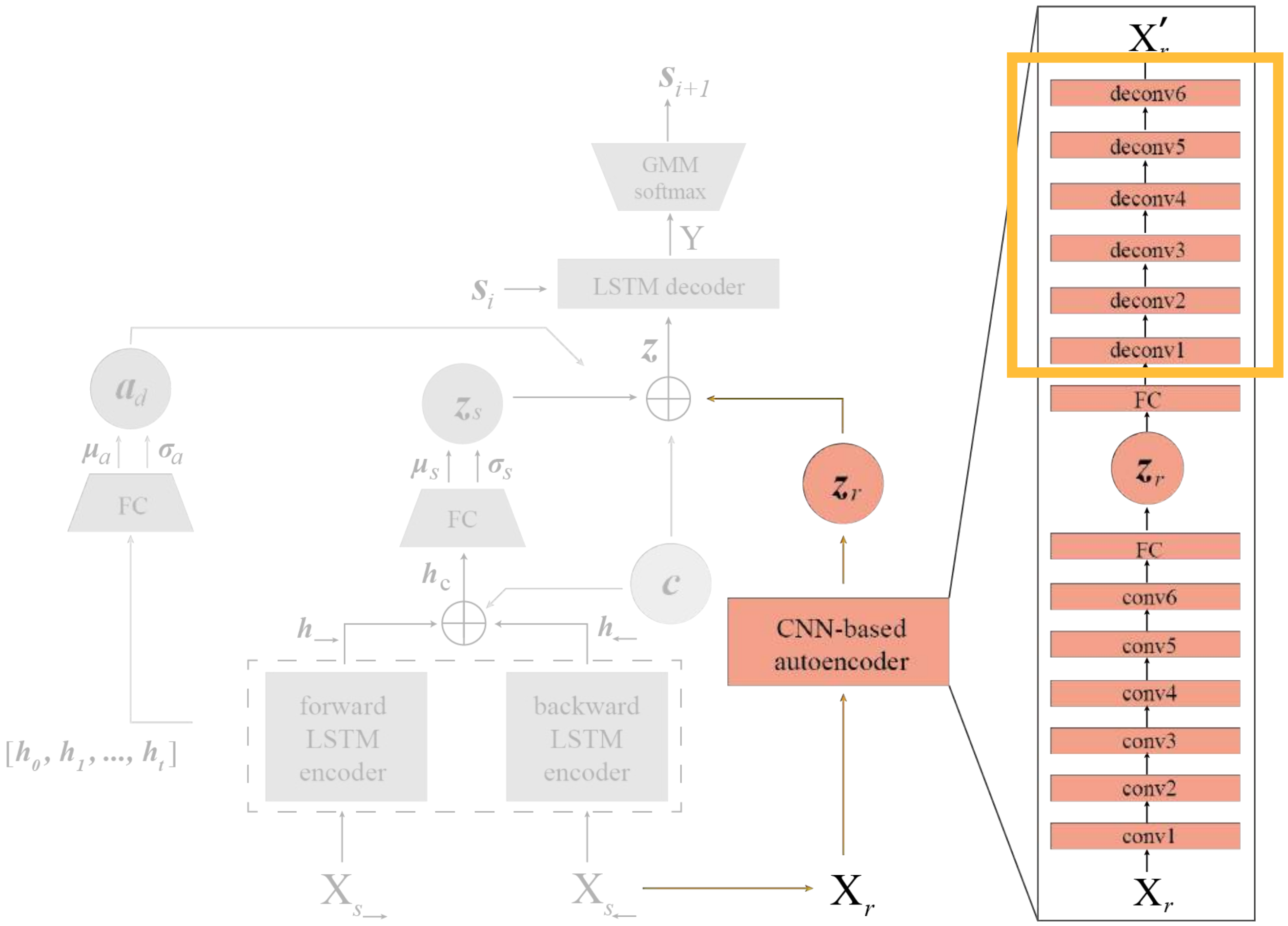


- **A CNN-based autoencoder is employed to capture the spatial information of a training set.**

Encoder:

- The last layer is a fully-connected neural network to produce the latent feature vector z_r with 128 dimensions.

AI-SKETCHER - CNN-based Autoencoder

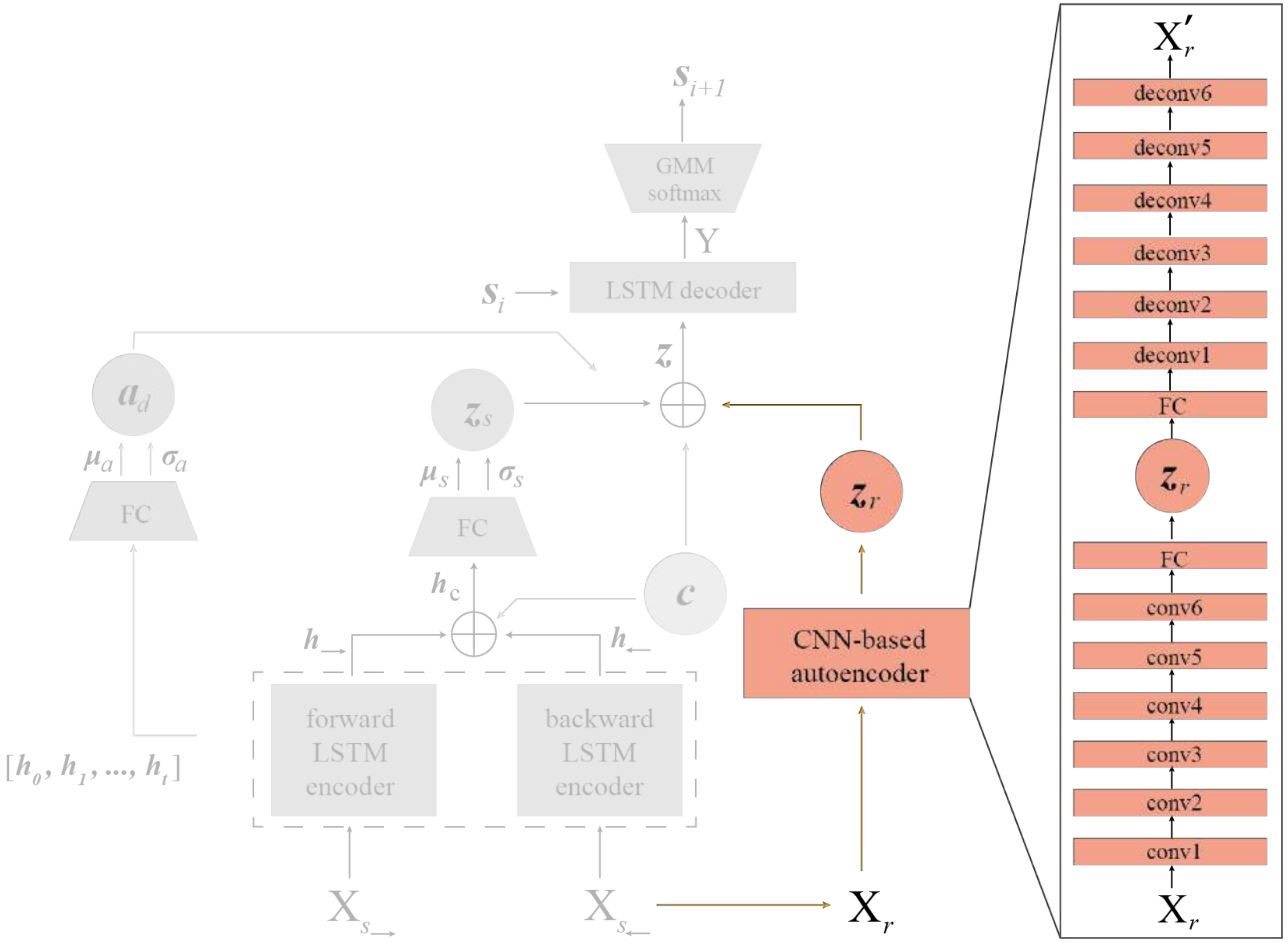


- **A CNN-based autoencoder is employed to capture the spatial information of a training set.**

Decoder:

- three deconvolutional layers with stride sizes equal to 2.
- the other three layers with stride sizes equal to 1.

AI-SKETCHER - CNN-based Autoencoder

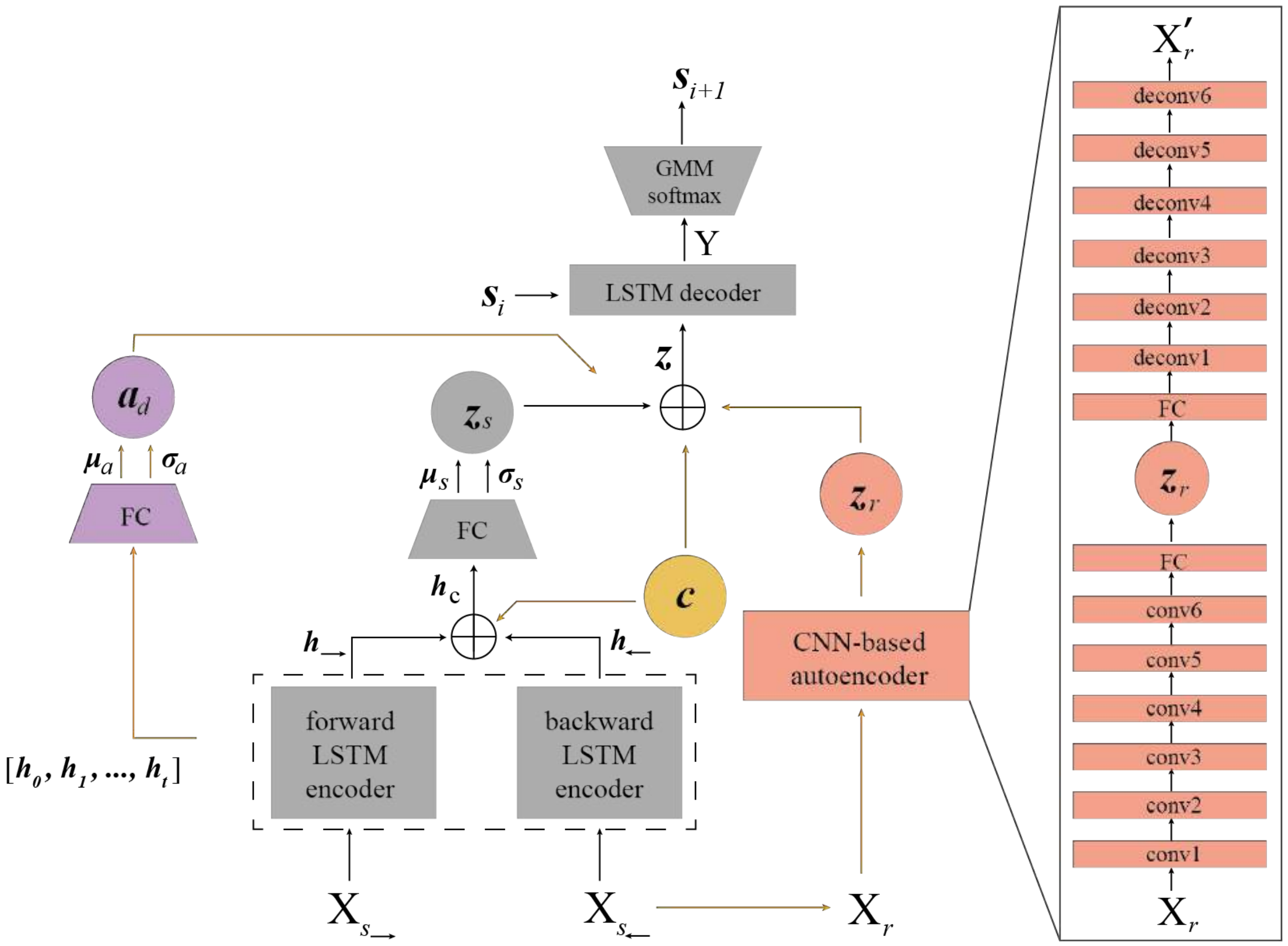


- **A CNN-based autoencoder is employed to capture the spatial information of a training set.**

ReLU is used as the activation function in both convolutional and deconvolutional layers.

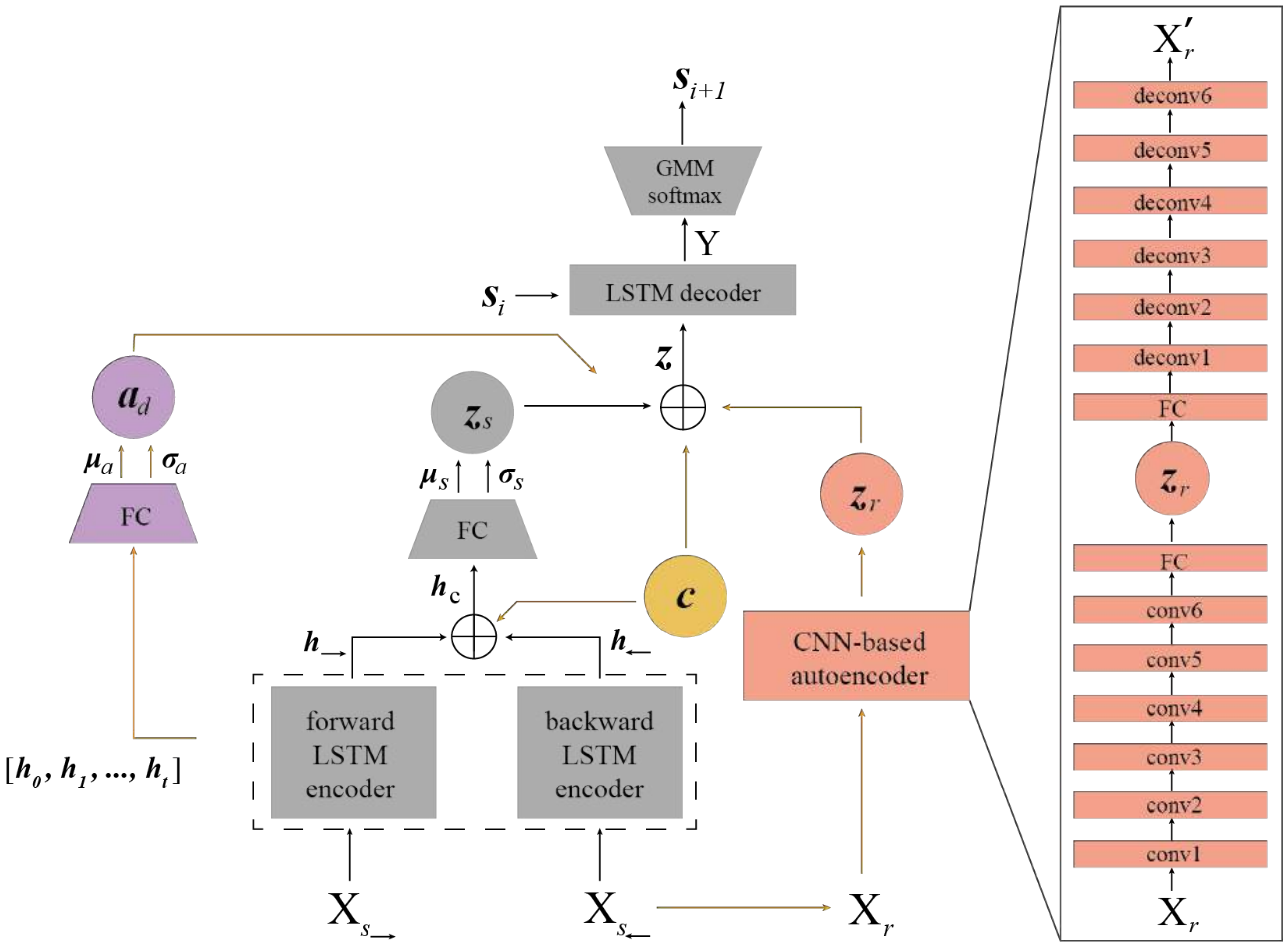
tanh is used as the activation function of the fully-connected neural network.

AI-SKETCHER



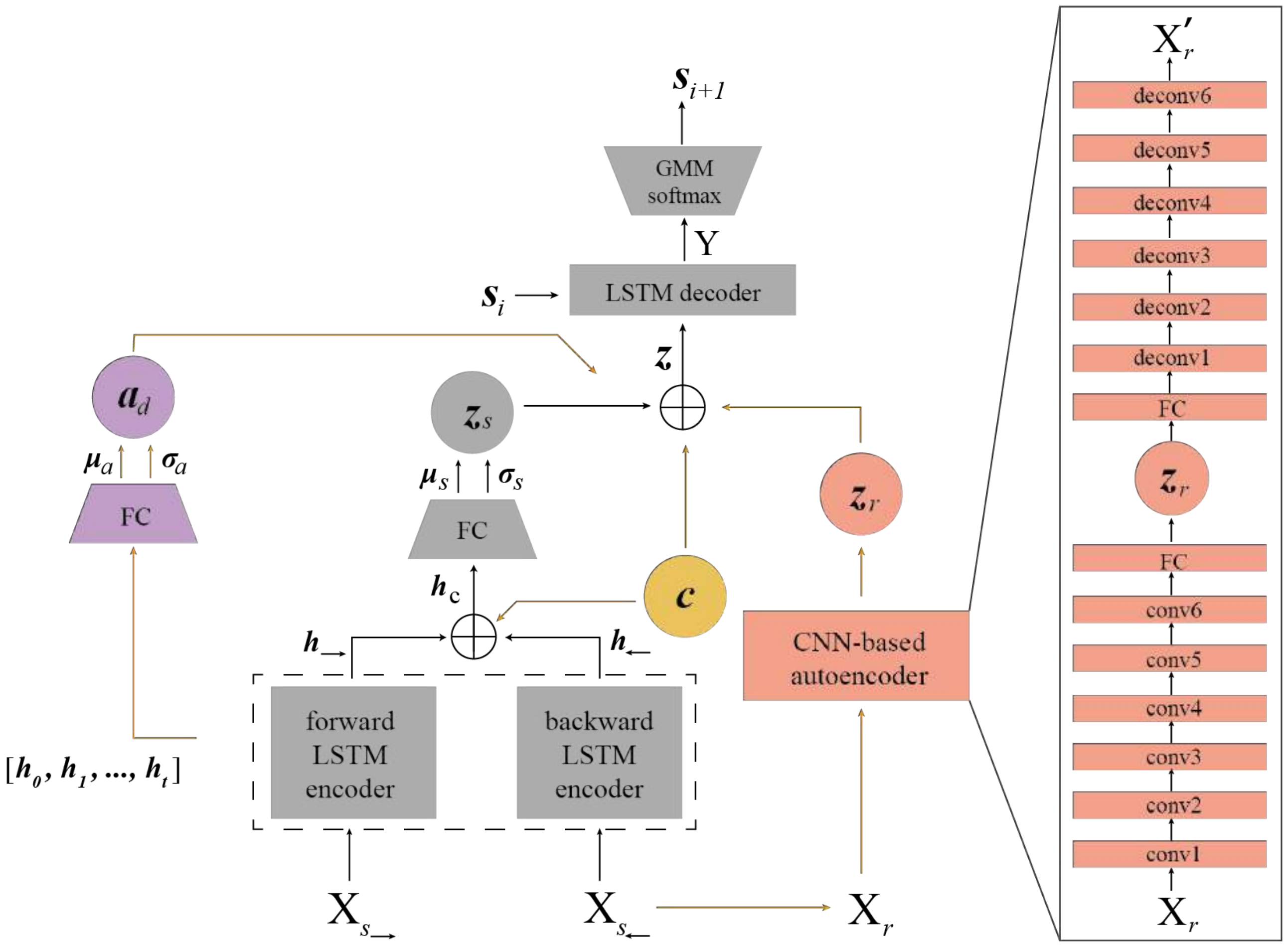
- **A conditional vector** is used to ensure a high quality generation of sketches from multiple categories.
- **A fully-connected layer** is introduced to estimate how the previous strokes will influence on the next stroke.
- **A CNN-based autoencoder** is employed to capture the spatial information of a training set.
- **Loss function** is modified.

AI-SKETCHER



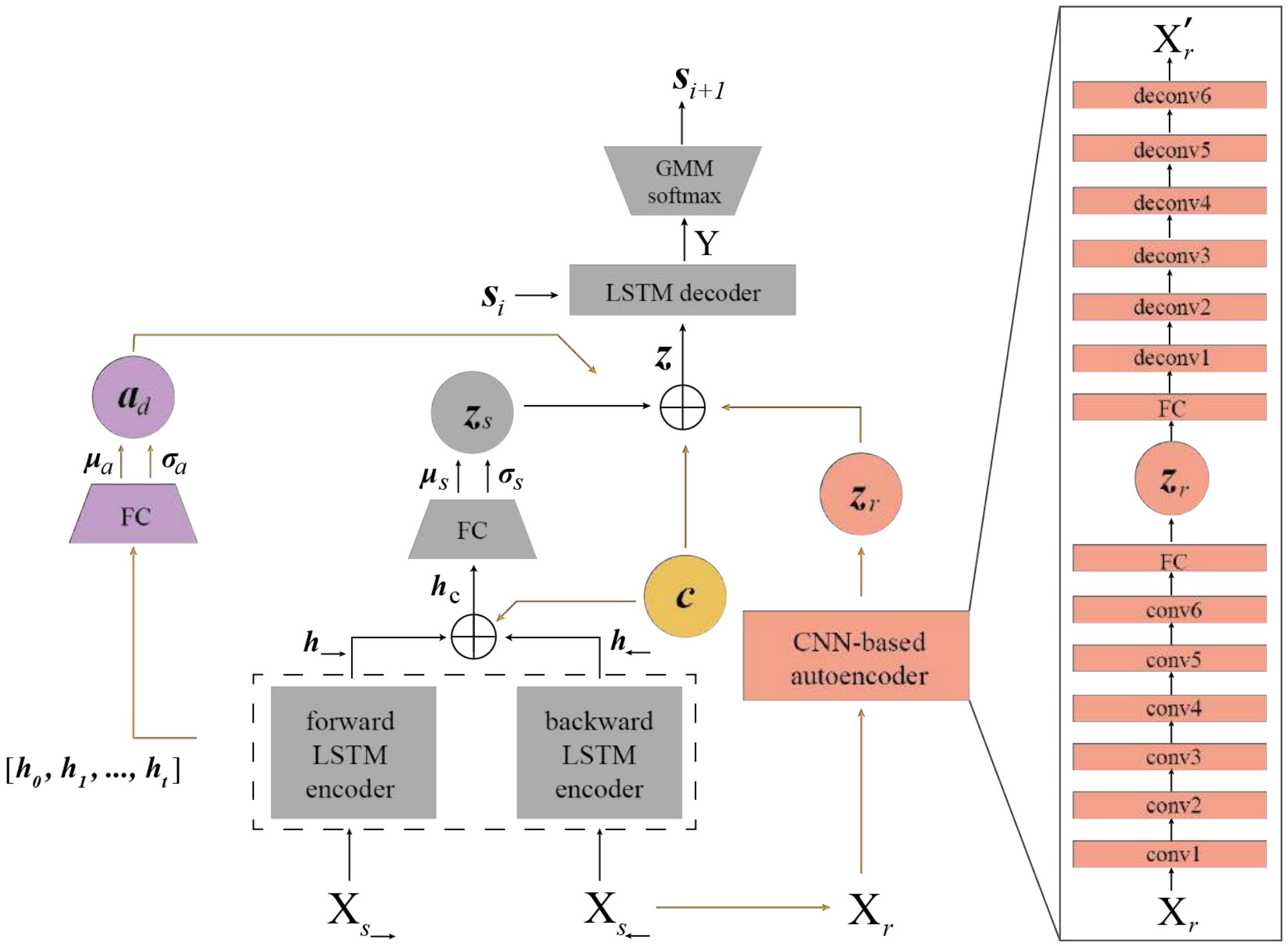
- **A conditional vector** is used to ensure a high quality generation of sketches from multiple categories.
- **A fully-connected layer** is introduced to estimate how the previous strokes will influence on the next stroke.
- **A CNN-based autoencoder** is employed to capture the spatial information of a training set.
- **Loss function is modified.**

AI-SKETCHER - Loss function



- **Loss function is modified.**

AI-SKETCHER - Loss function



- **Loss function is modified.**

$$Loss = l_r + \alpha \cdot max(l_{kl}, \epsilon)$$

AI-SKETCHER - Loss function

- **Loss function is modified.**

$$Loss = \underline{l_r} + \underline{\alpha \cdot max(\boxed{l_{kl}}, \epsilon)}$$

the reconstruction loss
estimates the differences between the generated strokes and the training samples.

estimates the distribution differences between the generated strokes and the strokes in the training set.

AI-SKETCHER - Loss function

- **Loss function is modified.**

$$Loss = l_r + \alpha \cdot max(\boxed{l_{kl}}, \epsilon)$$

$$\begin{aligned}
l_z &= -\frac{1}{2n_z}\sum_{i=1}^{n_z}(1 + \boldsymbol{\sigma}_{s_i} - exp(\boldsymbol{\sigma}_{s_i}) - \boldsymbol{\mu}_{s_i}^2) \\
l_a &= -\frac{1}{2n_a}\sum_{j=1}^{n_a}(1 + \boldsymbol{\sigma}_{a_j} - exp(\boldsymbol{\sigma}_{a_j}) - \boldsymbol{\mu}_{a_j}^2) \\
l_{kl} &= l_z + \beta l_a
\end{aligned}$$

AI-SKETCHER - Loss function

- **Loss function is modified.**

$$Loss = l_r + \alpha \cdot max(\boxed{l_{kl}}, \epsilon)$$

$$\begin{aligned} l_z &= -\frac{1}{2n_z}\sum_{i=1}^{n_z}(1+\boldsymbol{\sigma}_{s_i} - exp(\boldsymbol{\sigma}_{s_i}) - \boldsymbol{\mu}_{s_i}^2) \\ l_a &= -\frac{1}{2n_a}\sum_{j=1}^{n_a}(1+\boldsymbol{\sigma}_{a_j} - exp(\boldsymbol{\sigma}_{a_j}) - \boldsymbol{\mu}_{a_j}^2) \\ l_{kl} &= l_z + \beta l_a \end{aligned}$$

n_a: the dimensions of a_d

AI-SKETCHER - Loss function

- **Loss function is modified.**

$$Loss = l_r + \alpha \cdot max(\boxed{l_{kl}}, \epsilon)$$

$$\begin{aligned} l_z &= -\frac{1}{2n_z}\sum_{i=1}^{n_z}(1+\boldsymbol{\sigma}_{s_i} - exp(\boldsymbol{\sigma}_{s_i}) - \boldsymbol{\mu}_{s_i}^2) \\ l_a &= -\frac{1}{2n_a}\sum_{j=1}^{n_a}(1+\boldsymbol{\sigma}_{a_j} - exp(\boldsymbol{\sigma}_{a_j}) - \boldsymbol{\mu}_{a_j}^2) \\ l_{kl} &= l_z + \beta l_a \end{aligned}$$

n_a: the dimensions of a_d

l_z / l_a: the KL divergence between the distribution of the latent vector z_s / a_d and the distribution of the strokes in the training data

AI-SKETCHER - Loss function

- **Loss function is modified.**

$$Loss = l_r + \alpha \cdot max(\boxed{l_{kl}}, \epsilon)$$

$$\begin{aligned} l_z &= -\frac{1}{2n_z}\sum_{i=1}^{n_z}(1+\boldsymbol{\sigma}_{s_i} - exp(\boldsymbol{\sigma}_{s_i}) - \boldsymbol{\mu}_{s_i}^2) \\ l_a &= -\frac{1}{2n_a}\sum_{j=1}^{n_a}(1+\boldsymbol{\sigma}_{a_j} - exp(\boldsymbol{\sigma}_{a_j}) - \boldsymbol{\mu}_{a_j}^2) \\ l_{kl} &= l_z + \beta l_a \end{aligned}$$

n_a: the dimensions of a_d

l_z / l_a: the KL divergence between the distribution of the latent vector z_s / a_d and the distribution of the strokes in the training data

β: a hyperparameter that balances the two terms

EVALUATION

We performed three experiments in purpose of validating:

- the AI-Sketcher's **drawing quality**
- its capability of generating sketches from **multiple classes**
- generation **diversity**

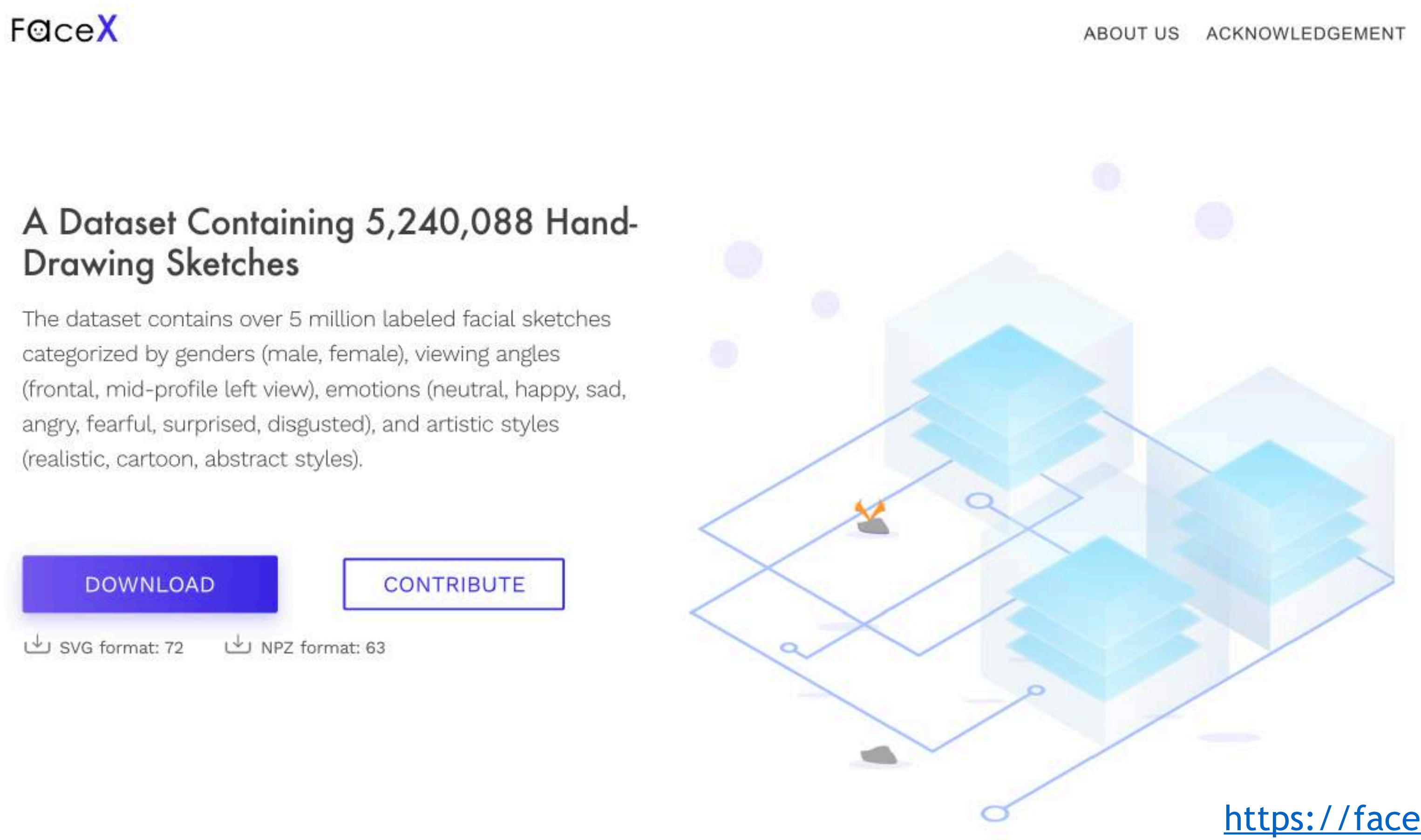
EVALUATION - Dataset

The QuickDraw dataset contains over 50 million sketches in 75 object categories and originally used for training Sketch-RNN.

https://quickdraw.withgoogle.com/

EVALUATION - Dataset

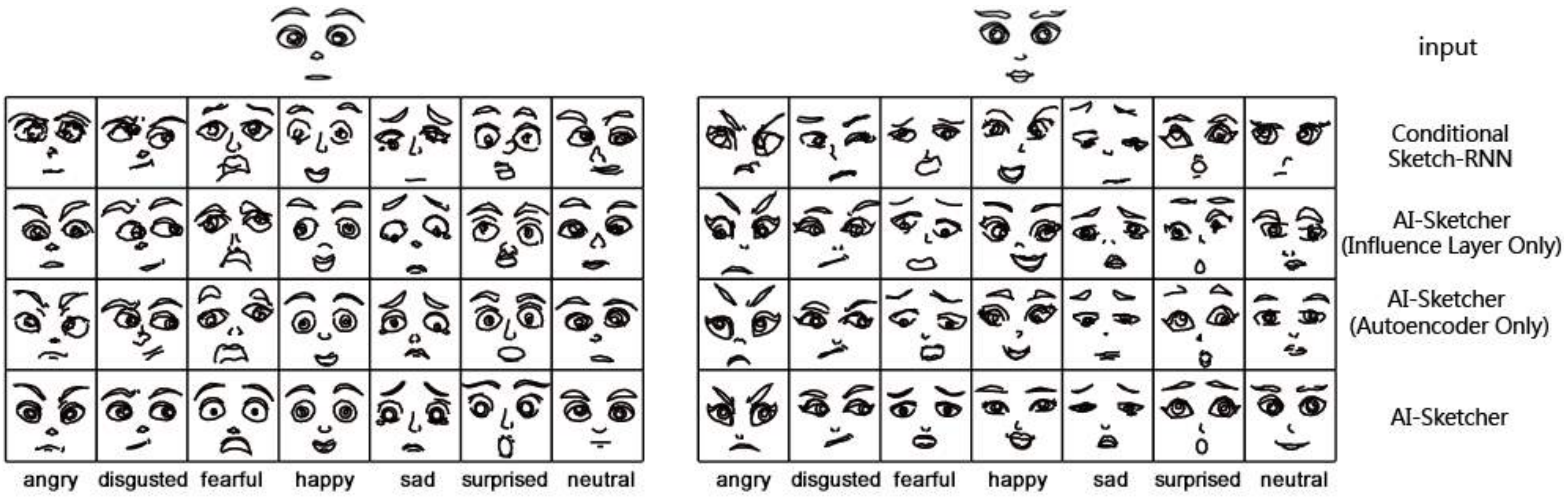
The FaceX dataset consists of 5 million sketches of both male's and female's facial expressions showing seven different types of emotions.



https://facex.idvxlab.com/

EVALUATION - 1st Experiment

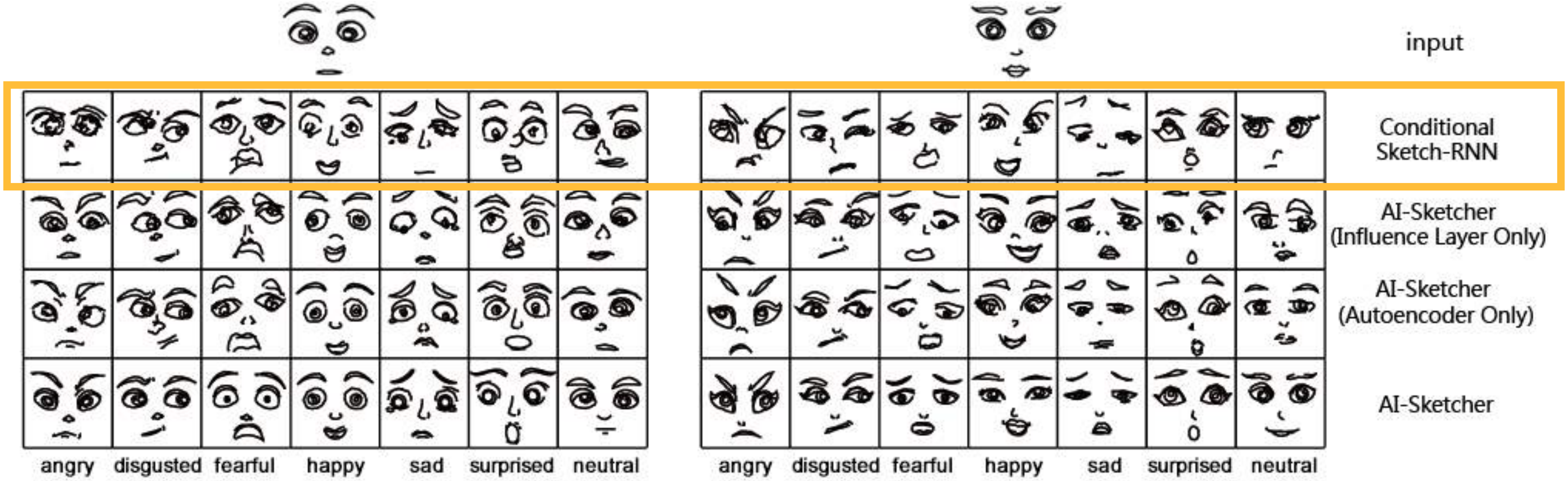
Drawing quality



Experiments based on FaceX Dataset

EVALUATION - 1st Experiment

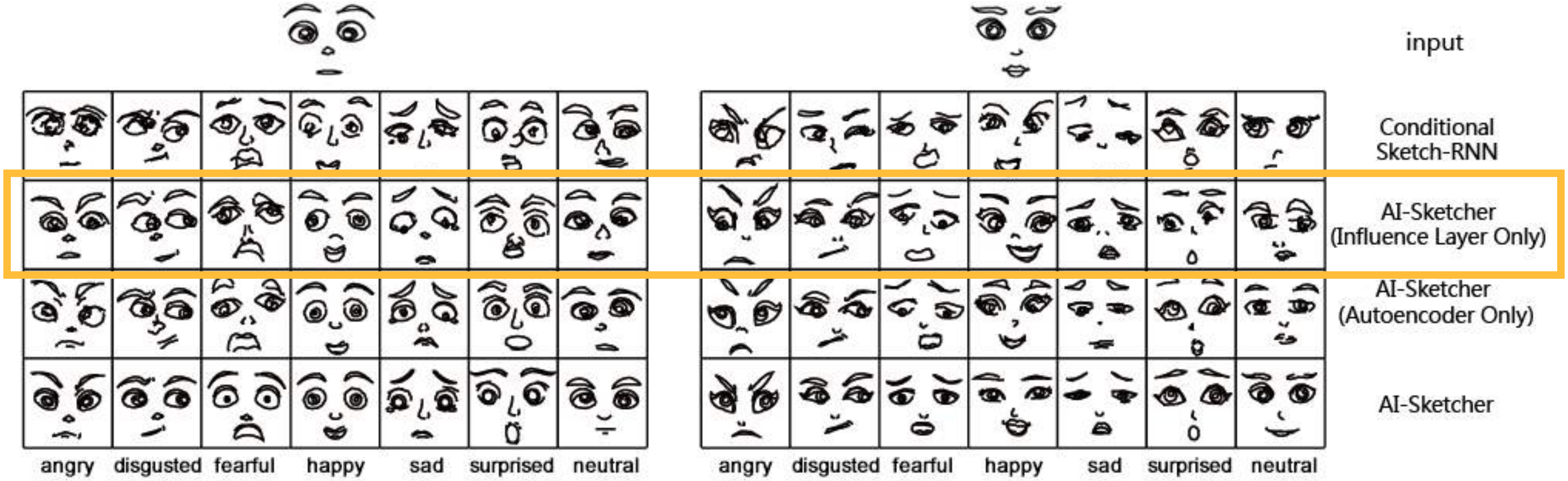
Drawing quality



Experiments based on FaceX Dataset

EVALUATION - 1st Experiment

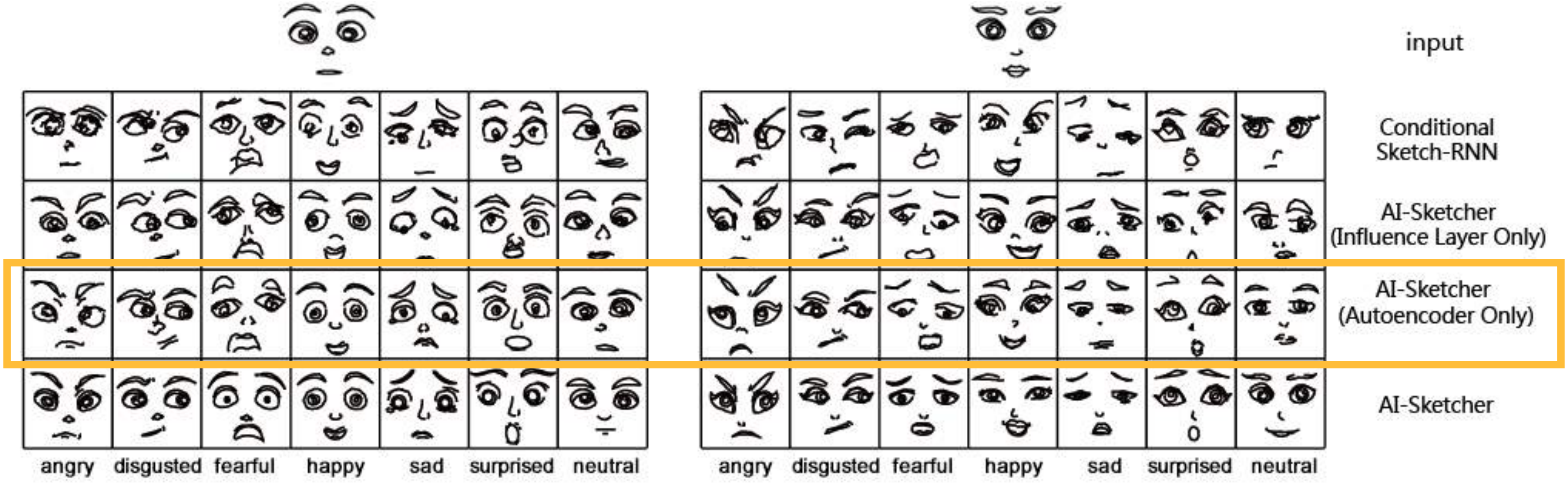
Drawing quality



Experiments based on FaceX Dataset

EVALUATION - 1st Experiment

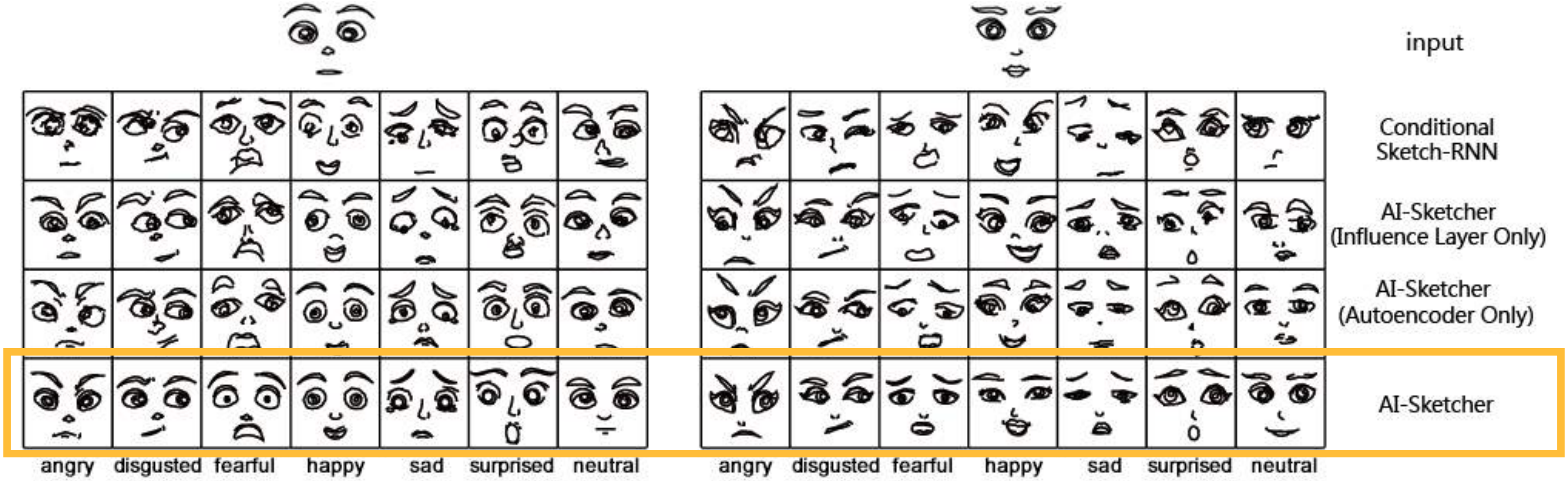
Drawing quality



Experiments based on FaceX Dataset

EVALUATION - 1st Experiment

Drawing quality



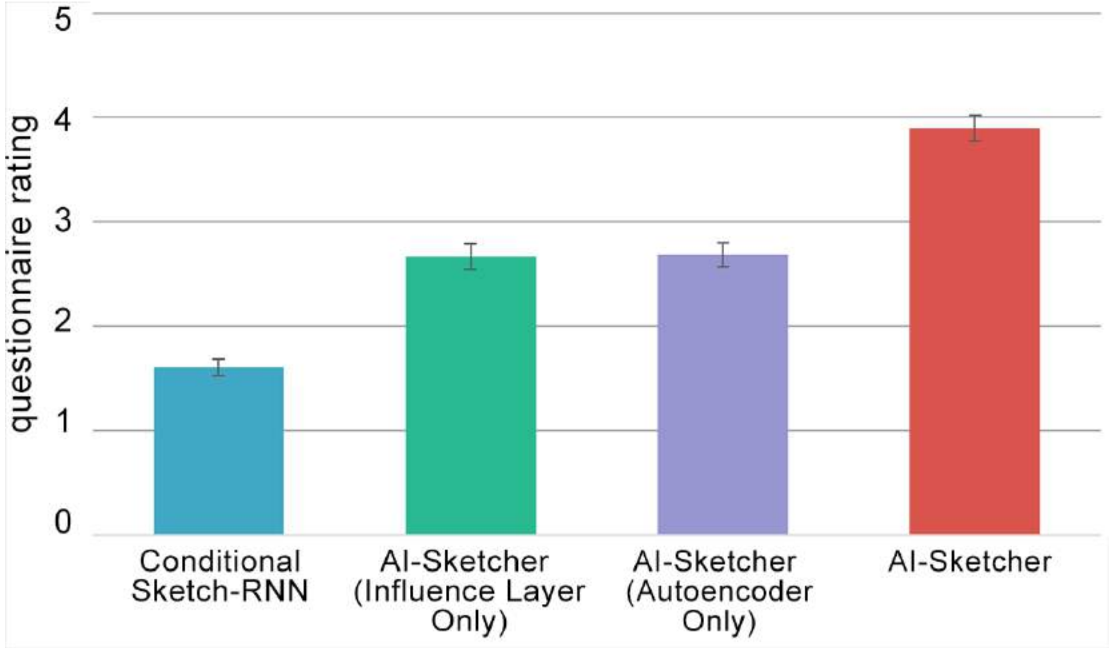
Experiments based on FaceX Dataset

EVALUATION - 1st Experiment

Drawing quality

A within-subject user study

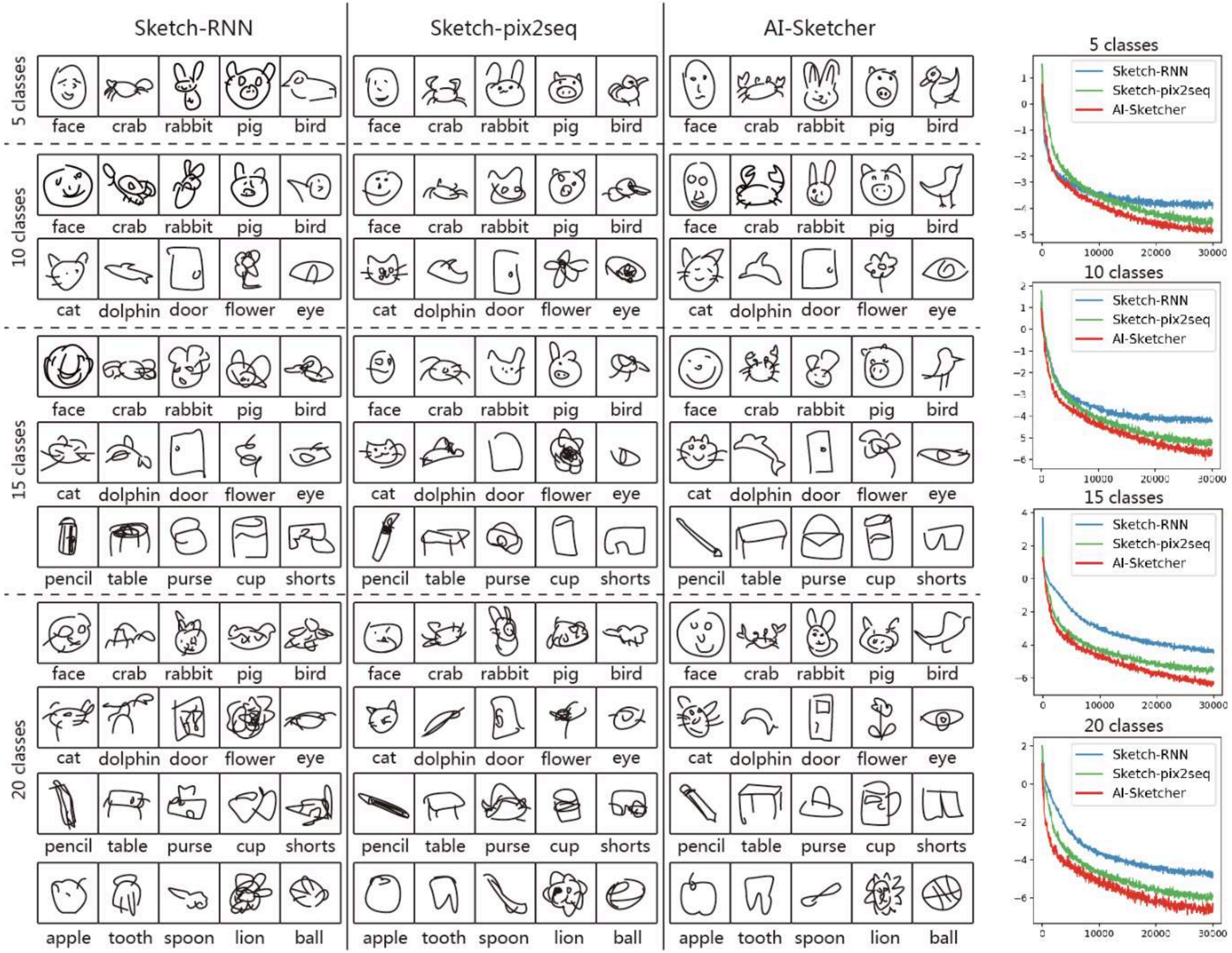
- 20 participants (10 females)
- The repeated measures one way ANOVA analysis showed that the generation quality of AI-Sketcher had an average rating of **3.9** and was significantly better than that of the baseline models (with all p<.01).



EVALUATION - 2nd Experiment

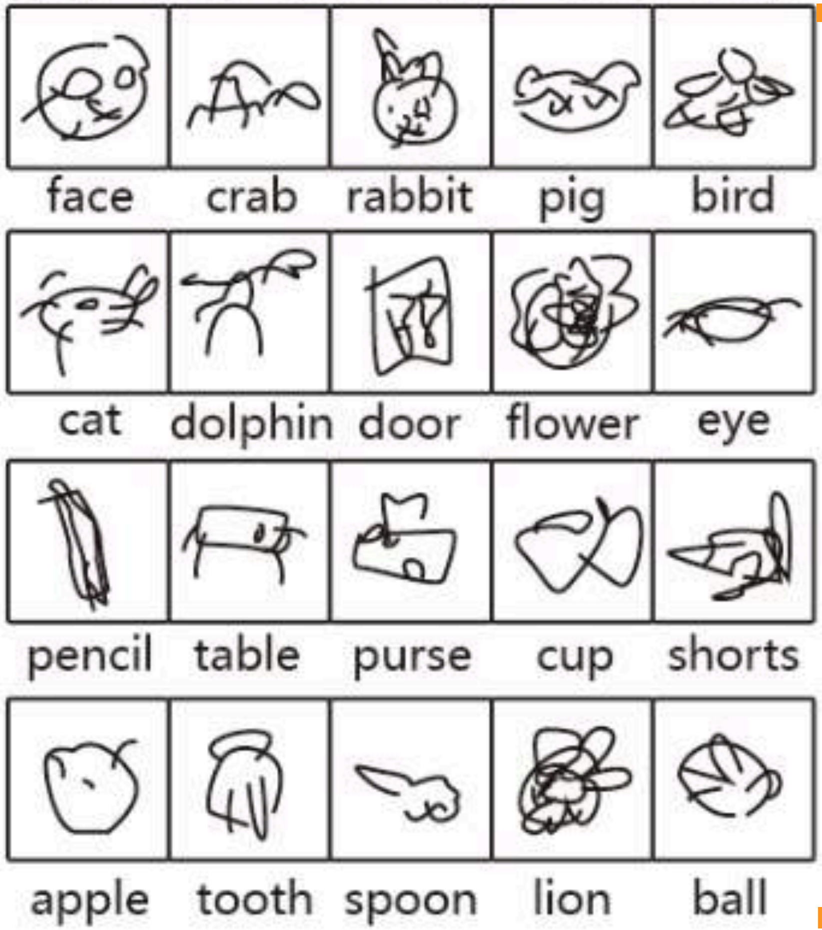
Generating sketches from
multiple classes

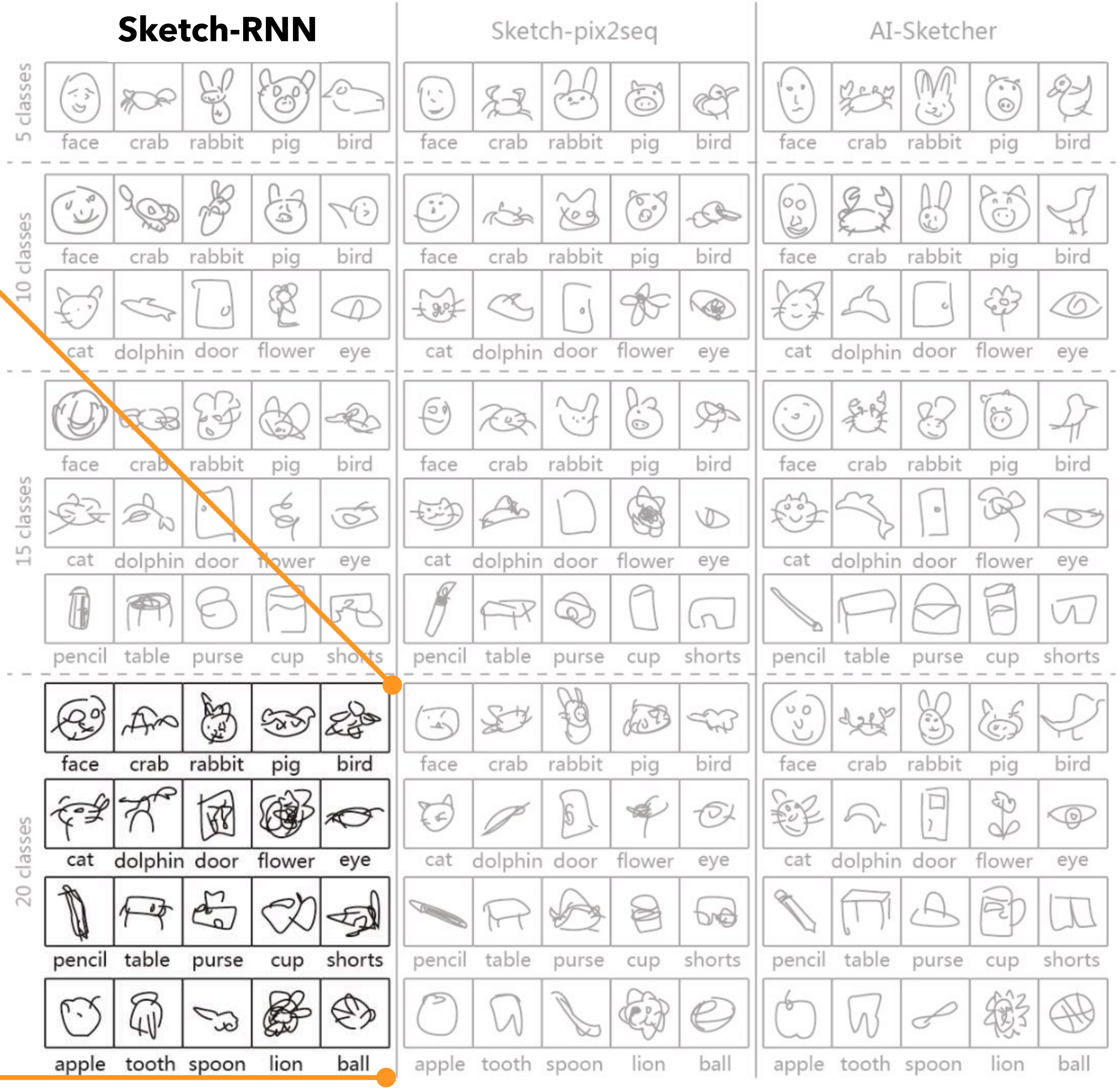
Experiments
based on QuickDraw Dataset

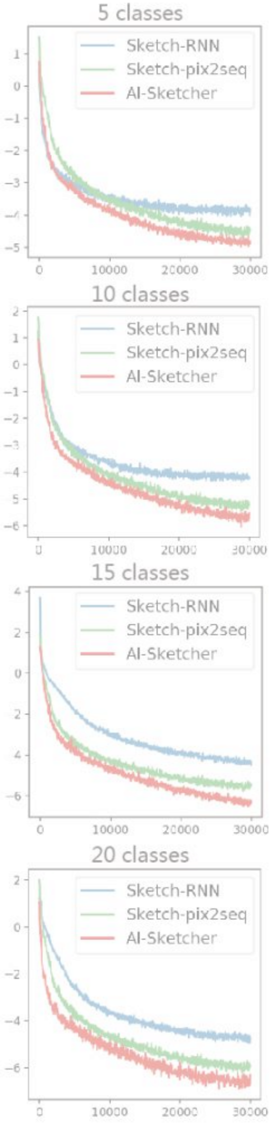


EVALUATION - 2nd Experiment

Generating sketches from
multiple classes

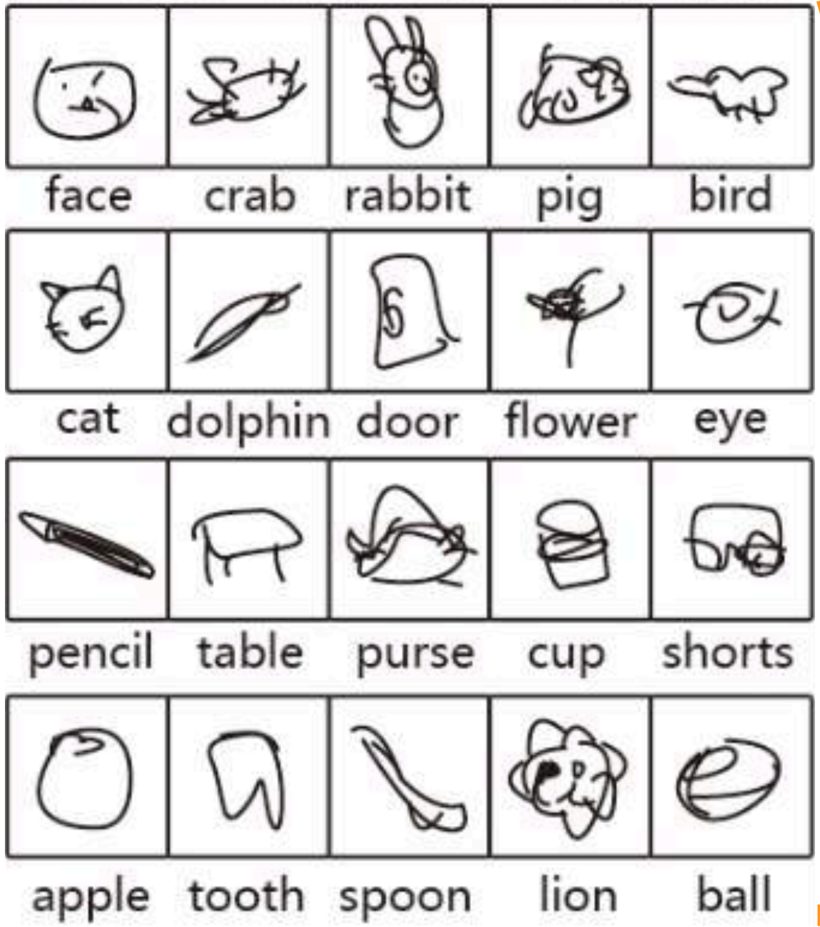


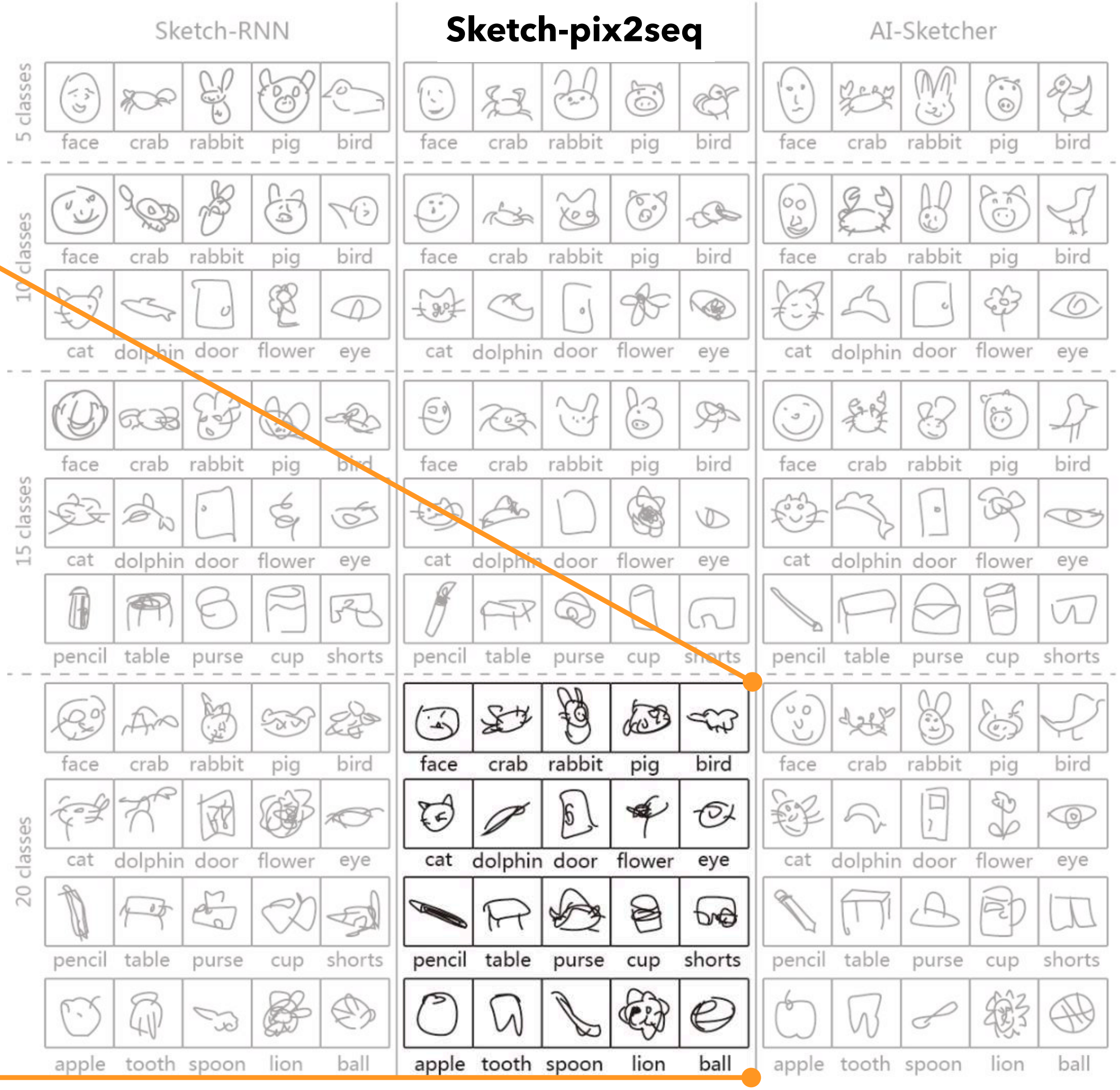


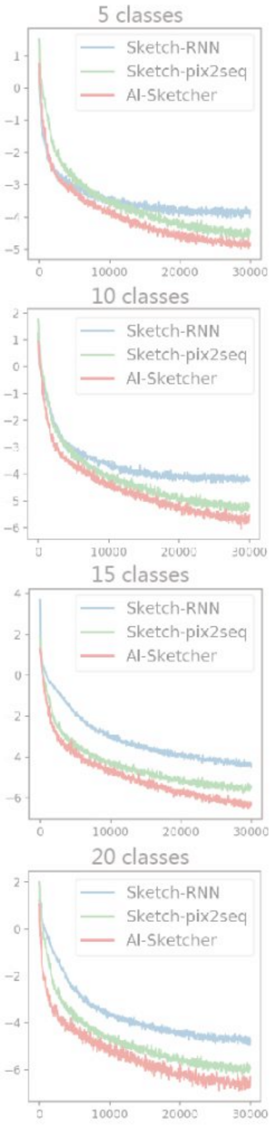


EVALUATION - 2nd Experiment

Generating sketches from
multiple classes

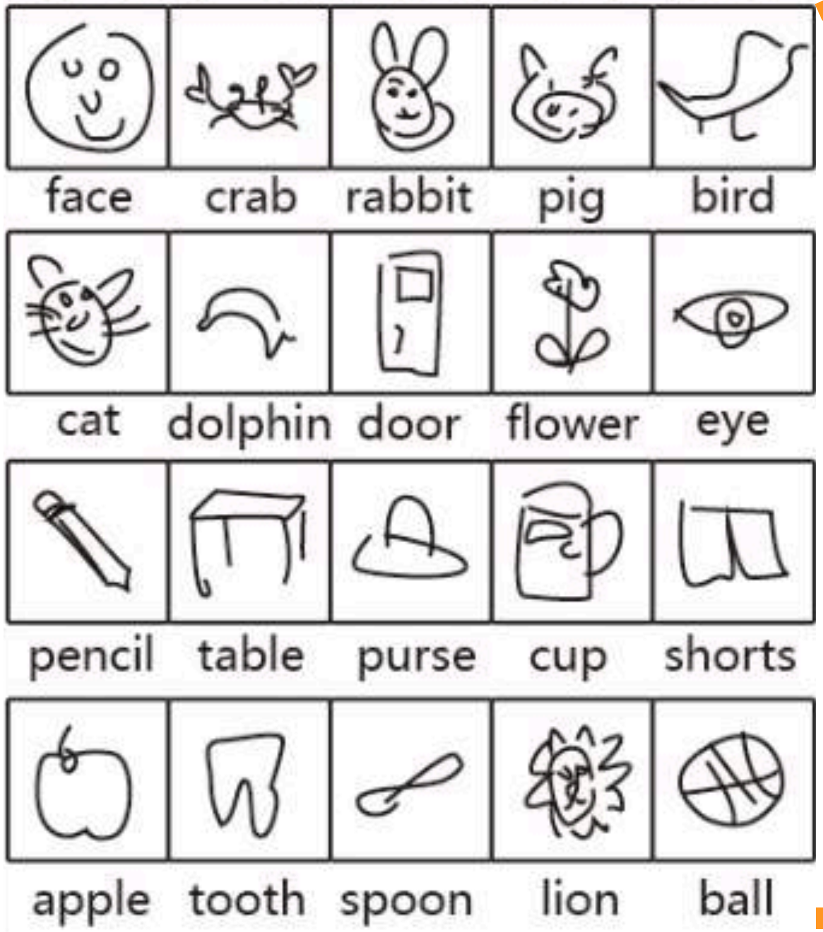


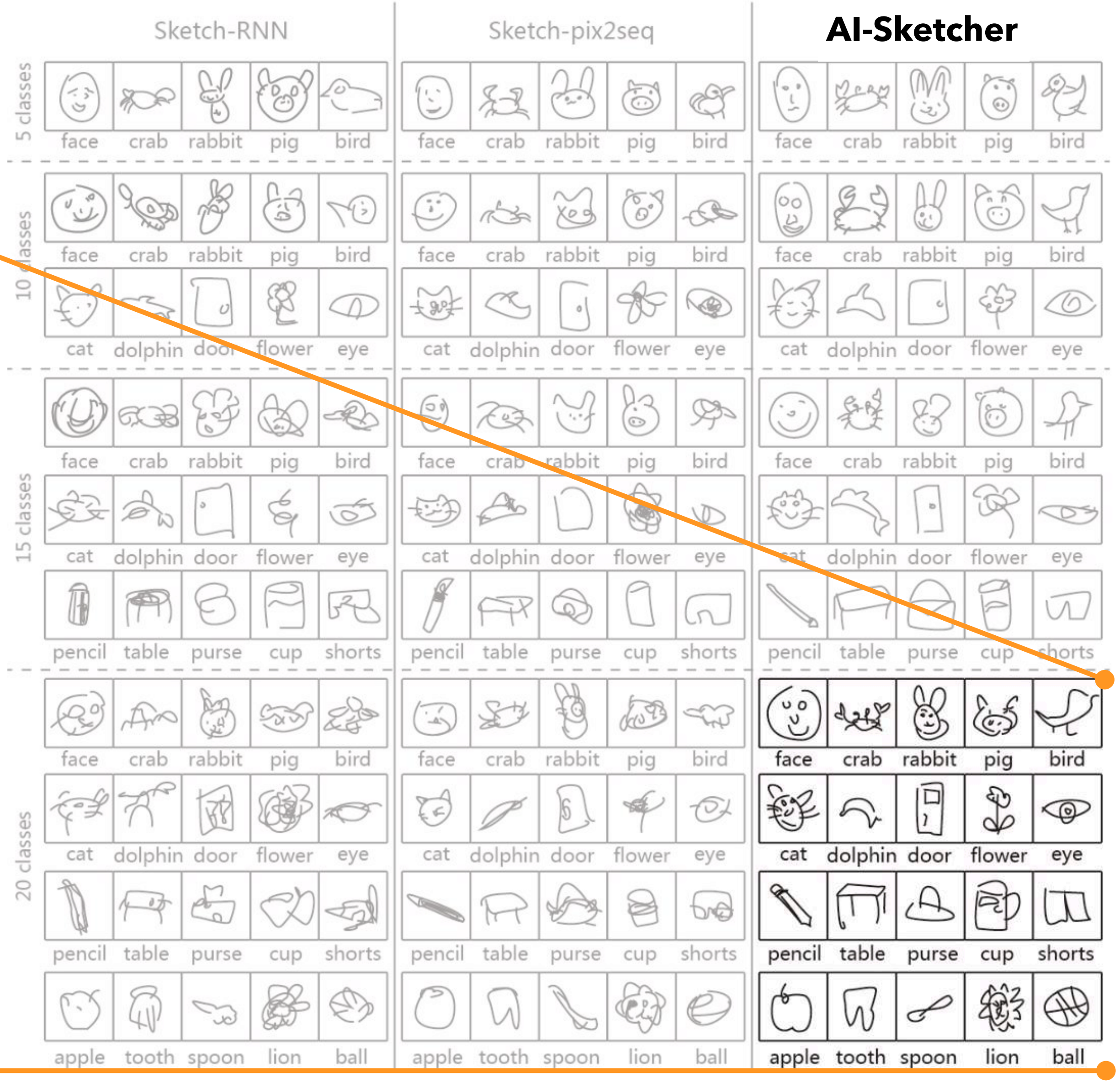


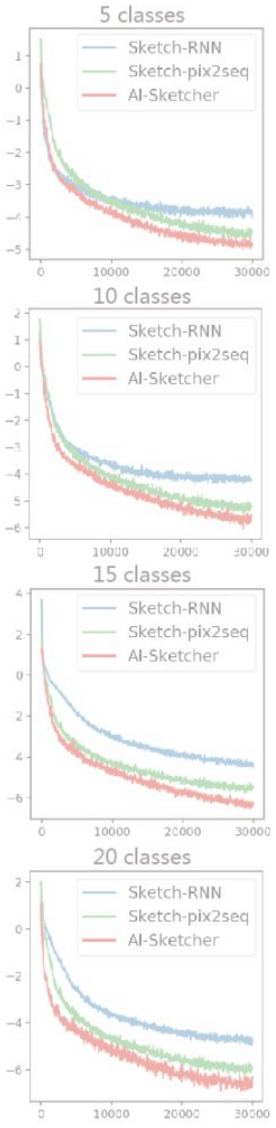


EVALUATION - 2nd Experiment

Generating sketches from multiple classes



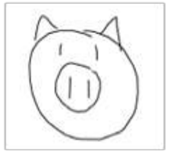
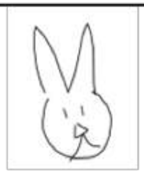




EVALUATION - 3rd Experiment

Generation Diversity

The unpaired t-test showed that AI-Sketcher and Sketch-RNN had no significant difference.

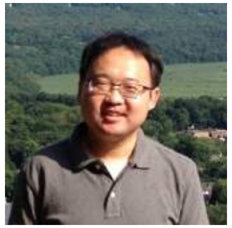
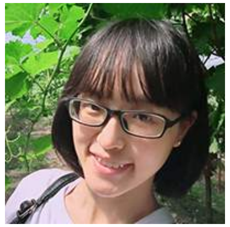
Input										
Model	●	●	●	●	●	●	●	●	●	●
Mean	30.66	30.97	29.76	29.87	28.98	28.82	29.23	29.45	29.66	30.00
SD	5.18	5.54	5.79	5.84	5.93	6.20	6.08	5.87	5.37	5.83
t(198)	-1.42		-0.53		0.65		-0.92		-1.53	
p	0.16 >.05		0.56 > .05		0.51 > .05		0.35 > .05		0.13 > .05	

● AI-Sketcher ● Sketch-RNN

Experiments based on QuickDraw Dataset

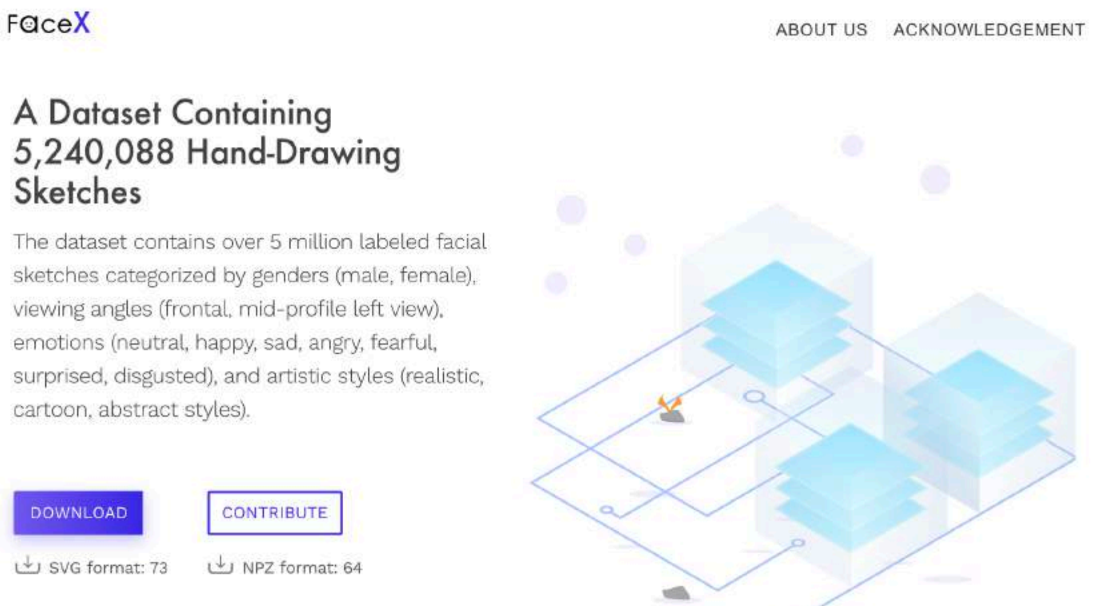
Thank You

Nan Cao, **Xin Yan**, Yang Shi, Chaoran Chen

Intelligent Big Data Visualization Lab

Tongji University



https://facex.idvxlab.com/