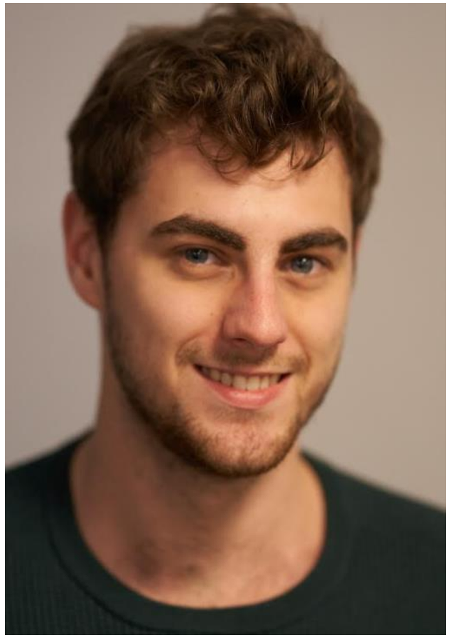
Workshop on Deep generative models for image and text generation

PyData Eindhoven 2019

Who are we?

Dimitra Gkorou

Sonali Fotendar



Koen Vannisselroij

Why image and text generation in one workshop?

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo

- Continuous points
- Spatial Dependencies
- Not sensitive to small local changes

- Discrete data
- Time sequences with potential long-term sequential dependencies
- Sensitive to small local changes
- Follows rule-based grammatical structure

Generative Model

Basics of Generative Modeling

- We have a dataset of observations X generated according to some unknown distribution
- A **generative model** tries to learn and mimic the data distribution
- We can sample from the distribution learned by the generative model to generate new data points
- We want these new data points to:
  - appear drawn from the original data distribution
  - be different from the original observations X

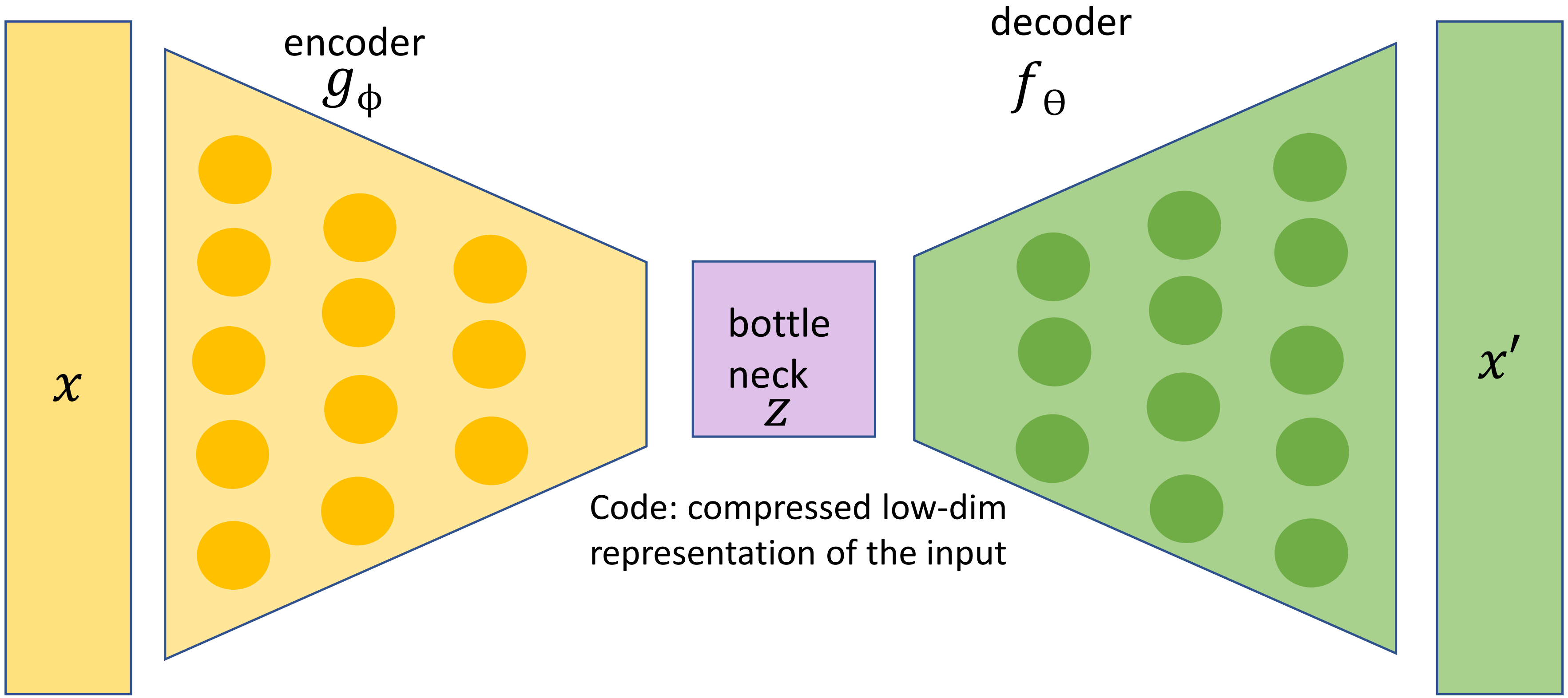
Example

Potential candidate for generative Modeling… ???

Auto-Encoders

Autoencoder architecture

A neural network where the input is the same as the output.



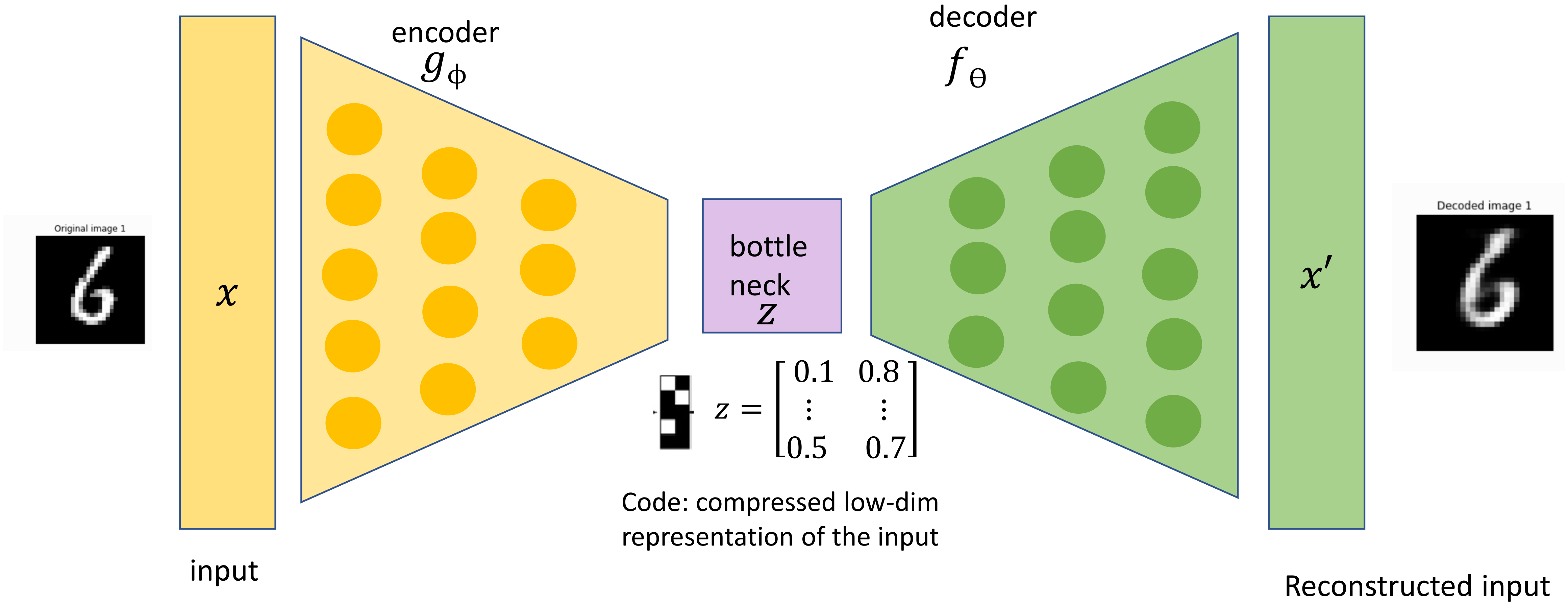
input

Encoder network:
the original high-dimension input → latent low-dimensional code.

Decoder network:
recovers the data in the original high dimensional space from the code

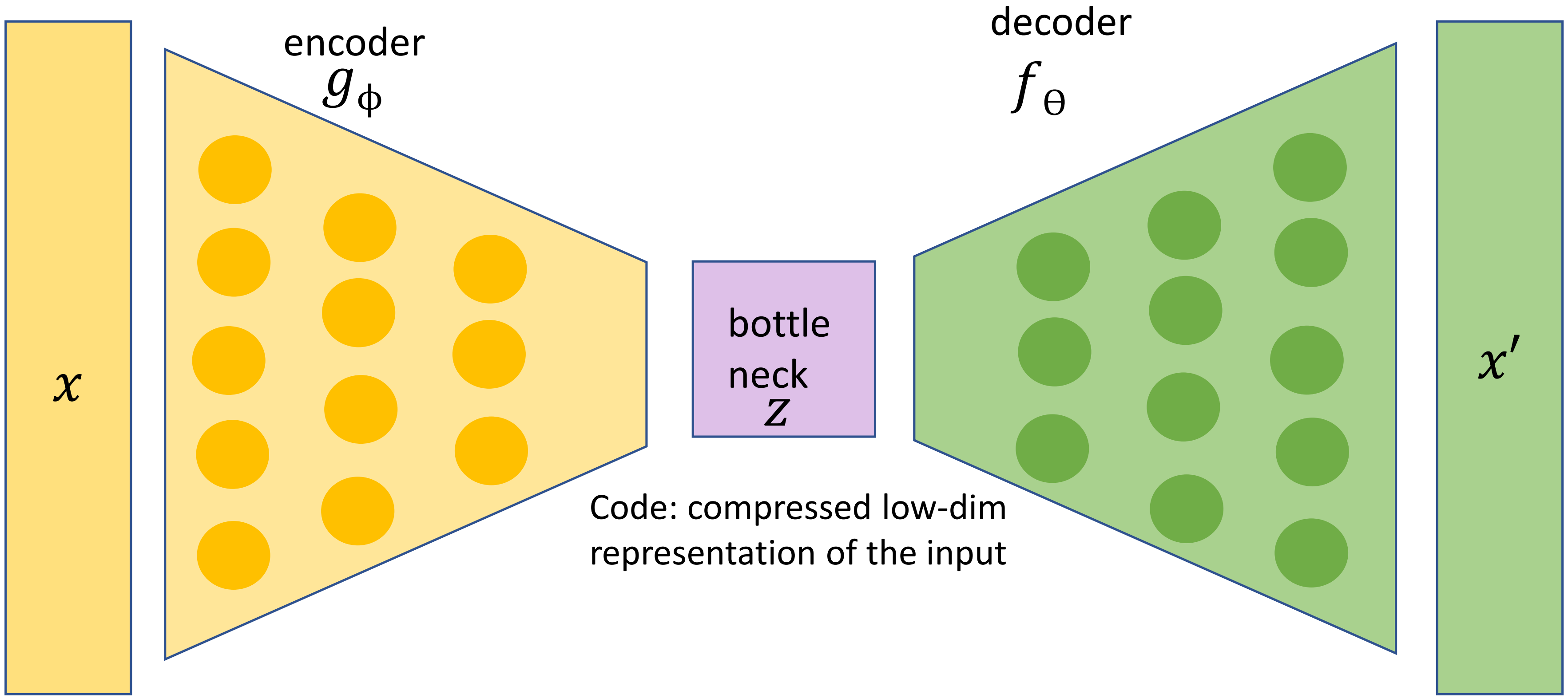
Reconstructed input

Autoencoder loss example



Applications: Dimensionality Reduction, Visualization, One class Classification, Anomaly detection

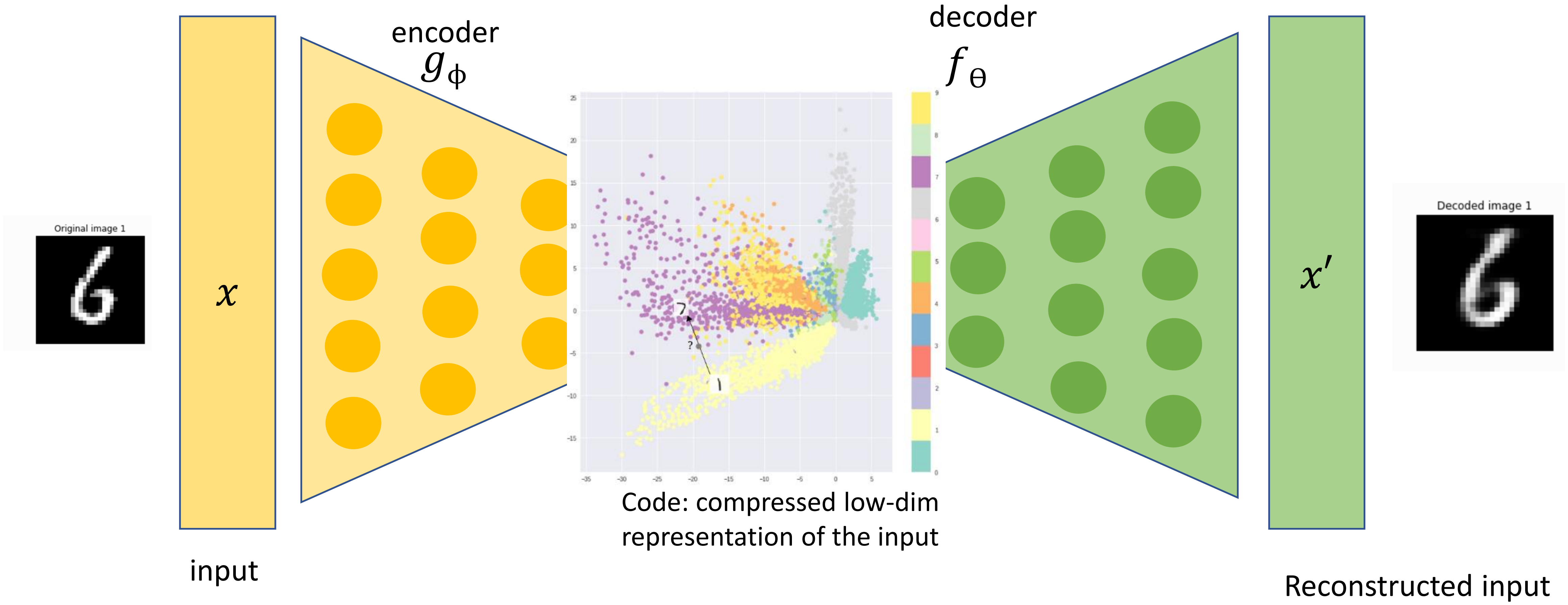
Autoencoder loss function



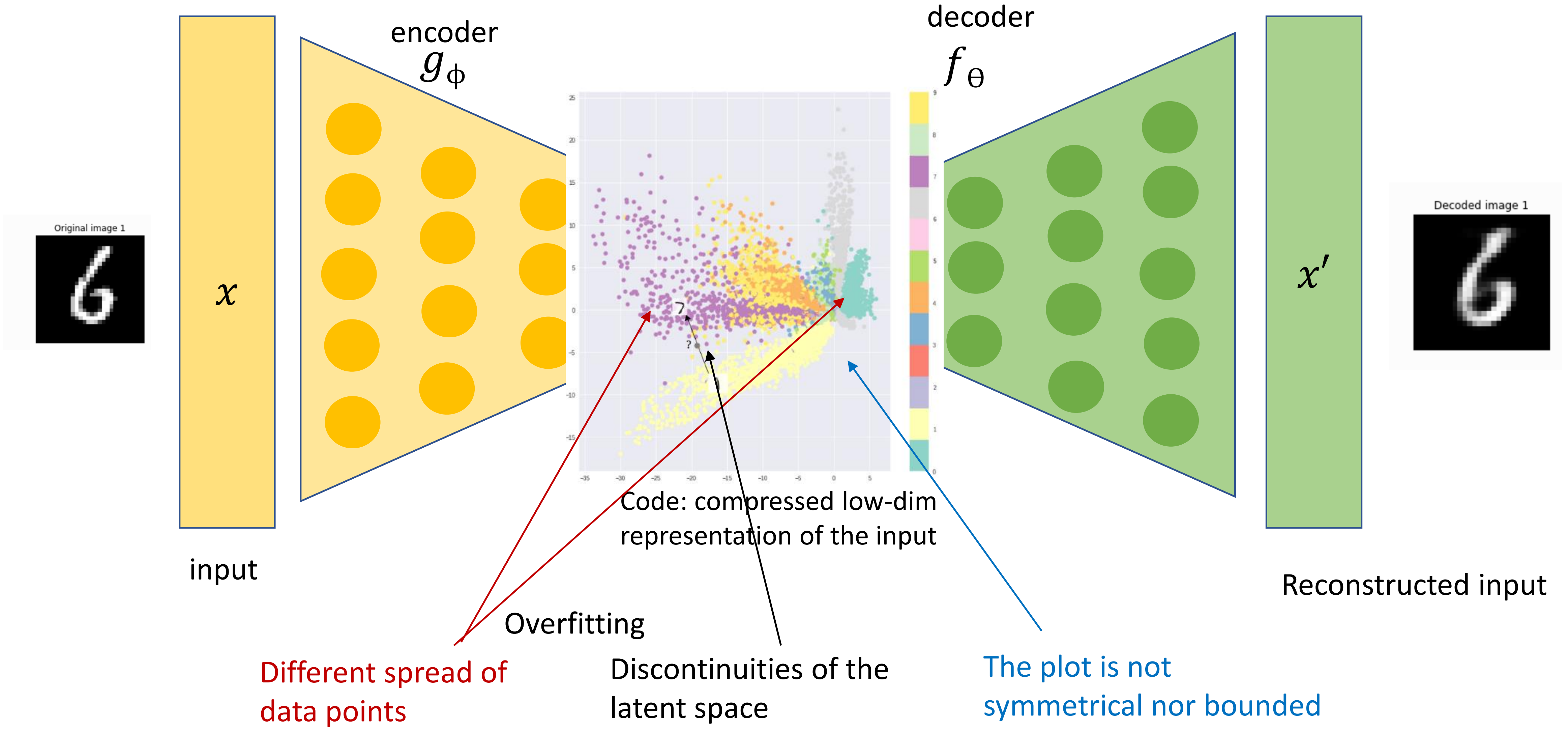
$$LAE(\Theta, \phi) = \mathcal{L}_{reconstruction} = ||x' - x||$$

$$LAE(\Theta, \phi) = 1/n \sum_{i=1}^{n} \left(x^{(i)} - x'^{(i)}\right)^2 = 1/n \sum_{i=1}^{n}\left(x^{(i)} - f_\Theta\left(g_\phi\left(x^{(i)}\right)\right)\right)^2$$

Why not AE as generative model?



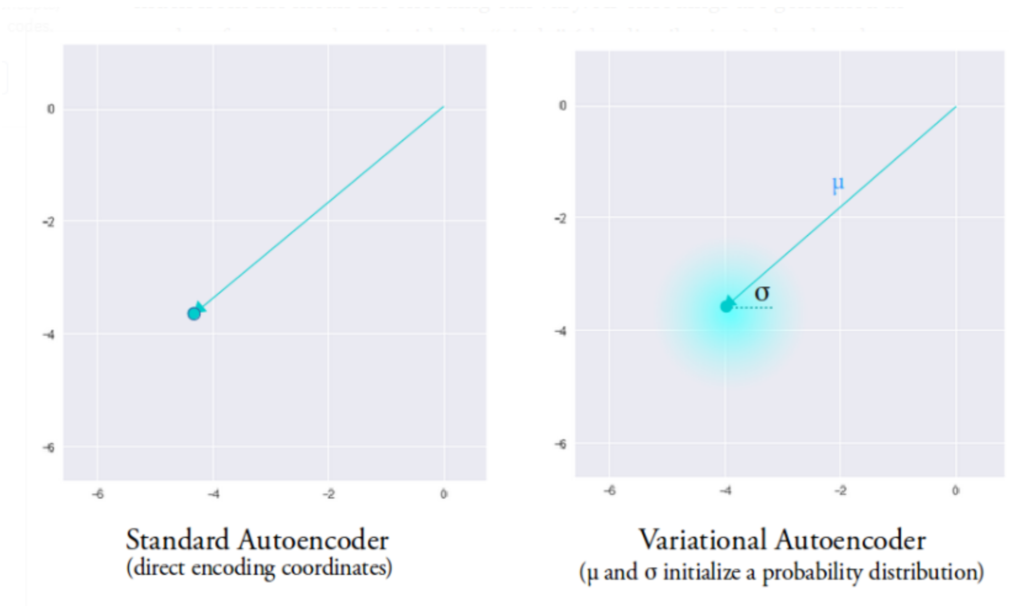
Why not AE as generative model?



Variational Auto-Encoders (VAE)

Variational Autoencoder Basics

Probabilistic view of autoencoders

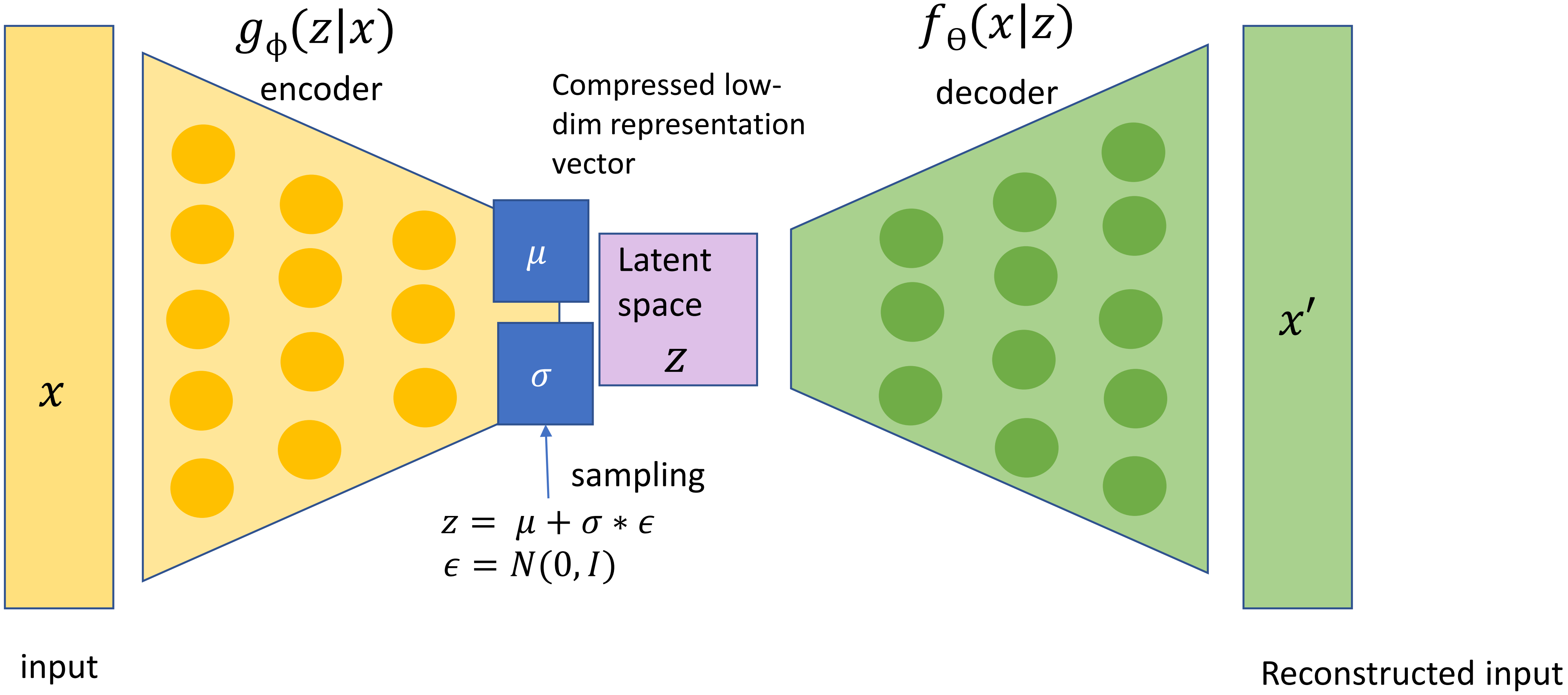


Each image mapped to a **multivariate normal distribution** around a point in the latent space

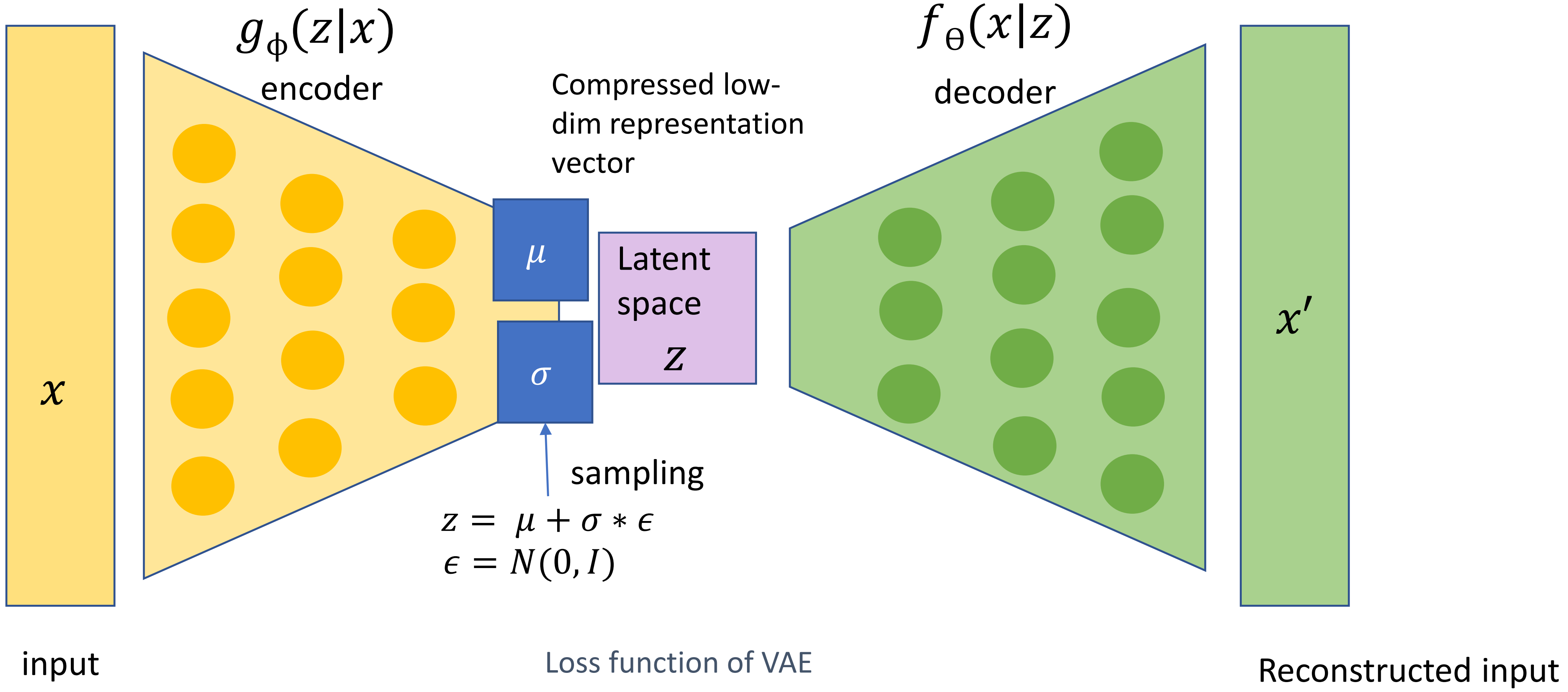
The encoder ensure that all **points in a neighborhood** produce **similar images** when decoded (with a small reconstruction error)

As a result, even points in the latent space that are not seen before, are likely to decode **well-formed images**

Variational Autoencoder architecture



Variational Autoencoder loss function

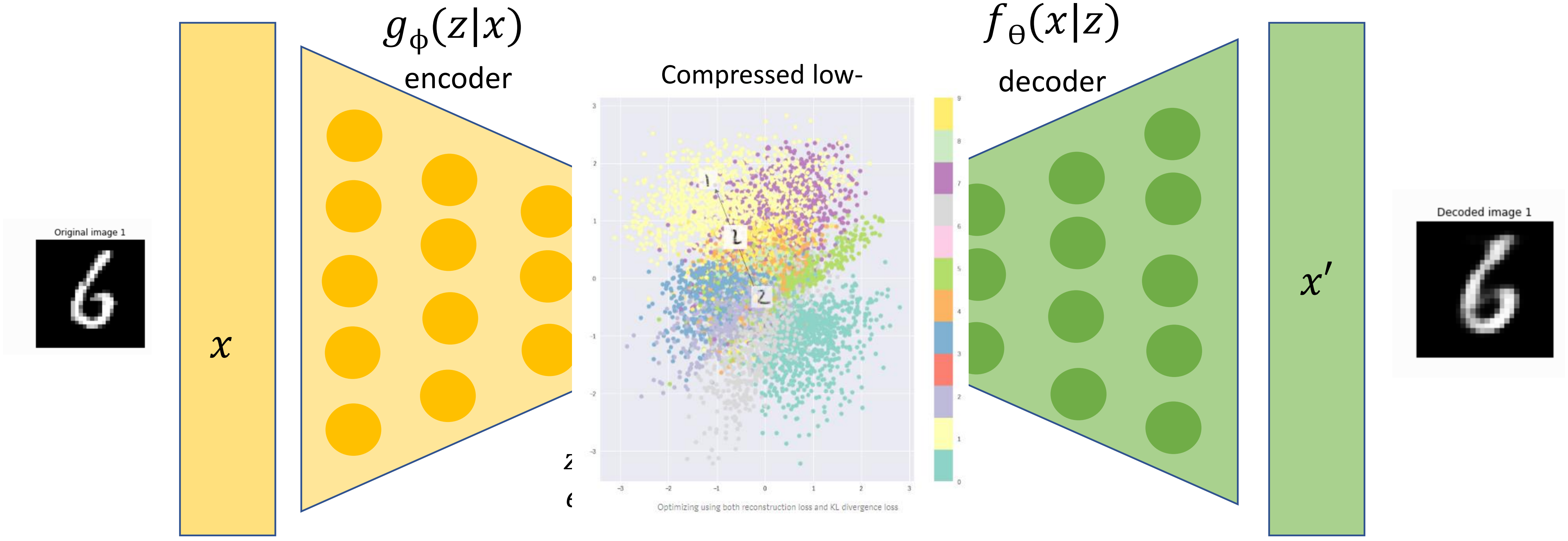


Loss function of VAE

$$LVAE = \mathcal{L}_{reconstruction} + \mathcal{L}_{KL}$$

$$LVAE = -E_{z\sim g_\phi}[\log\left(f_\Theta(x^{(i)}|z)\right)] + KL(g_\phi(z|x^{(i)})||p(z))$$

Variational Autoencoder loss function



input

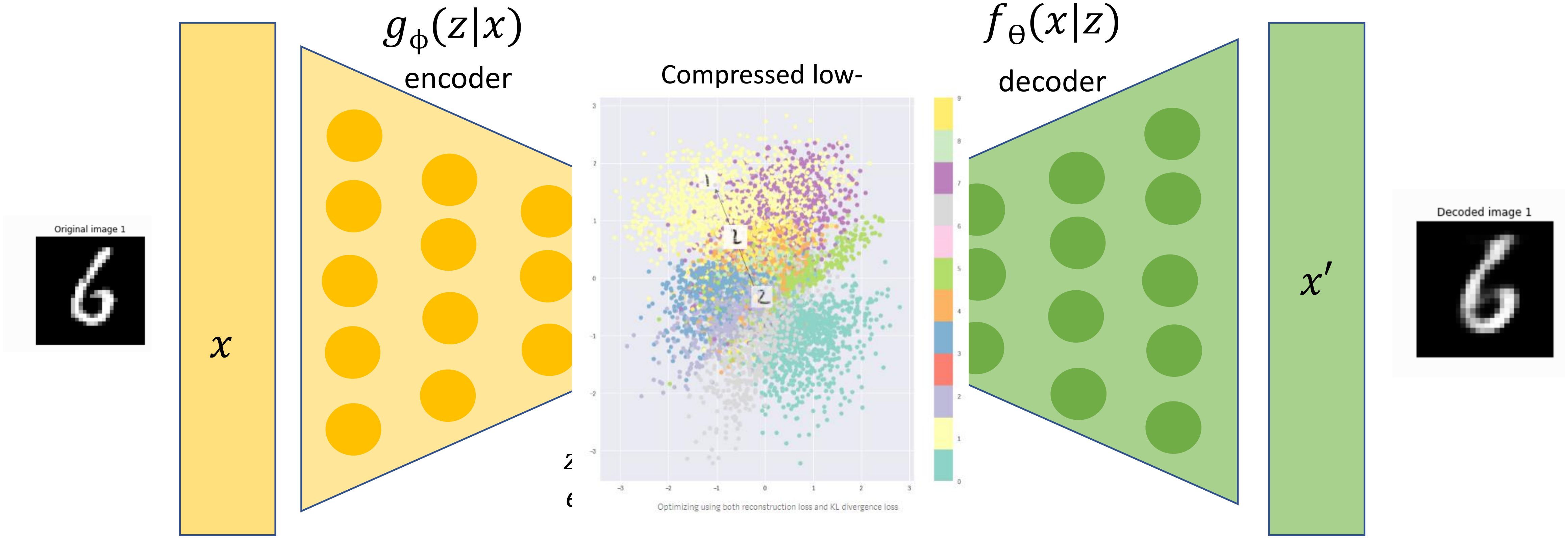
Loss function of VAE

Reconstructed input

$$LVAE = \mathcal{L}_{reconstruction} + \mathcal{L}_{KL}$$

$$LVAE = -E_{z \sim g_{\phi}}\left[\log\left(f_{\Theta}(x^{(i)}|z)\right)\right] + KL(g_{\phi}(z|x^{(i)})||p(z))$$

Variational Autoencoder loss function



input

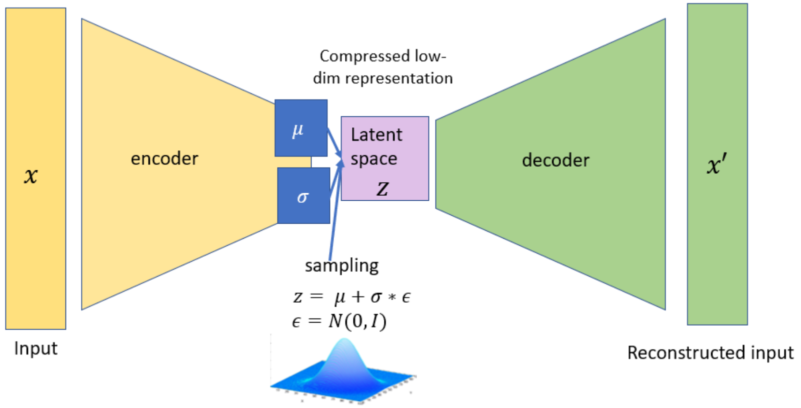
Reconstructed input

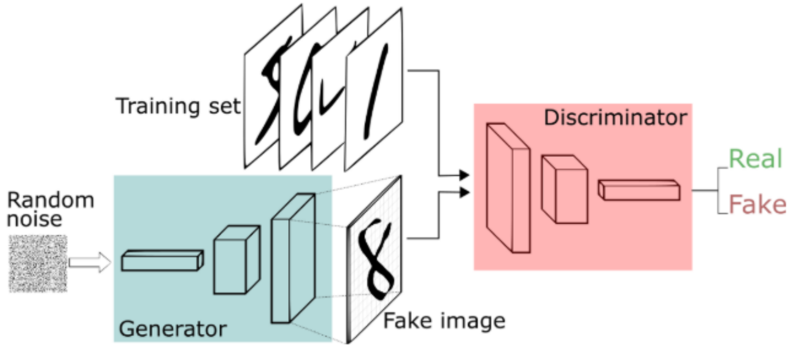
Good properties for sampling and interpolation

The latent space is by design continuous

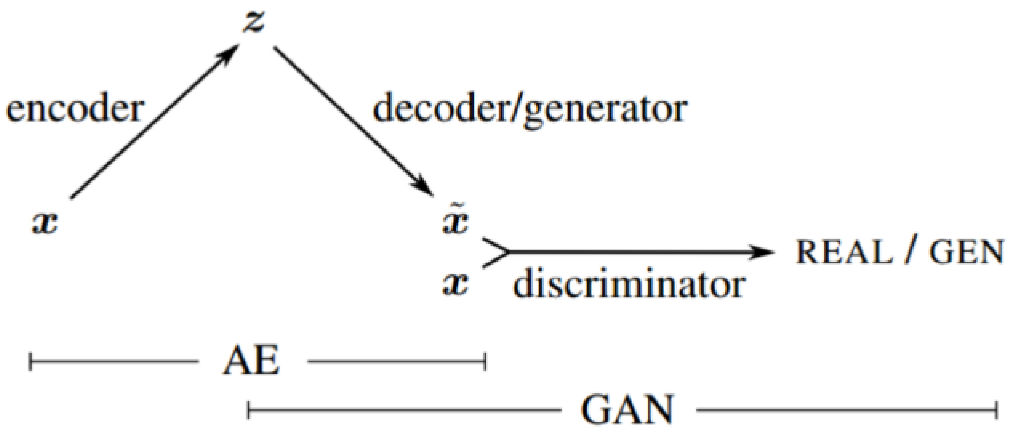
The plot looks symmetrical

Improving VAEs decoder by combining it with GANs





Source:https://towardsdatascience.com/what-the-heck-are-vae-gans-17b86023588a



VAE-GAN architecture, the discriminator from GAN takes input from VAE's decoder

Source: https://arxiv.org/pdf/1512.09300.pdf

Let’s practice…

https://github.com/dimtr/PyDataEHV_workshop

Let’s see an example

let’s help NumFocus committee to find new cities for upcoming pydata events using a generative model

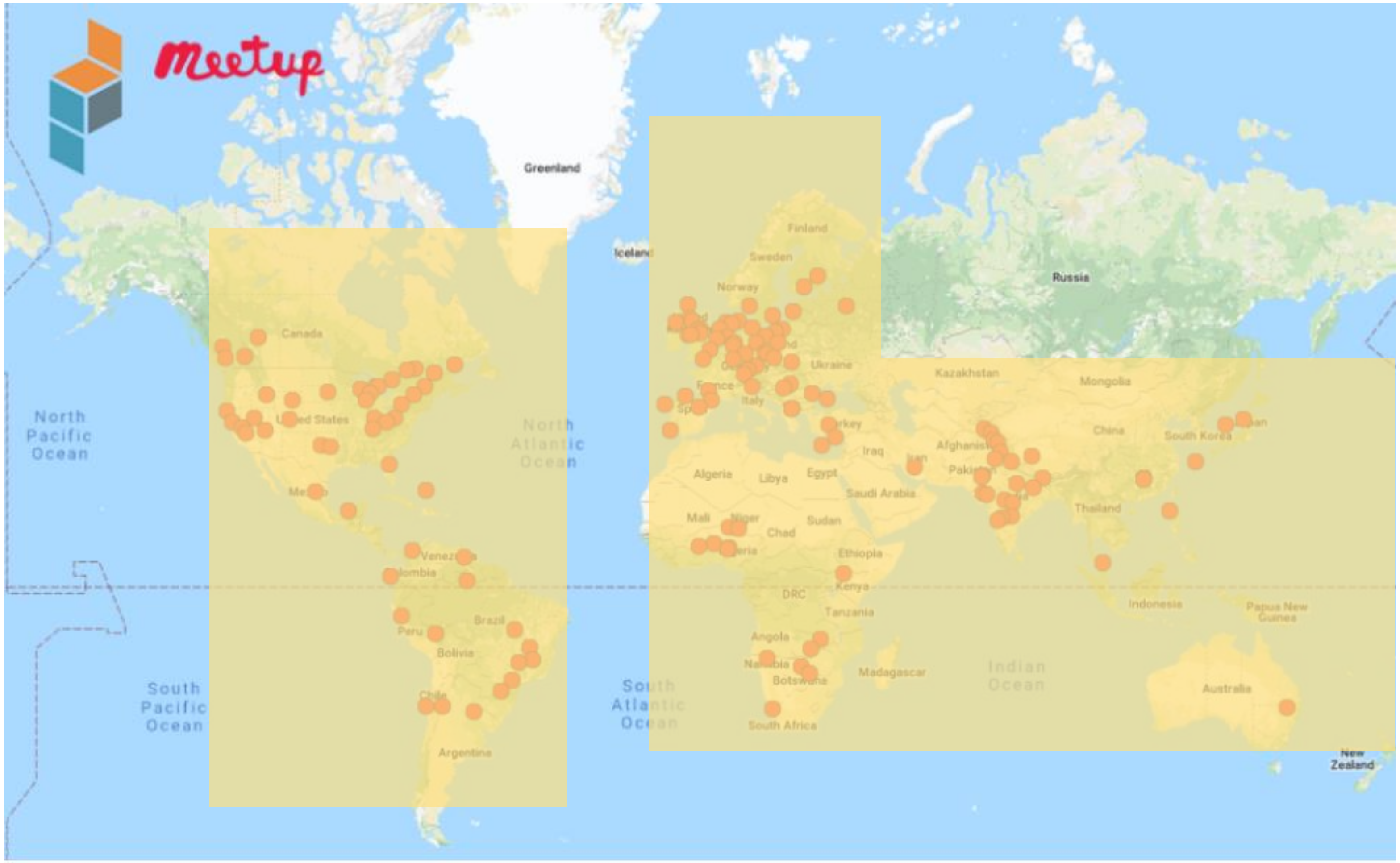
NUMFOCUS
OPEN CODE = BETTER SCIENCE

PyData

Observations: Pydata events across the world

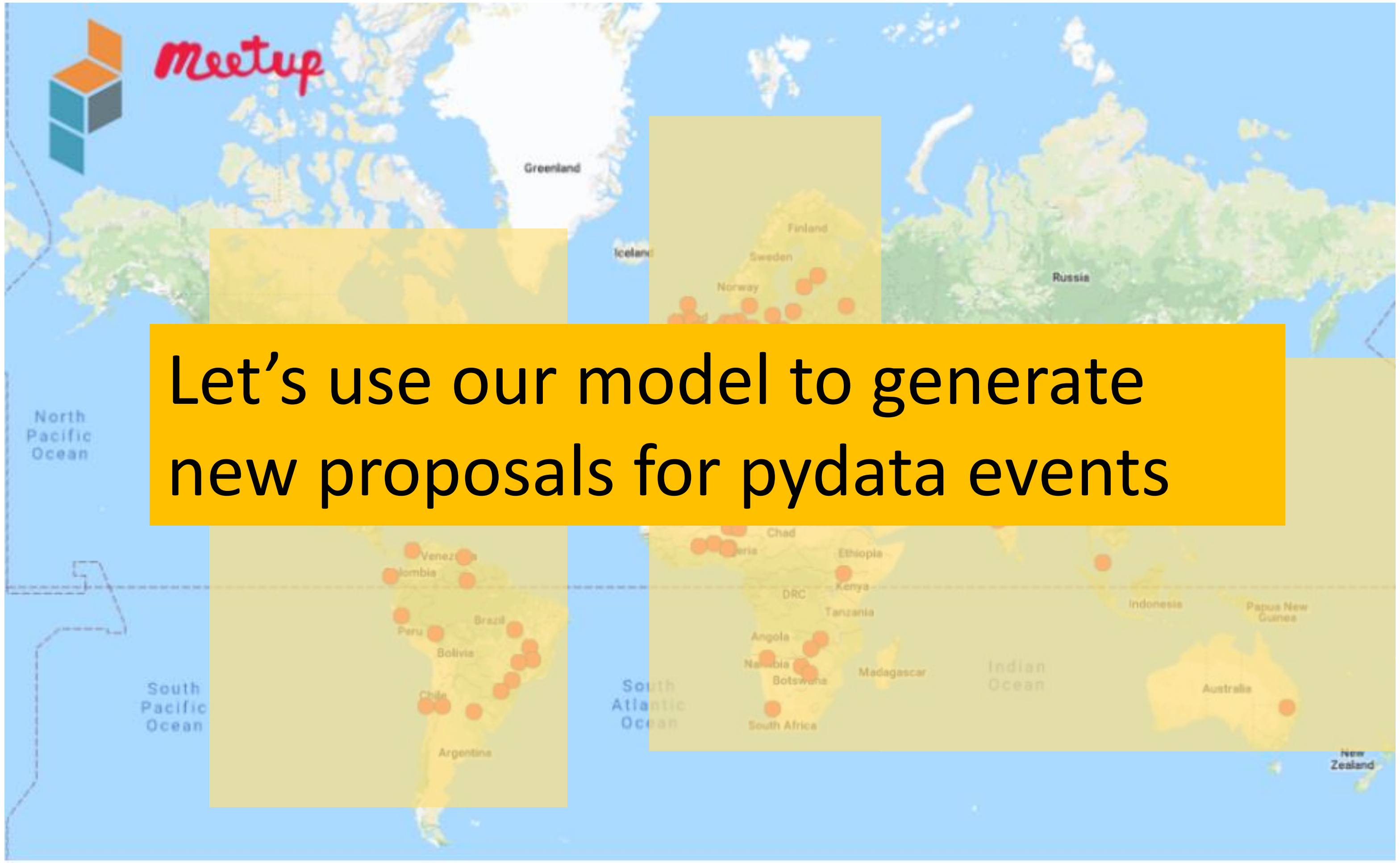
Cities with pydata events across the world

Oversimplified model



Cities with pydata events across the world

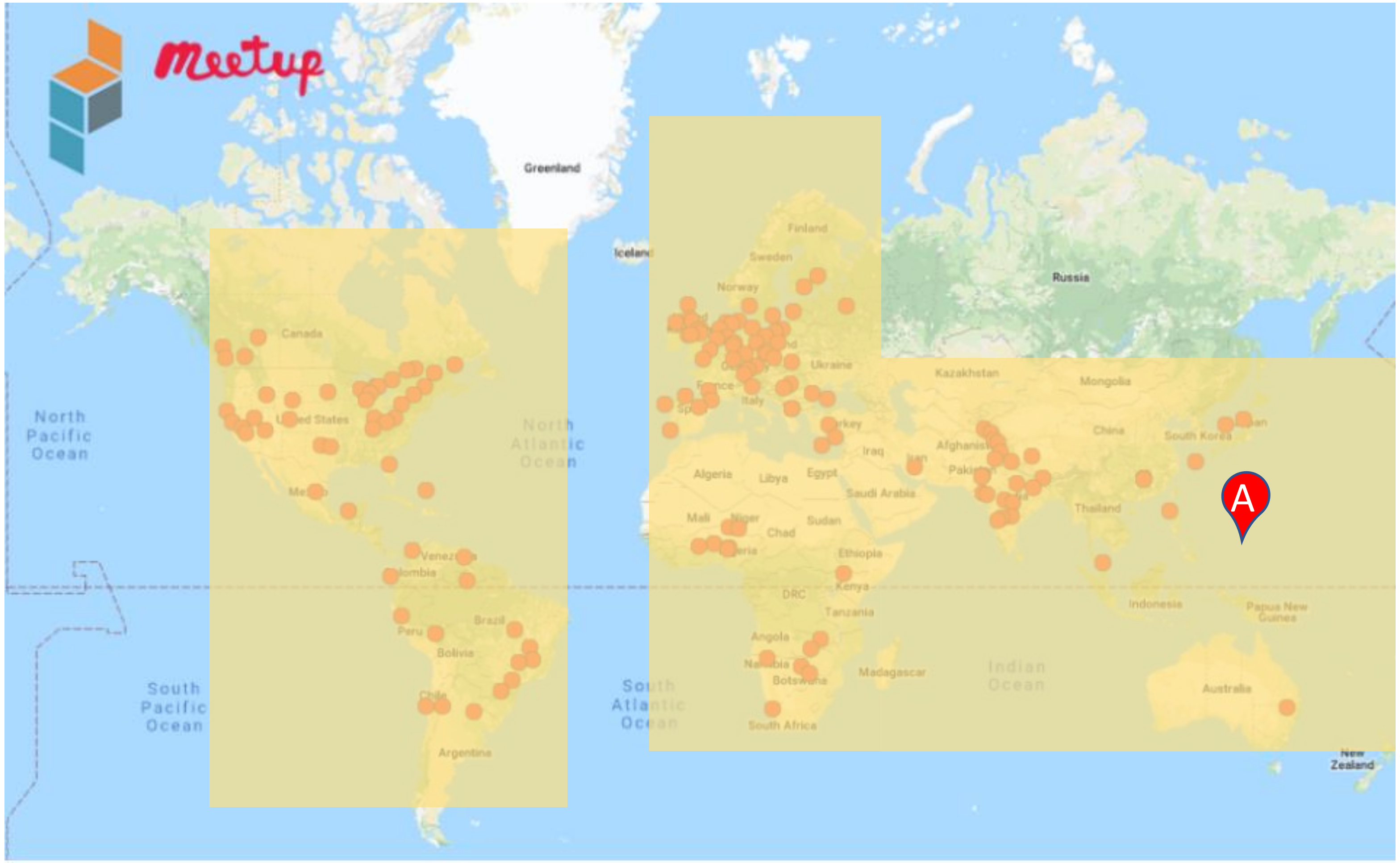
Oversimplified model



Let’s use our model to generate new proposals for pydata events

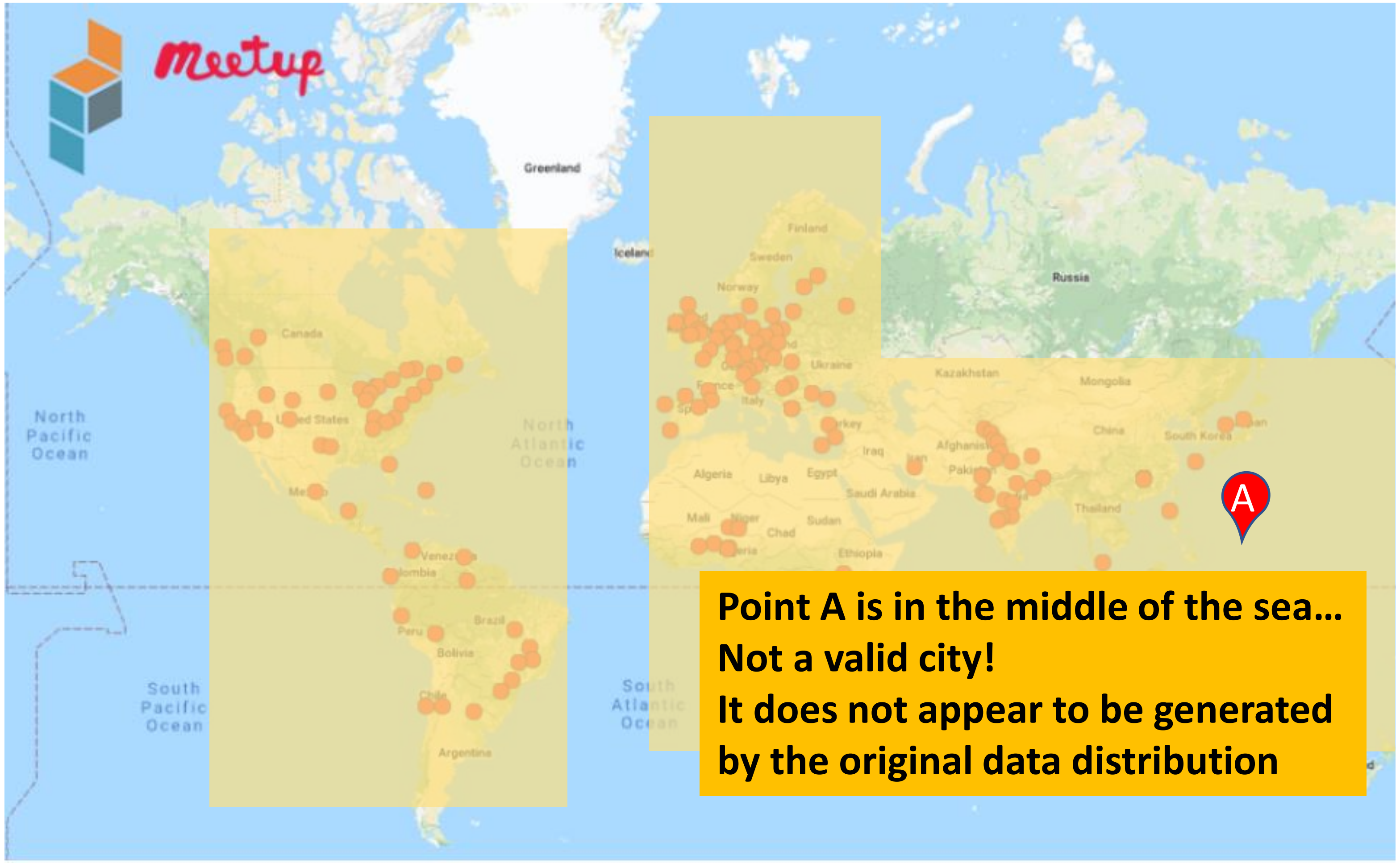
Cities with pydata events across the world

We sample from the model distribution city A



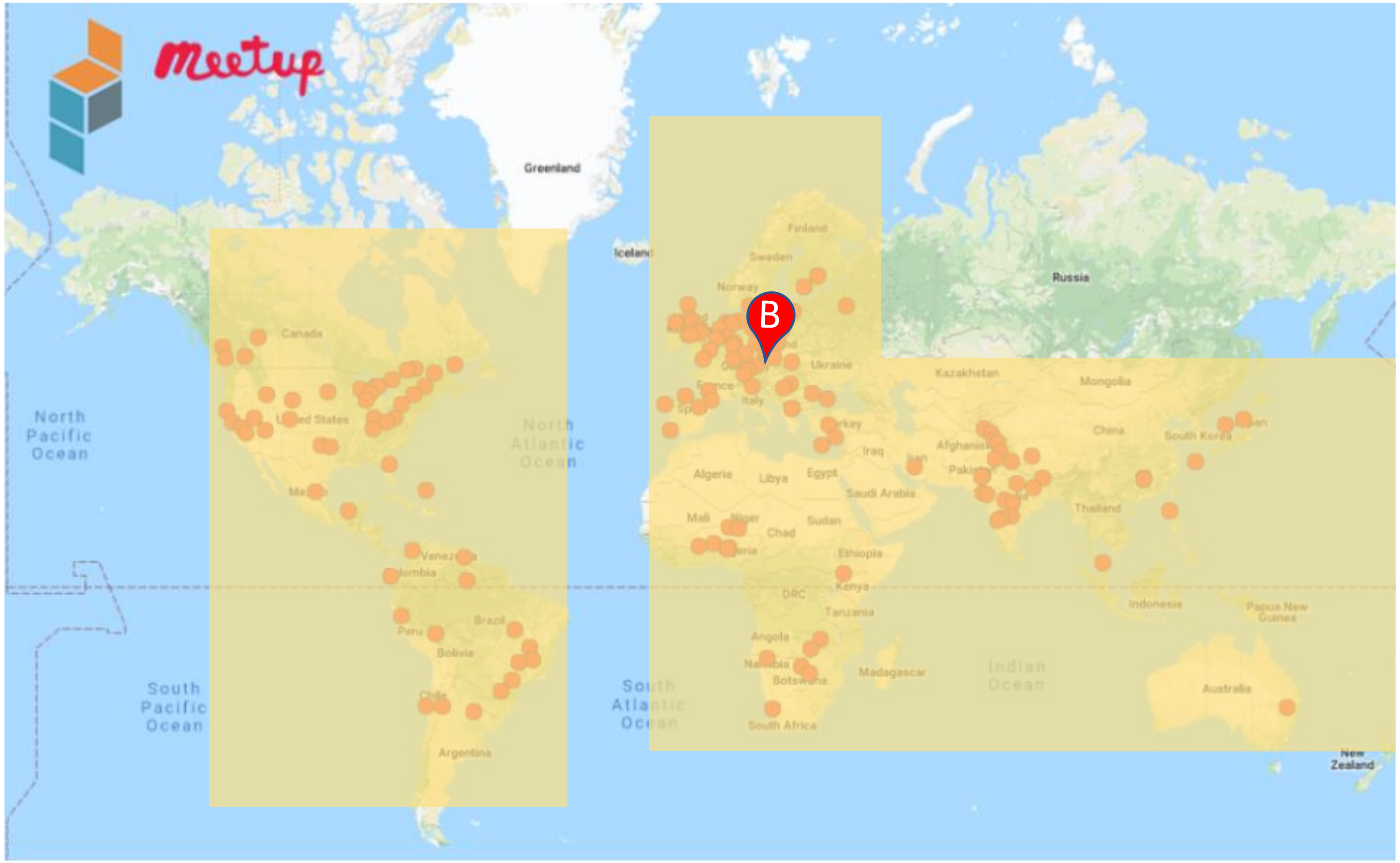
Cities with pydata events across the world

We sample from the model distribution city A



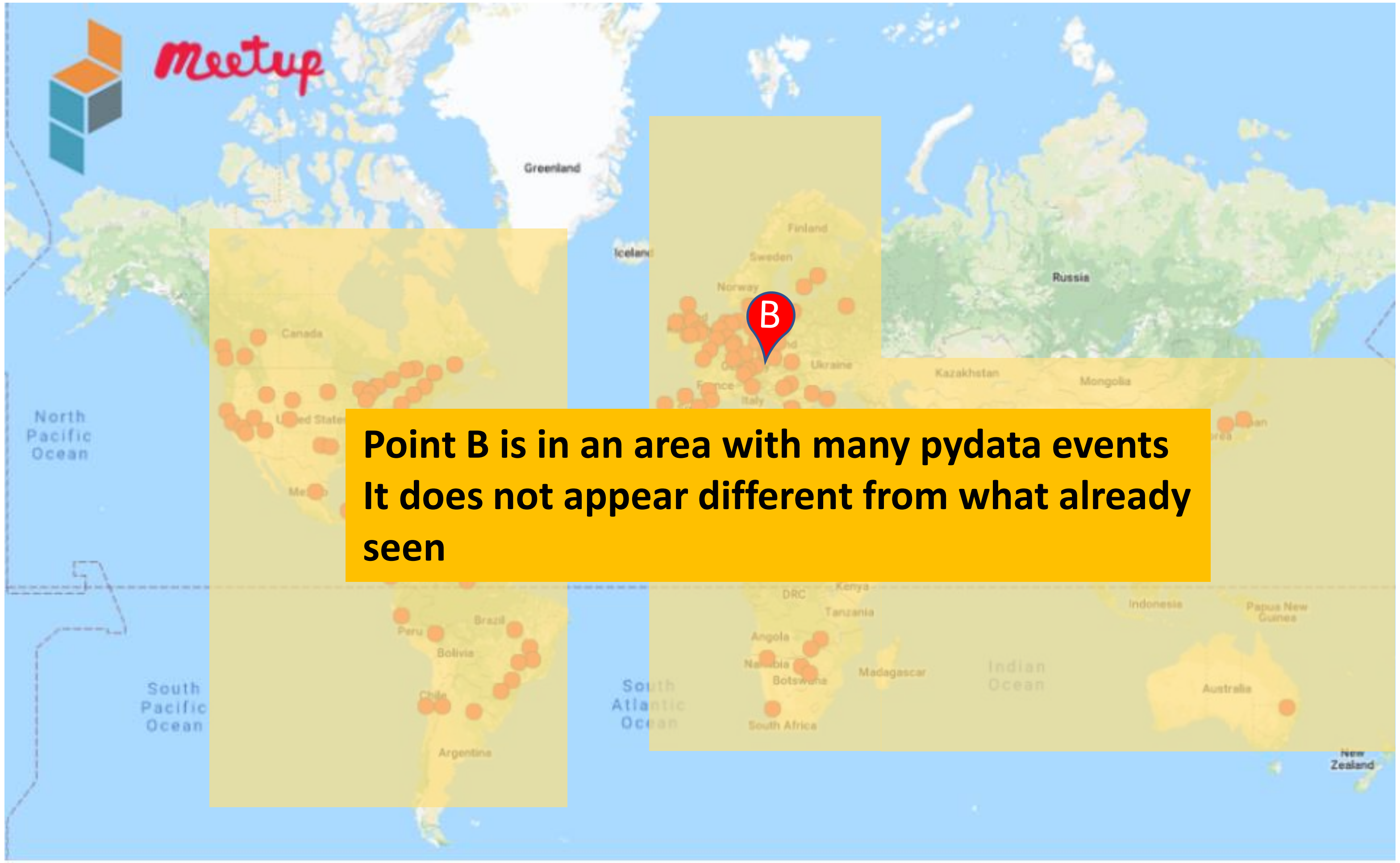
Cities with pydata events across the world

We sample from the model distribution city B



Cities with pydata events across the world

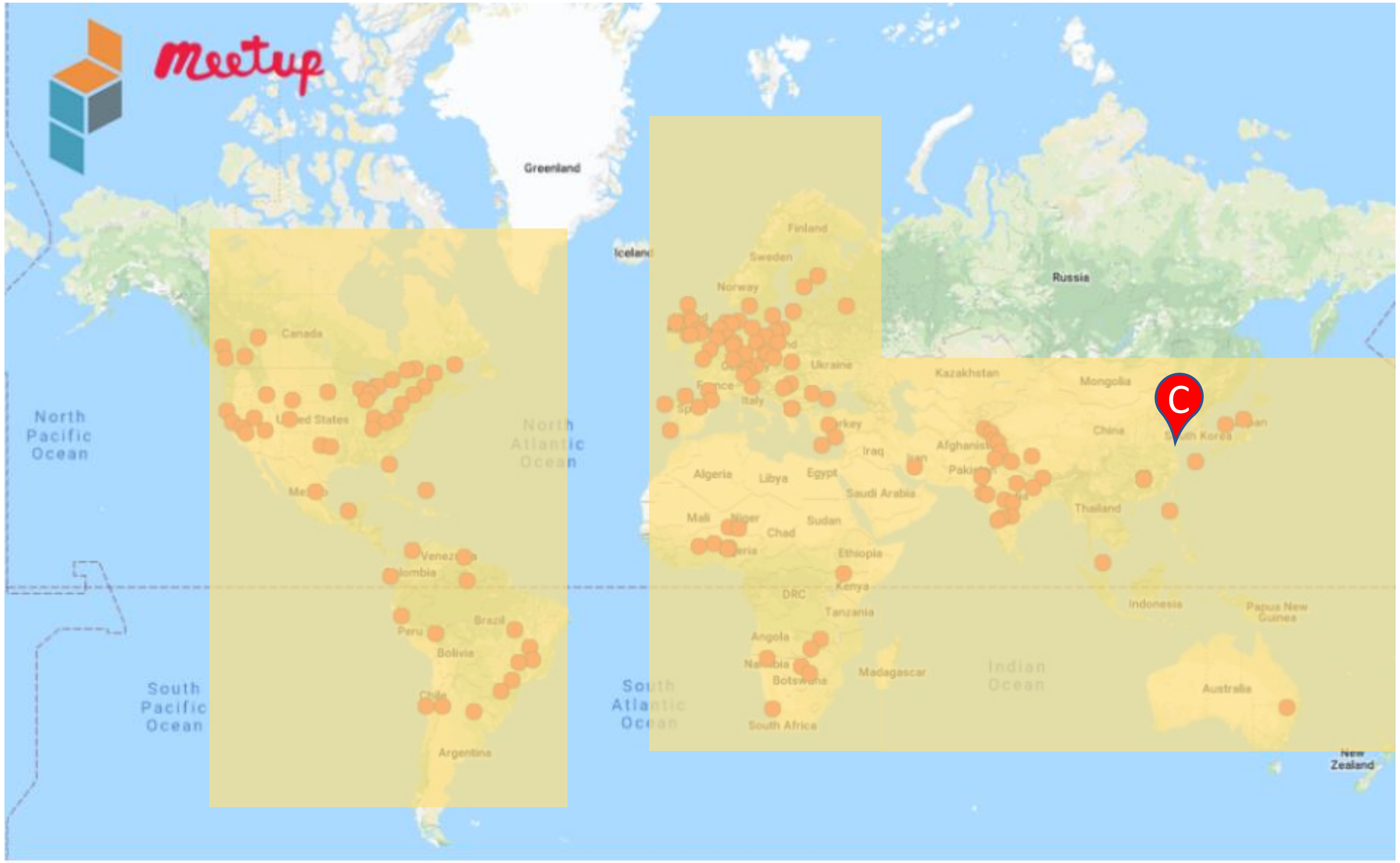
We sample from the model distribution city B



Point B is in an area with many pydata events
It does not appear different from what already seen

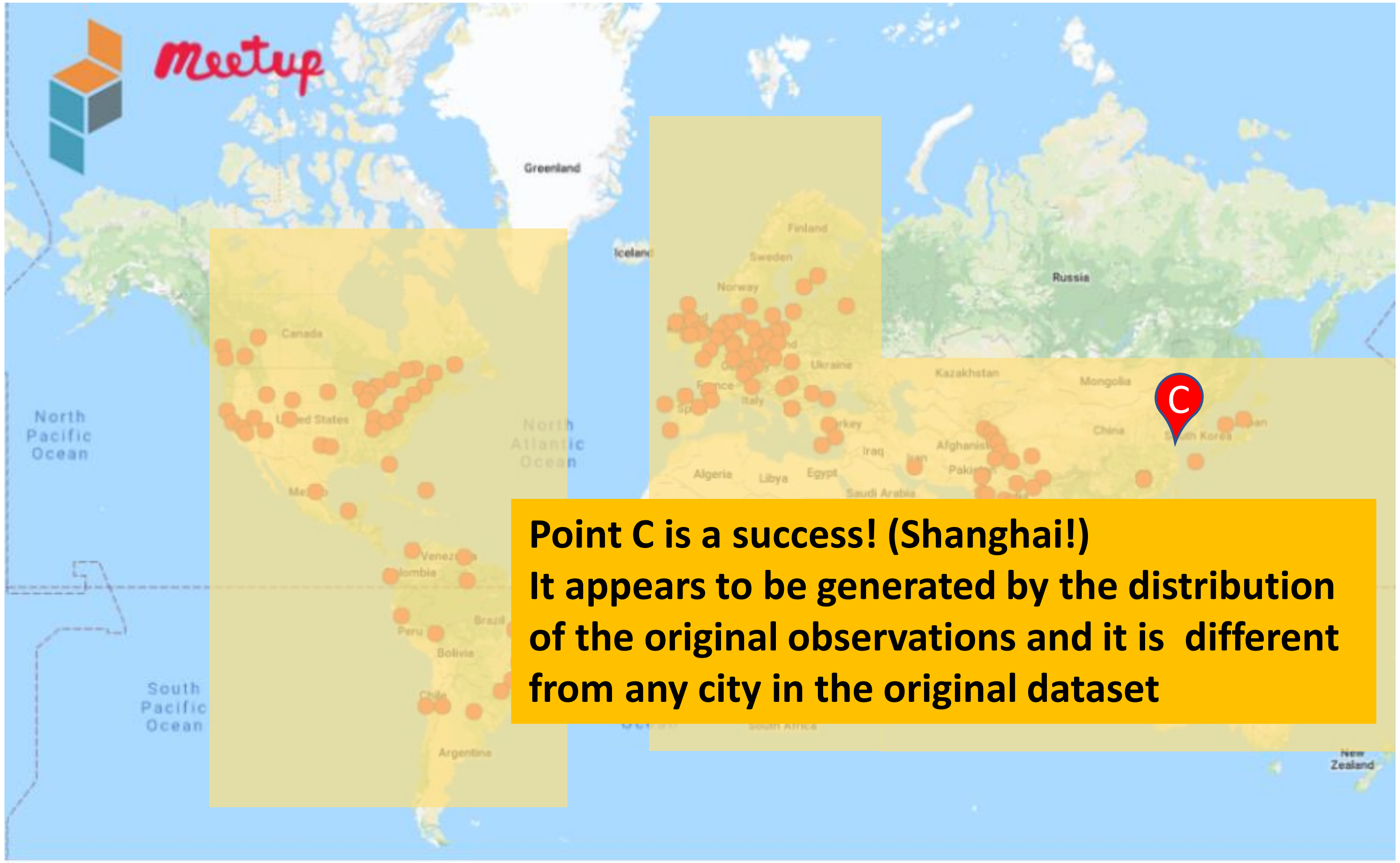
Cities with pydata events across the world

We sample from the model distribution city C



Cities with pydata events across the world

We sample from the model distribution city C



Point C is a success! (Shanghai!)
It appears to be generated by the distribution of the original observations and it is different from any city in the original dataset

Cities with pydata events across the world

return