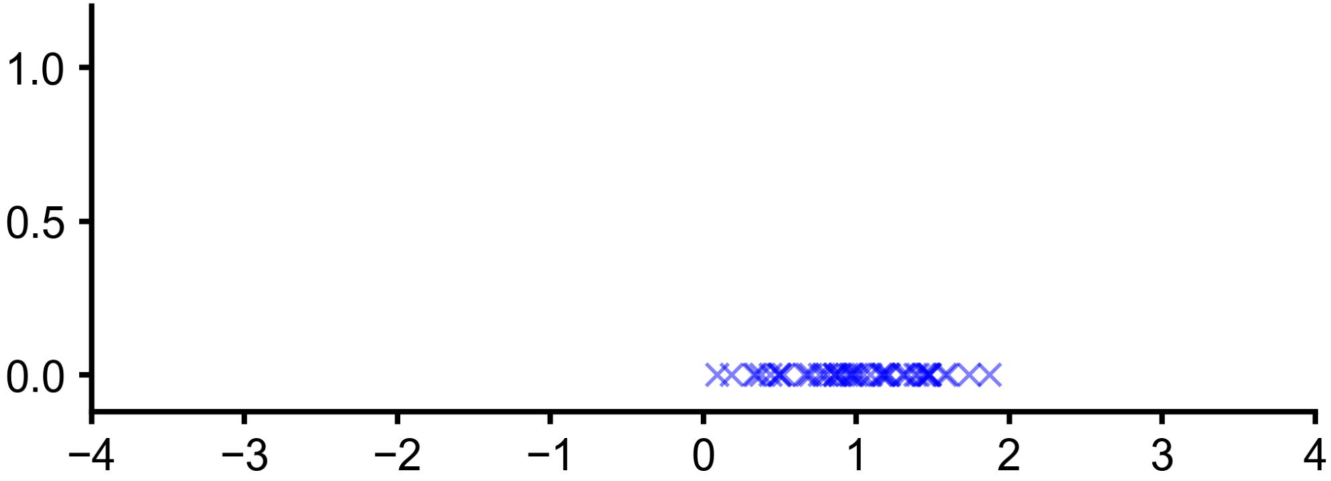
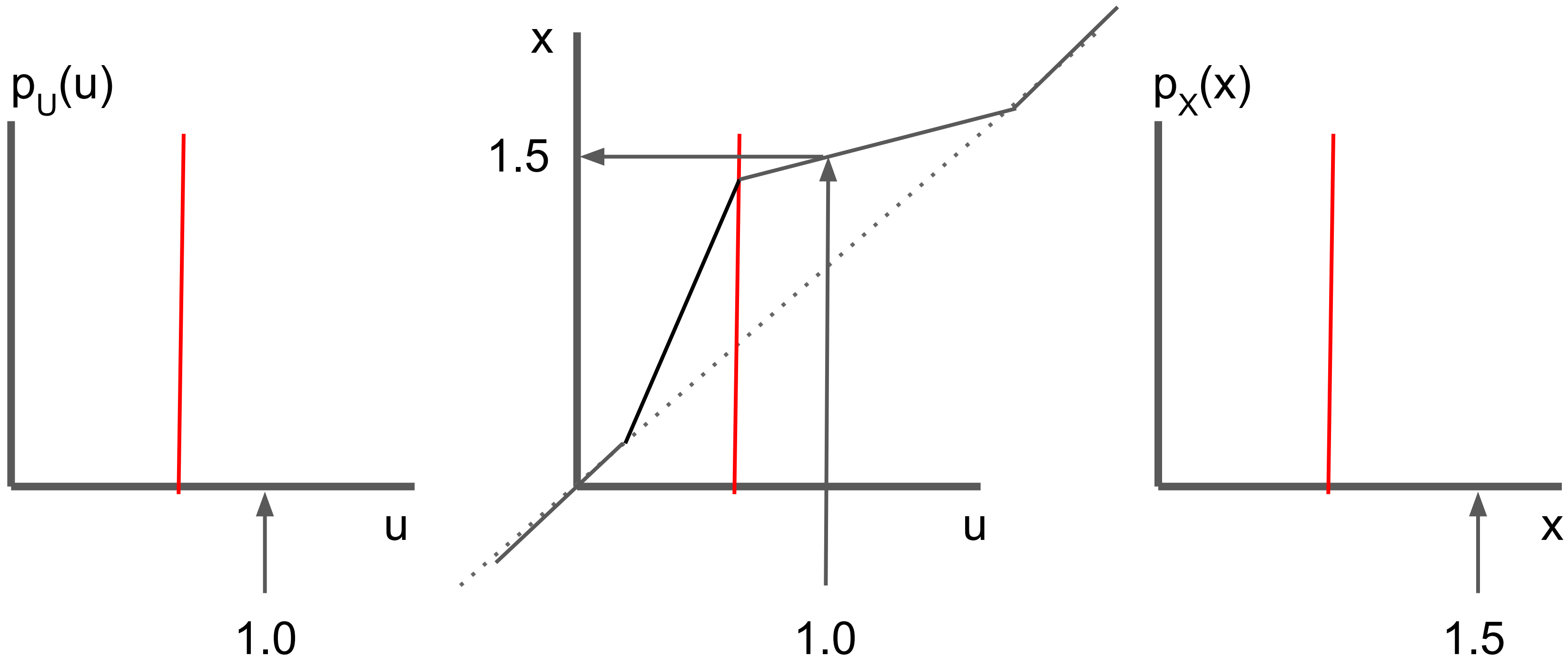
Introduction to normalizing flows

We will first talk about Density estimation (not: conditional density estimation) e.g.

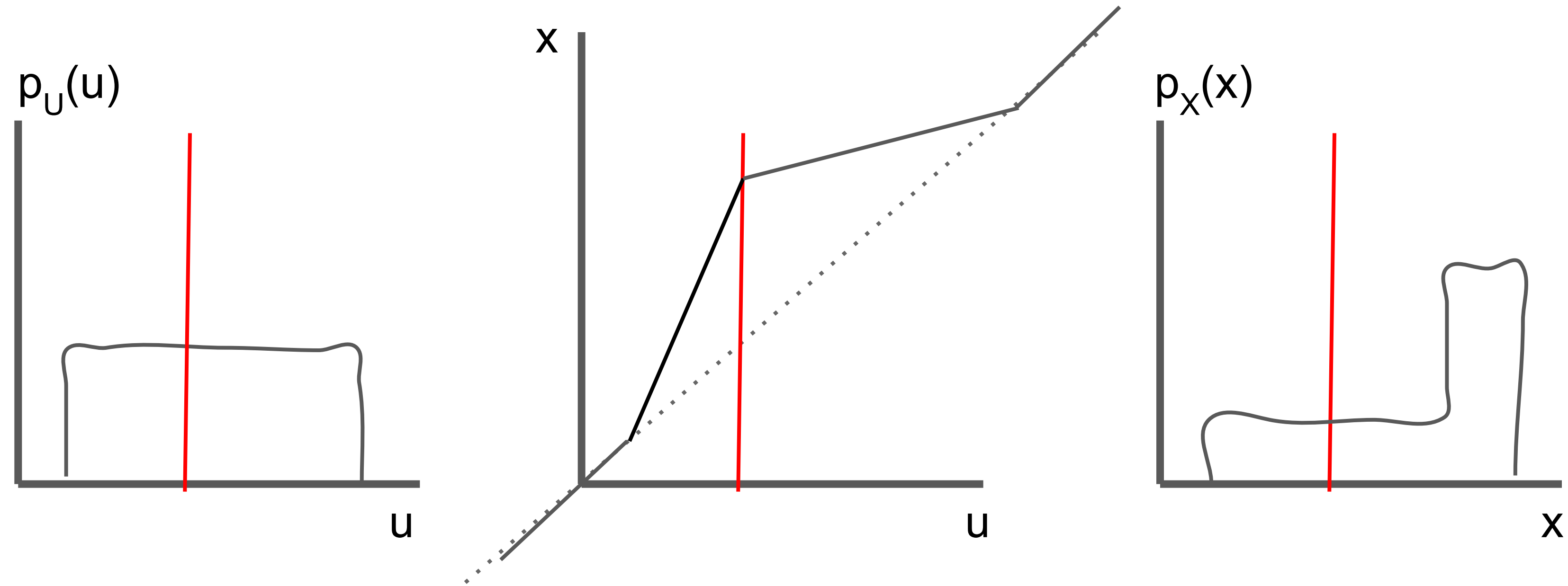


But first: change of variables in probability

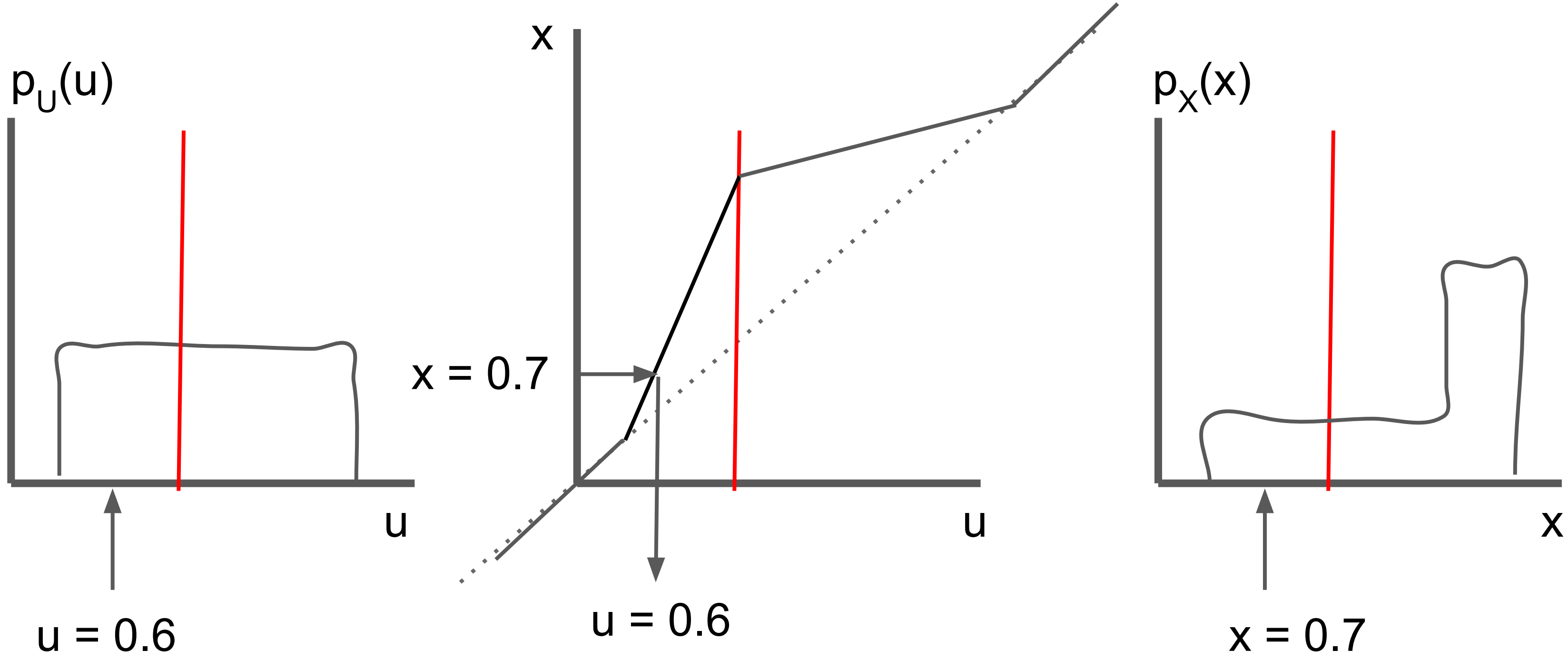
Transforming samples



Transforming distributions

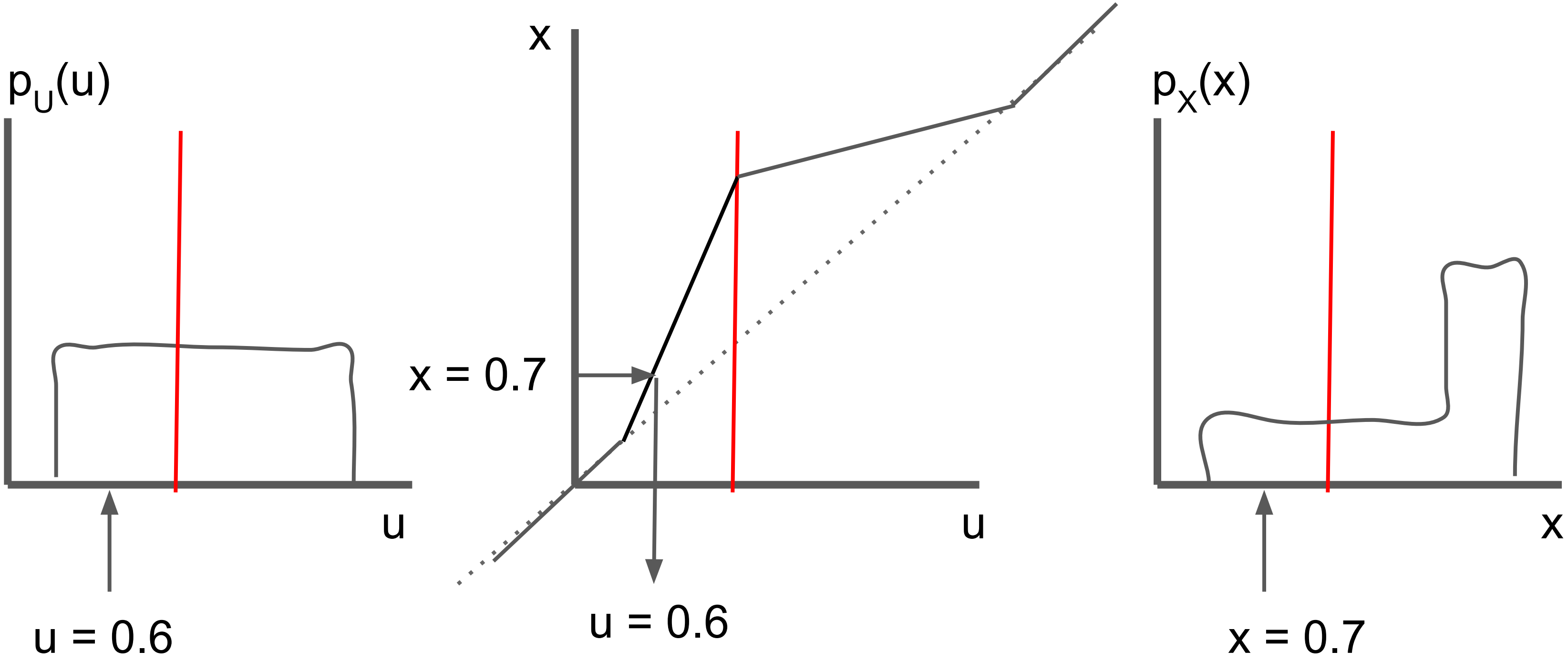


How can we evaluate $p_X(x)$?



$$u = T^{-1}(x)$$

How can we evaluate $p_X(x)$?



$$u = T^{-1}(x)$$

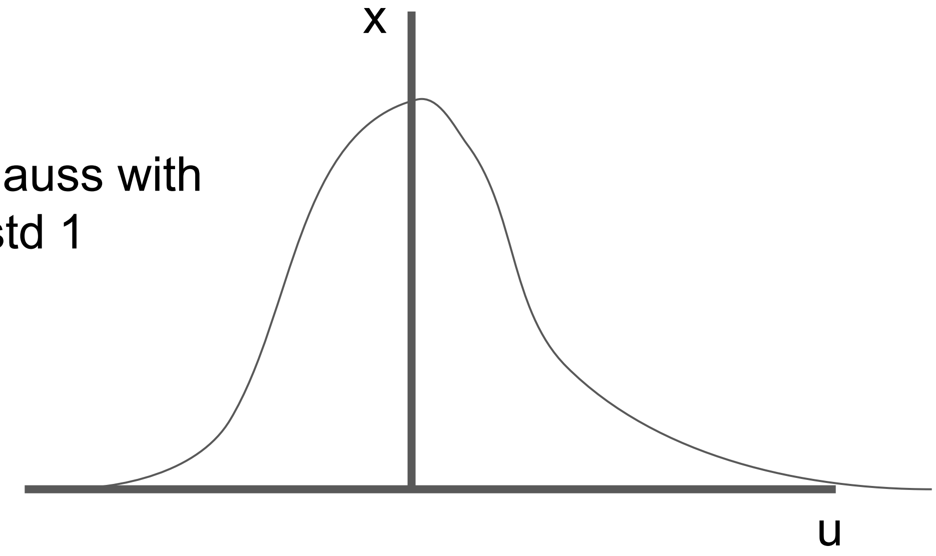
$$p_X(x) = p_U(u)\frac{1}{\frac{dT}{du}(u)}$$

derivative

Let’s build our first normalizing flow!

Step 1: pick a base distribution

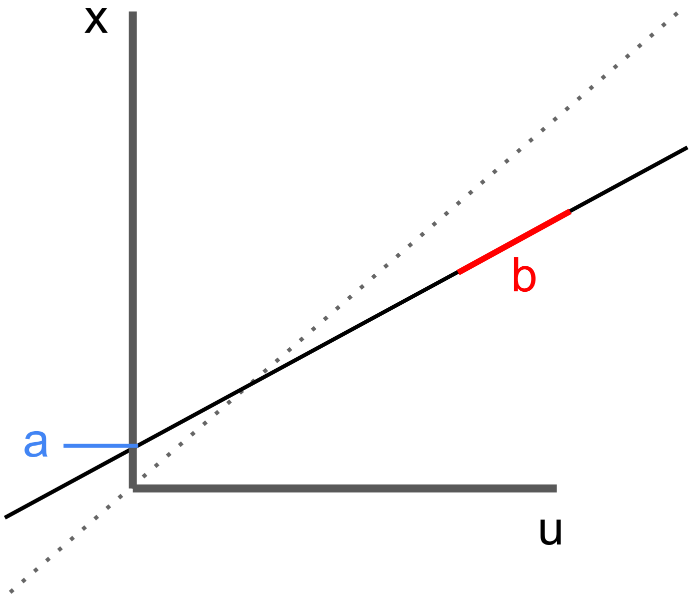
Usually Gauss with
mean 0, std 1



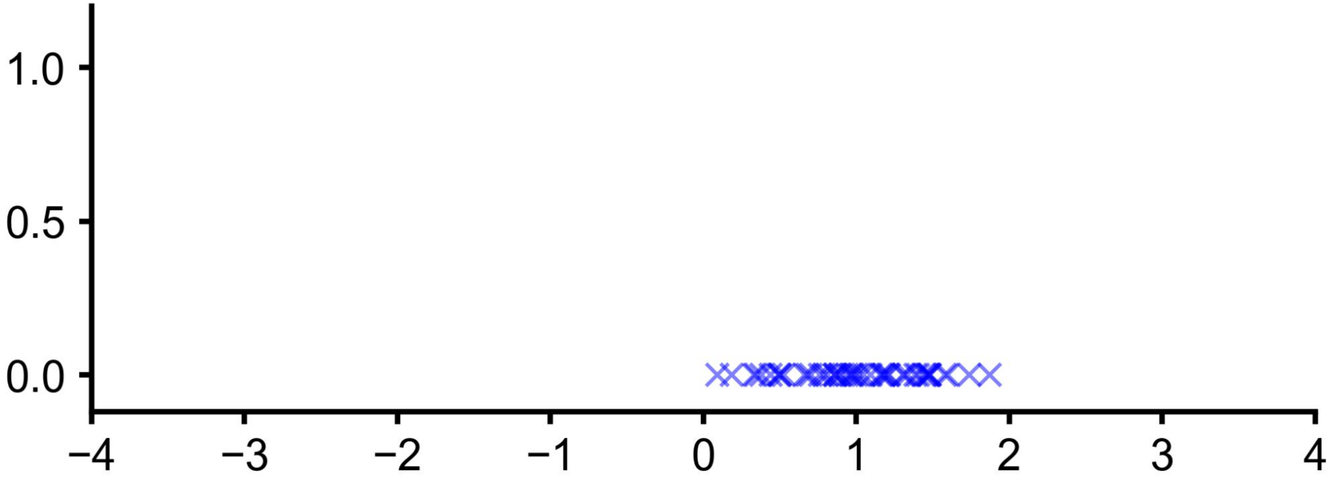
Step 2: decide on a transformation

Learnable parameters:

Offset a
Slope b

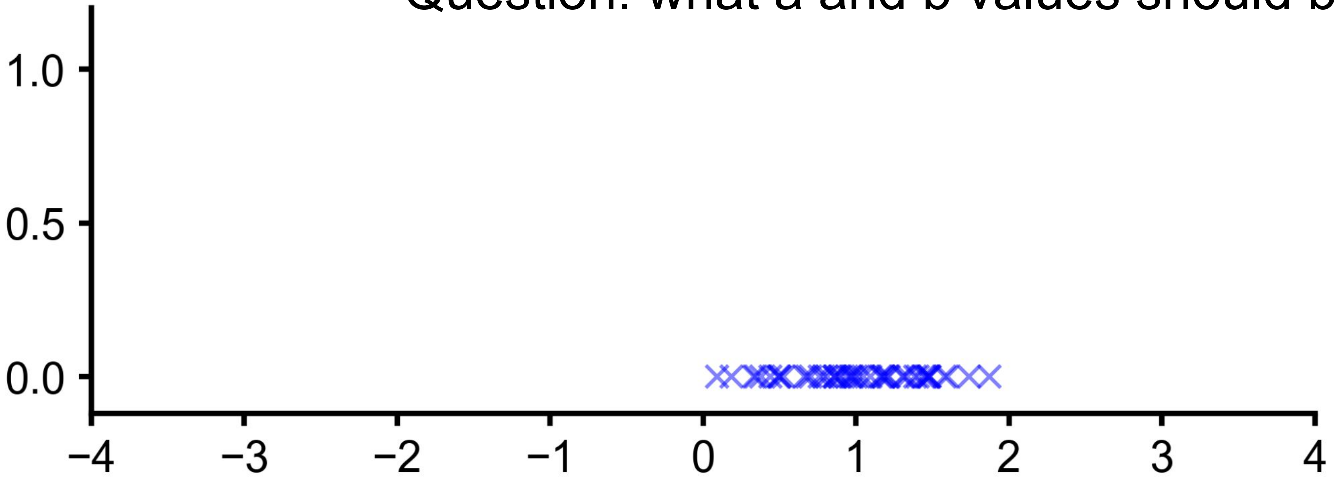


Let’s look at data



Let’s look at data

Question: what a and b values should be learnt?



-> Move to jupyter

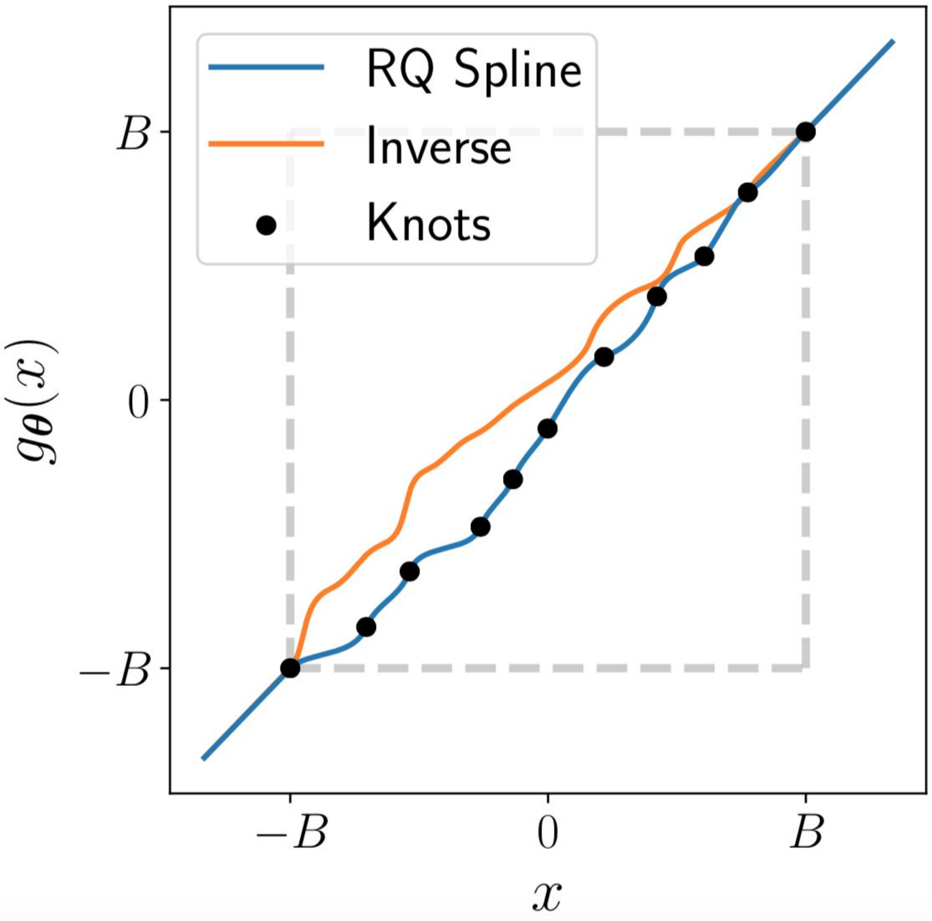
There are two problems right now:

1) If we use a linear transformation, it stays Gaussian
2) How do we extend this to multiple dimensions and model correlations?

1:

MAF indeed uses a linear transform

NSF uses splines (neural **spline** flow)



2:

u1

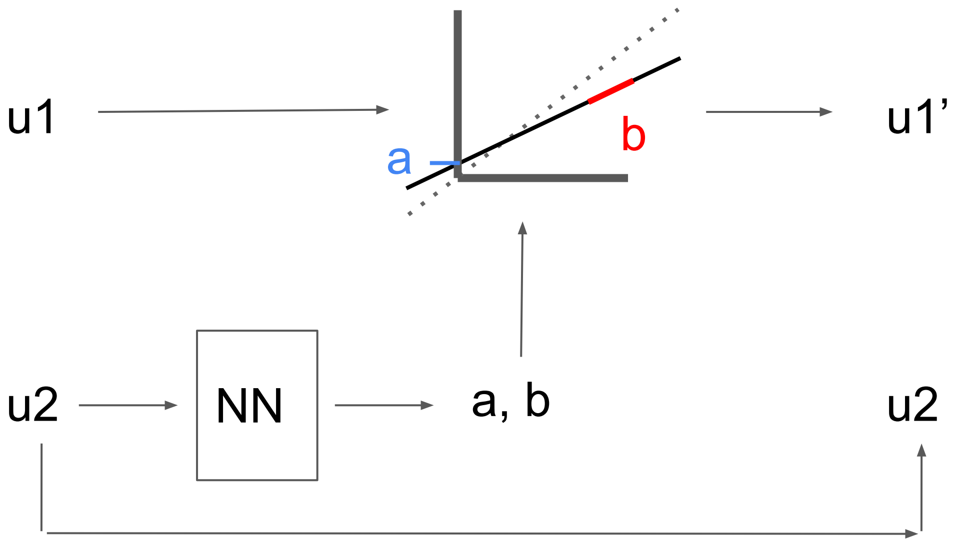
u2

2:

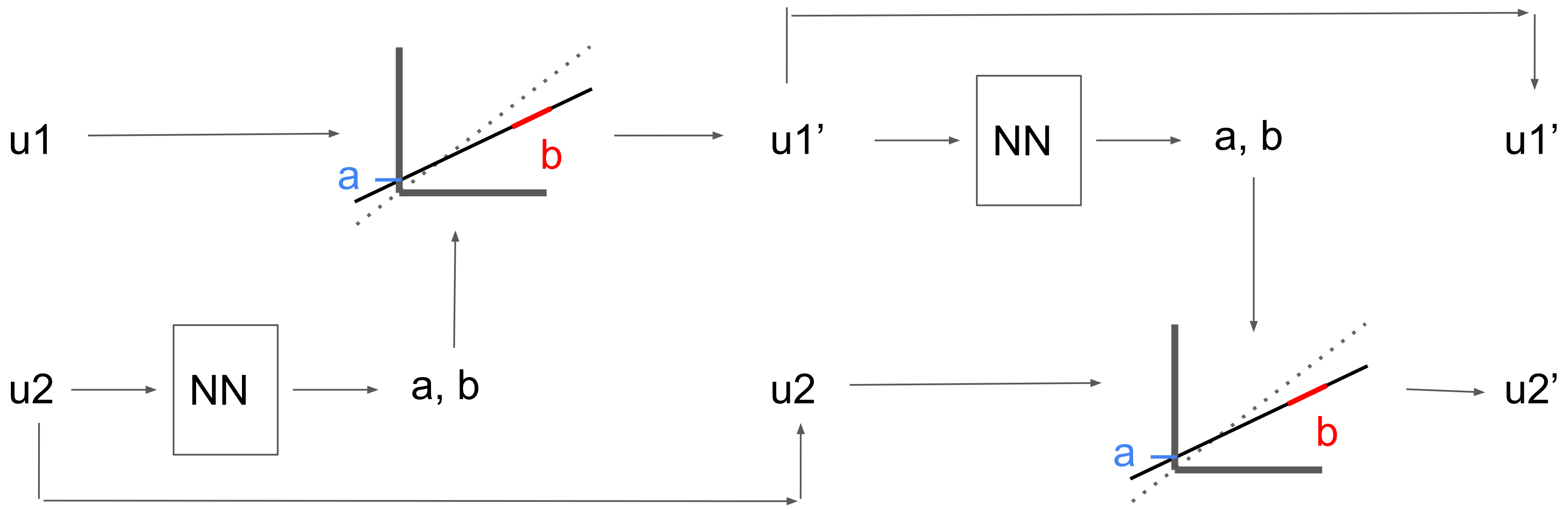
u1

$$u2 \longrightarrow \boxed{\text{NN}} \longrightarrow a, b$$

2:



2:



What about conditional density estimation p(x | y)?

2: