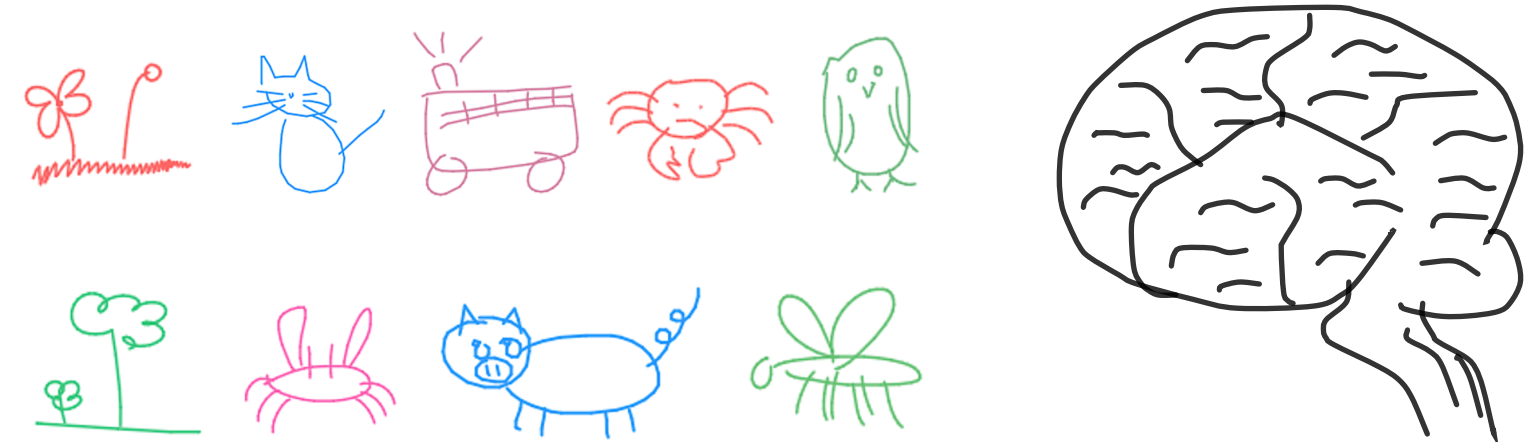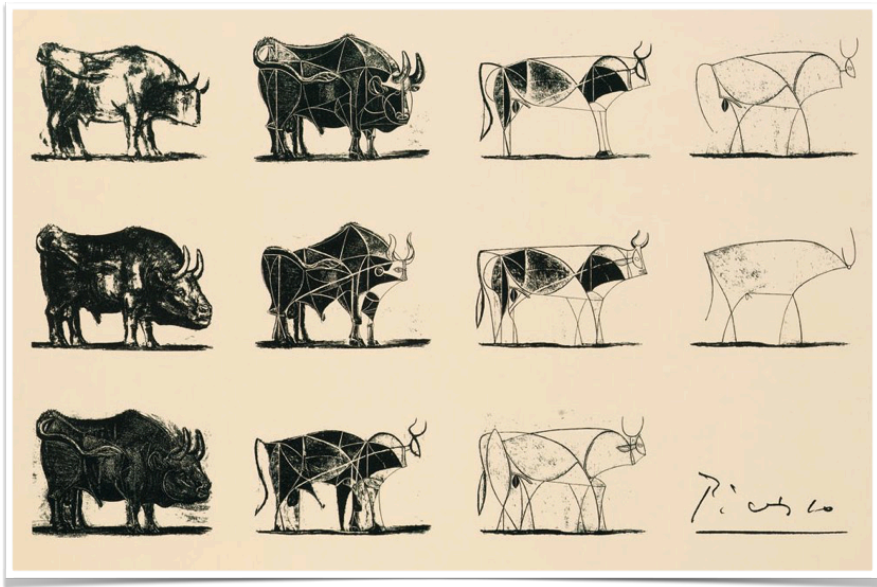# A Neural Representation of Sketch Drawings

David Ha (hadavid@google.com), Douglas Eck (deck@google.com)

## Motivation

"I never made a painting as a work of art. It's all research." – Pablo Picasso

Humans, do not understand the world as a grid of pixels, but develop abstract concepts to represent what we see.

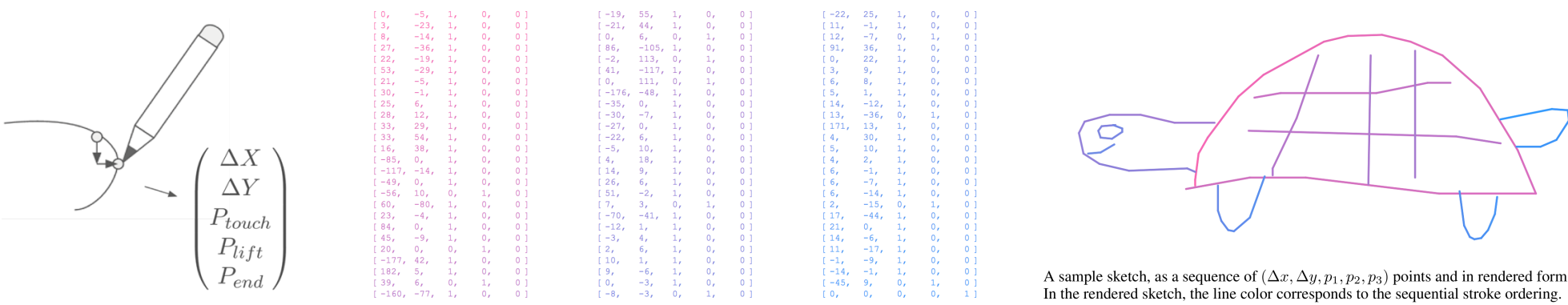We learn to express a sequential, vector representation of an image as a short sequence of strokes.

In this work, we investigate an alternative to pixel image modelling approaches, and propose a generative model for vector images.
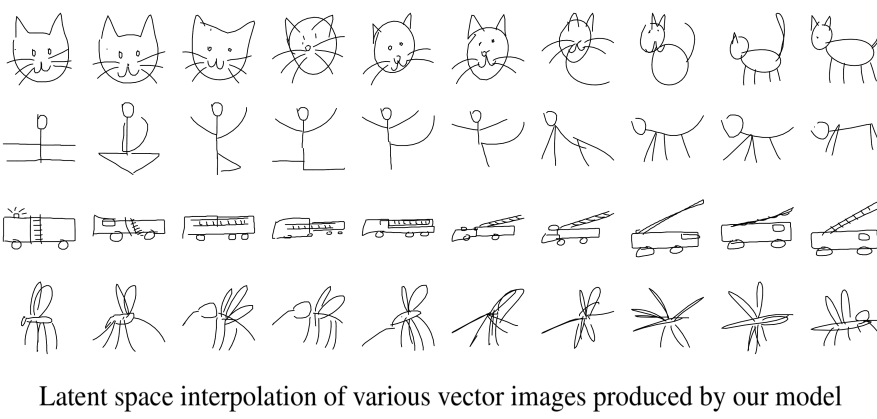
Latent space interpolation of various vector images produced by our model

## Quick, Draw! Dataset

quickdraw.withgoogle.com/data

A sample sketch, as a sequence of $(\Delta x, \Delta y, p_1, p_2, p_3)$ points and in rendered form.
In the rendered sketch, the line color corresponds to the sequential stroke ordering.
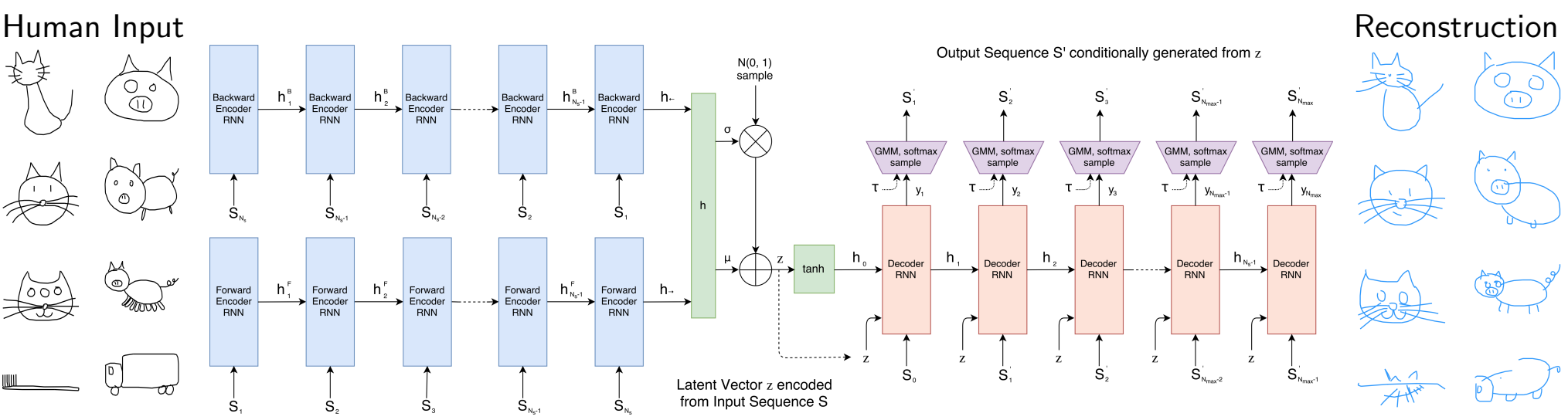
Sketches are represented as a sequence of motor actions controlling a pen.
Open sourced dataset of 50M doodles, collected from *Quick, Draw!* game.

Example sketch drawings from QuickDraw dataset.
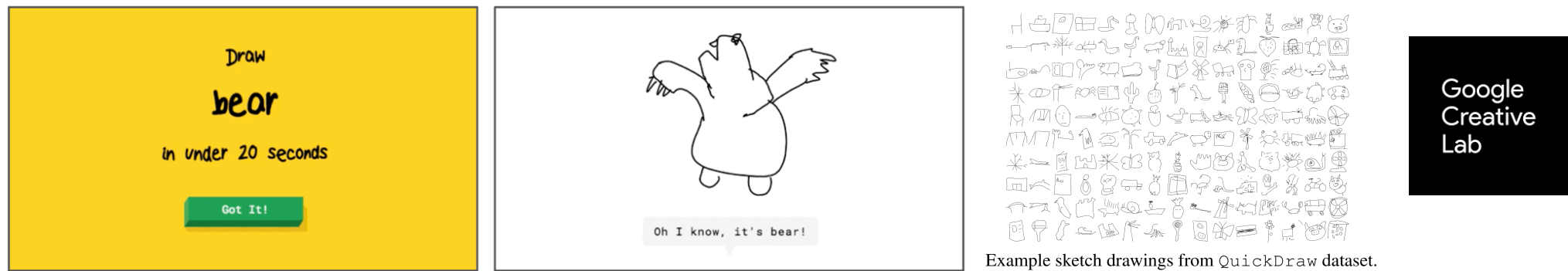
## sketch-rnn

magenta.tensorflow.org/sketch-rnn-demo
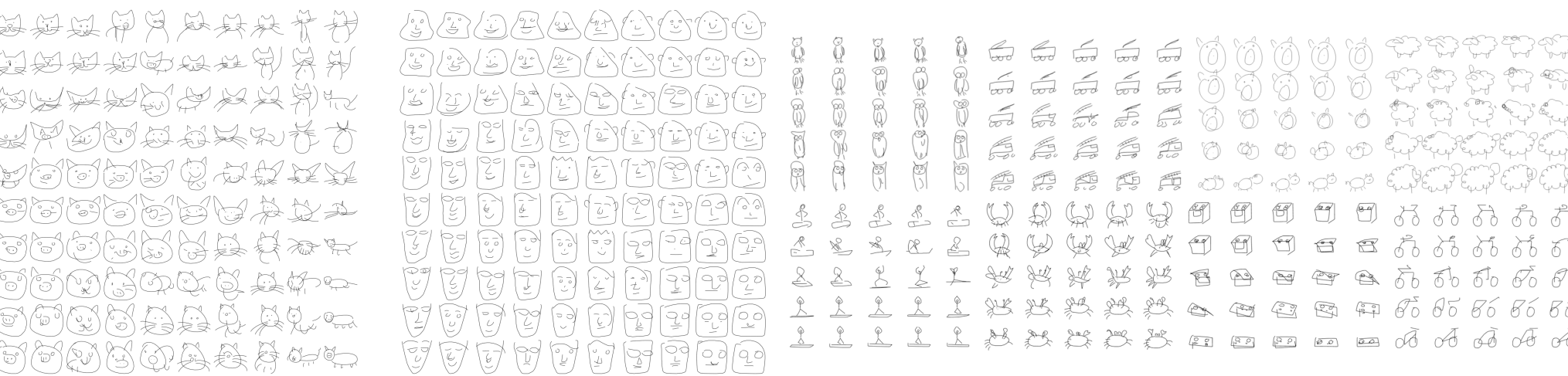
Human Input / Reconstruction

- Seq-to-Seq Variational Autoencoder
- Bidirectional RNN Encoder
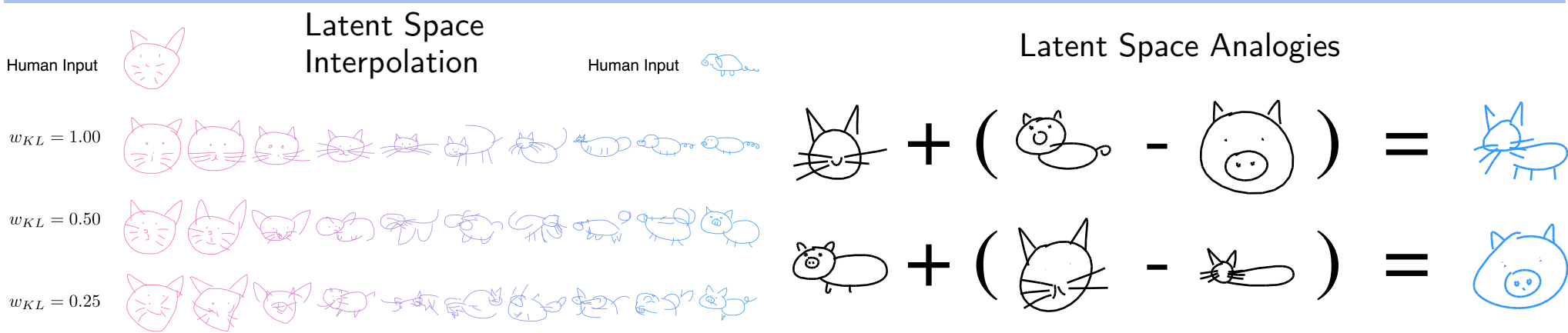- Mixture Density RNN Decoder
- Weighted Loss Function

Reconstruction loss term $L_R$
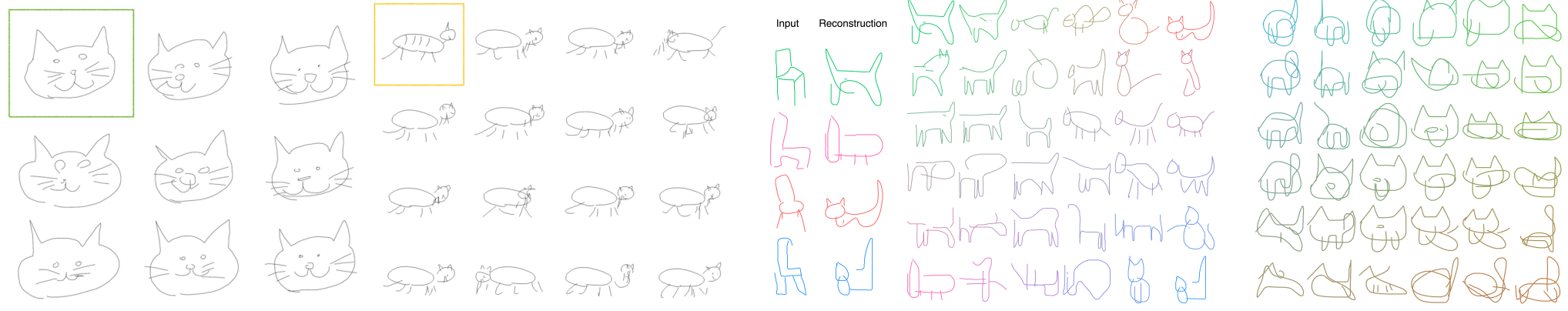
KL loss term $L_{KL}$

$$Loss = L_R + w_{KL} L_{KL}$$

## Conditional Generation: Latent Space Exploration
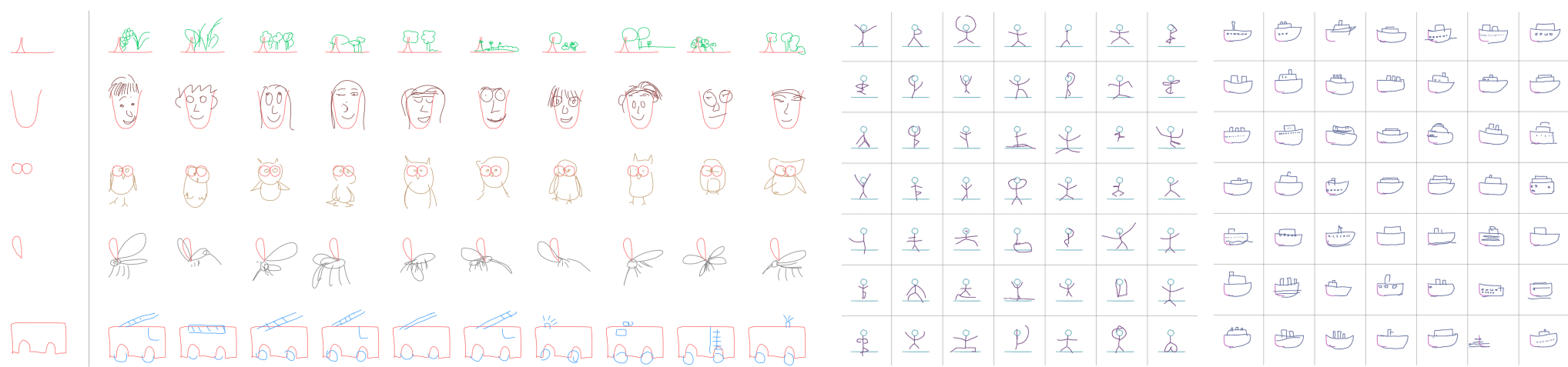
Latent Space Interpolation

Latent Space Analogies

- Sample multiple similar drawings from a single human input sketch.
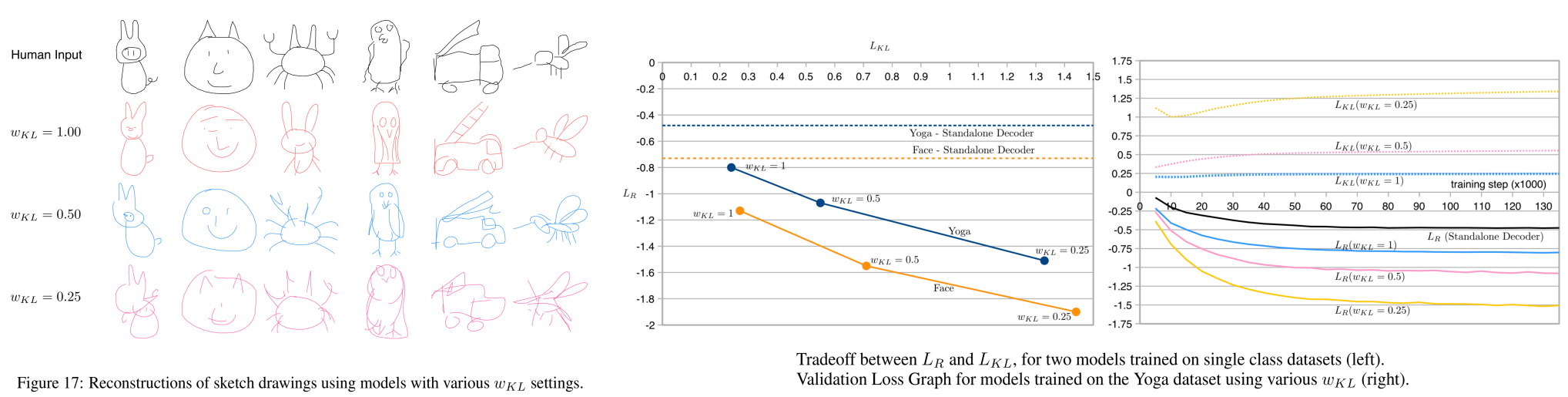- Generate a drawing in the style of another class, and also interpolate for more.

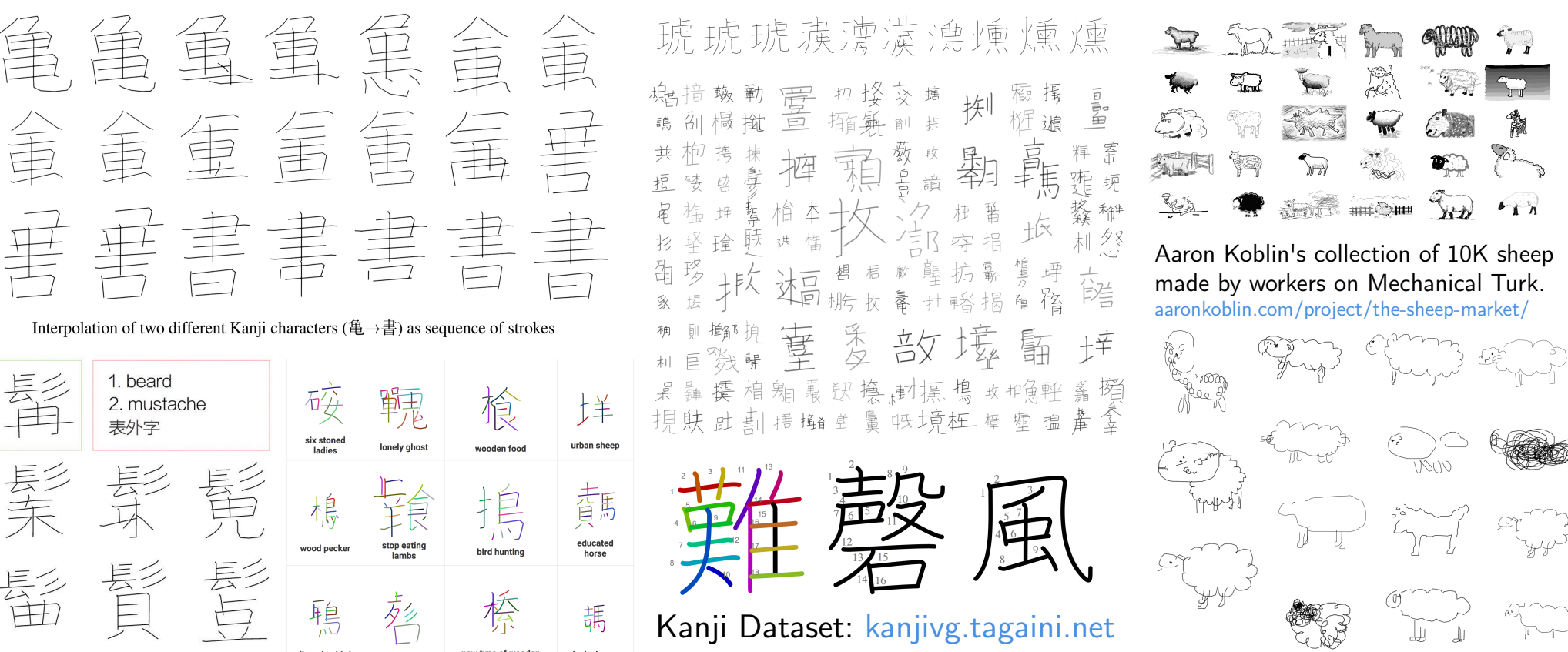## Unconditional Generation: Use Decoder Only

- Varying temperature parameter to control variability of generated sketches.
- Predicting possible endings of various incomplete sketches.
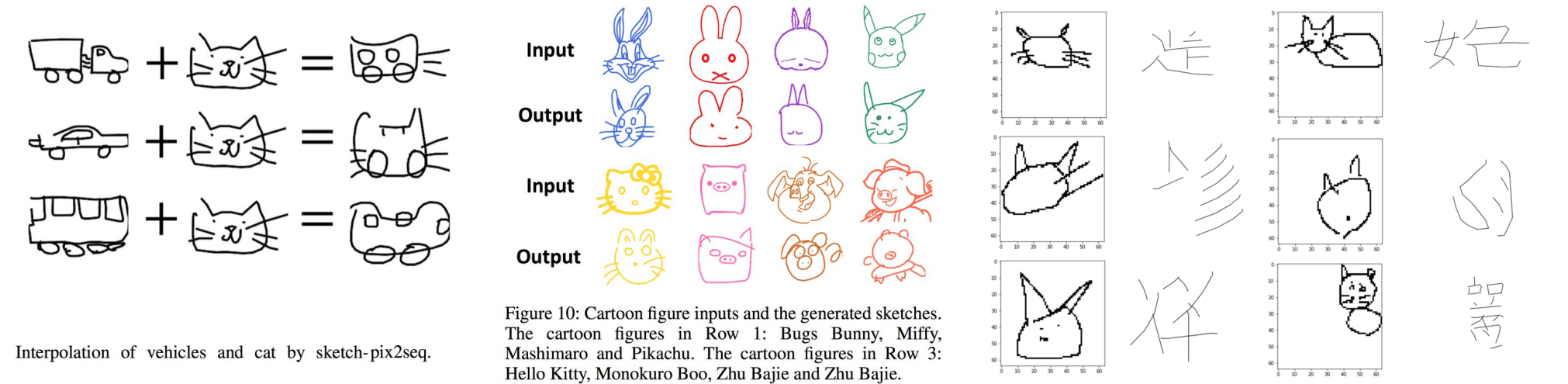
## Which Loss Controls Image Coherency?

Tradeoff between $L_R$ and $L_{KL}$ for two models trained on single class datasets (left).
Validation Loss Graph for models trained on the Yoga dataset using various $w_{KL}$ (right).

Figure 17: Reconstructions of sketch drawings using models with various $w_{KL}$ settings.

## Other Datasets: Kanji, Aaron's 10K Sheep Market

Interpolation of two different Kanji characters (亀→書) as sequence of strokes.

Aaron Koblin's collection of 10K sheep made by workers on Mechanical Turk.
aaronkoblin.com/project/the-sheep-market/
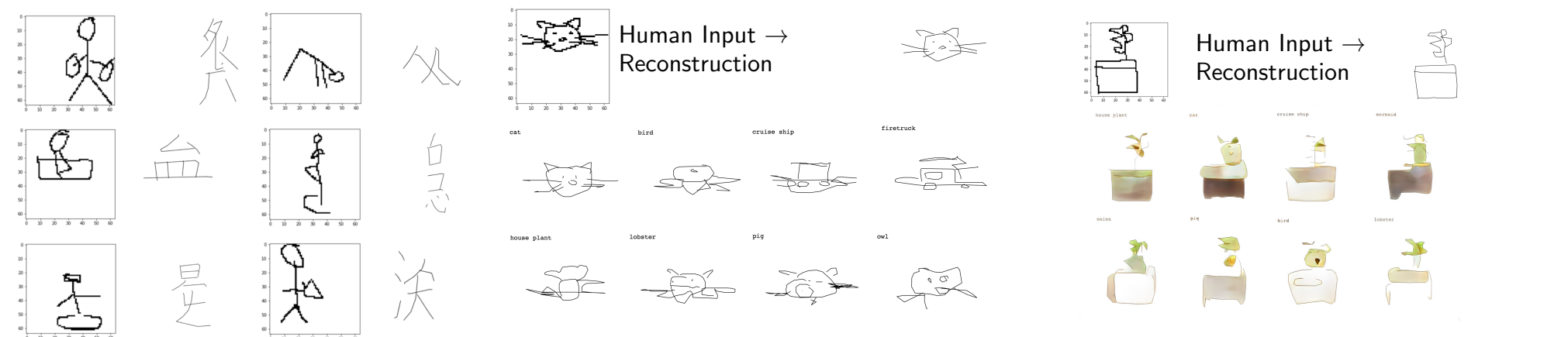
難 磬 風

Kanji Dataset: kanjivg.tagaini.net

## Pixel → Strokes

- We can replace the encoder from a RNN to a ConvNet to read in pixels.
- Chen et al. (2017) extended sketch-rnn to use ConvNet encoder for pix2strokes:

Figure 10: Cartoon figure inputs and the generated sketches.
The cartoon figures in Row 1: Bugs Bunny, Miffy, Mashimaro and Pikachu. The cartoon figures in Row 3: Hello Kitty, Monokuro Boo, Zhu Bajie and Zhu Bajie.

Interpolation of vehicles and cat by sketch-pix2seq.

- We can try this technique across domains, such as QuickDraw → Kanji.

Human Input → Reconstruction

- Or convert a pixel image of one QuickDraw class into strokes of another class.

## sketch-rnn with colorization

- Generated sketch drawings can be colorized using image-to-image translation.

## References

Alex Graves, Generating sequences with RNNs, 2013.

Bowman et al., Generating Sentences from a Continuous Space, 2015.

Zhang et al., Drawing and Recognizing Chinese Characters with RNN, 2016.

Chen et al., Sketch-pix2seq: A Model to Generate Sketches of Multiple Categories, 2017.

## Online Resources

| | |
|---|---|
| Fake Kanji Experiment | otoro.net/kanji |
| Quick, Draw! Game | quickdraw.withgoogle.com |
| Paints Chainer | github.com/pfnet/PaintsChainer |
| Paints Transfer | github.com/lllyasviel/style2paints |
| sketch-rnn github repo | magenta.tensorflow.org/sketch_rnn |
| other datasets | github.com/hardmaru/sketch-rnn-datasets |