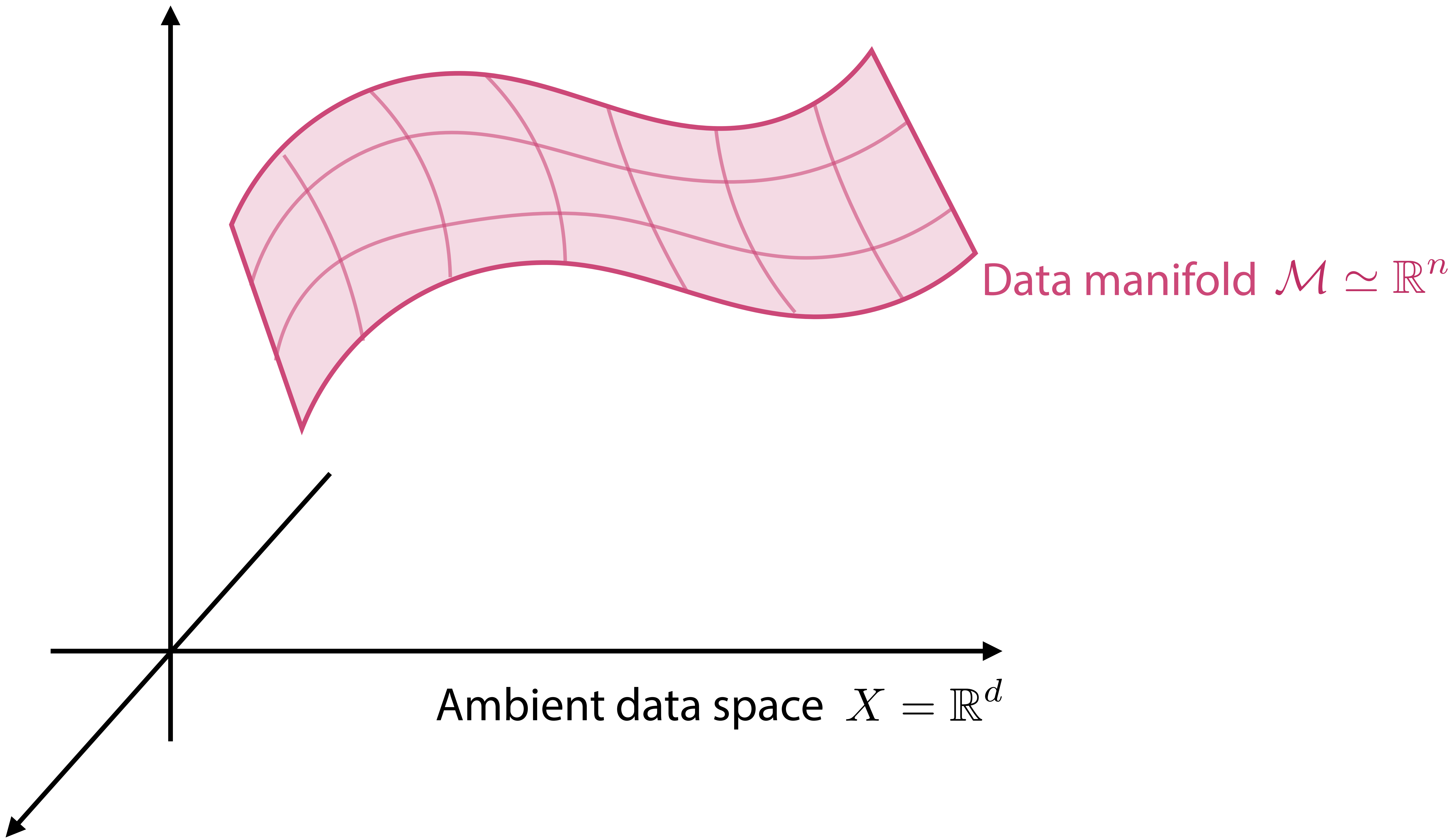
Flows for simultaneous manifold learning and density estimation

Johann Brehmer
New York University

Kyle Cranmer
New York University

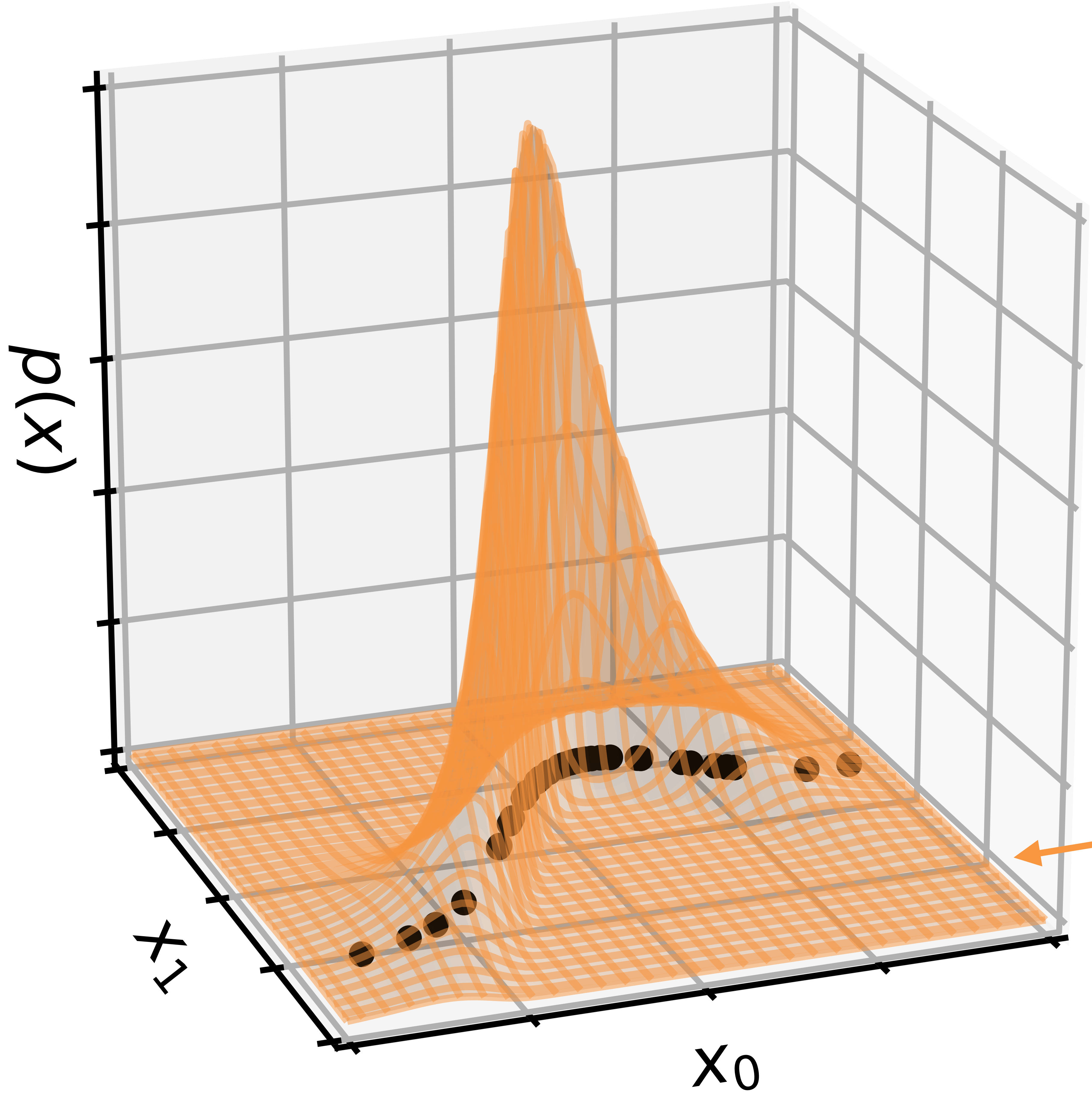
NeurIPS 2020

NYU MSDSE dianahep iris hep The SCAILFIN Project scailfin.github.io BROOKHAVEN NATIONAL LABORATORY | Scientific Data and Computing Center



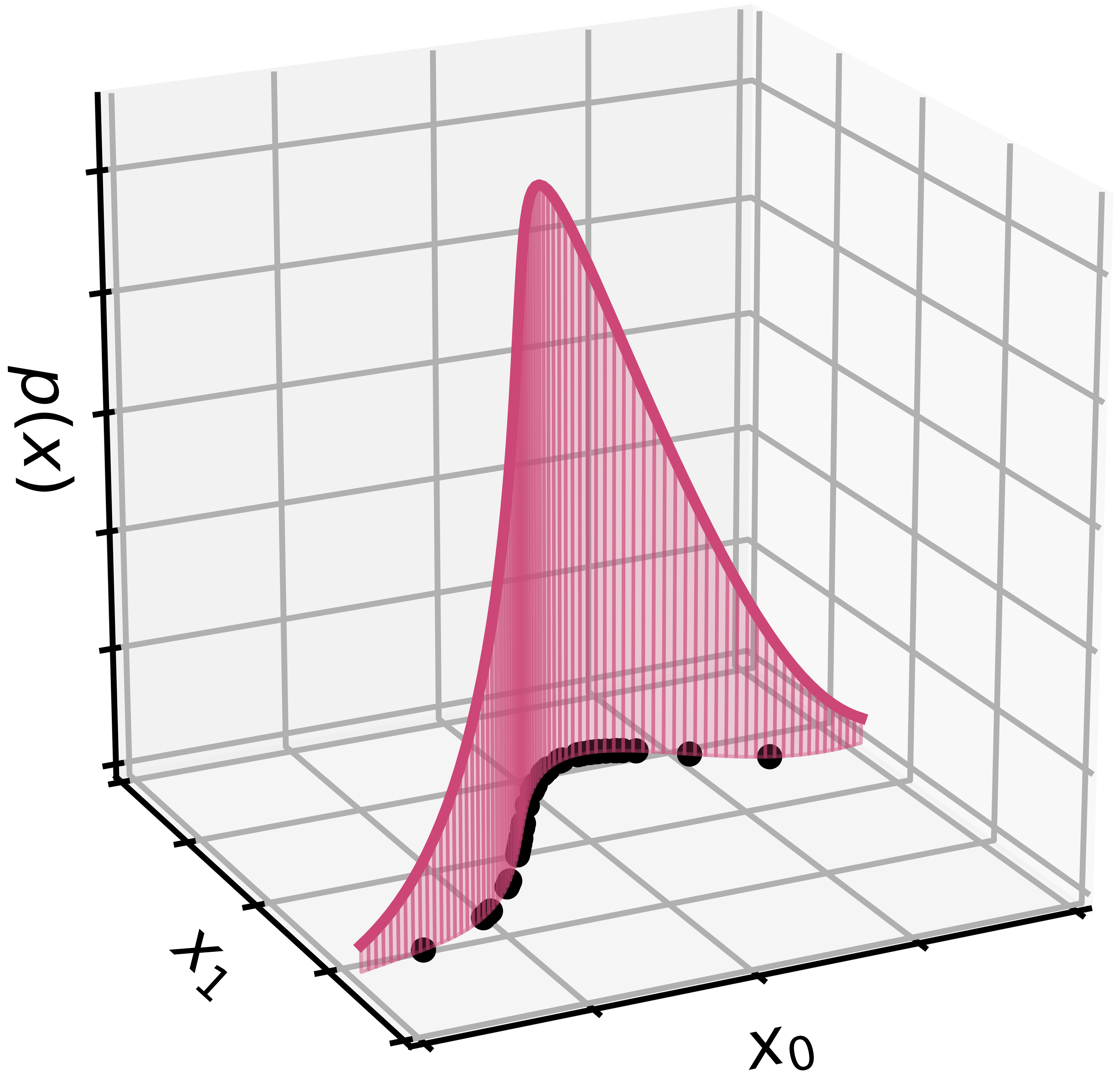
Data manifold $\mathcal{M} \simeq \mathbb{R}^n$
Ambient data space $X = \mathbb{R}^d$

Normalizing flow in ambient data space:

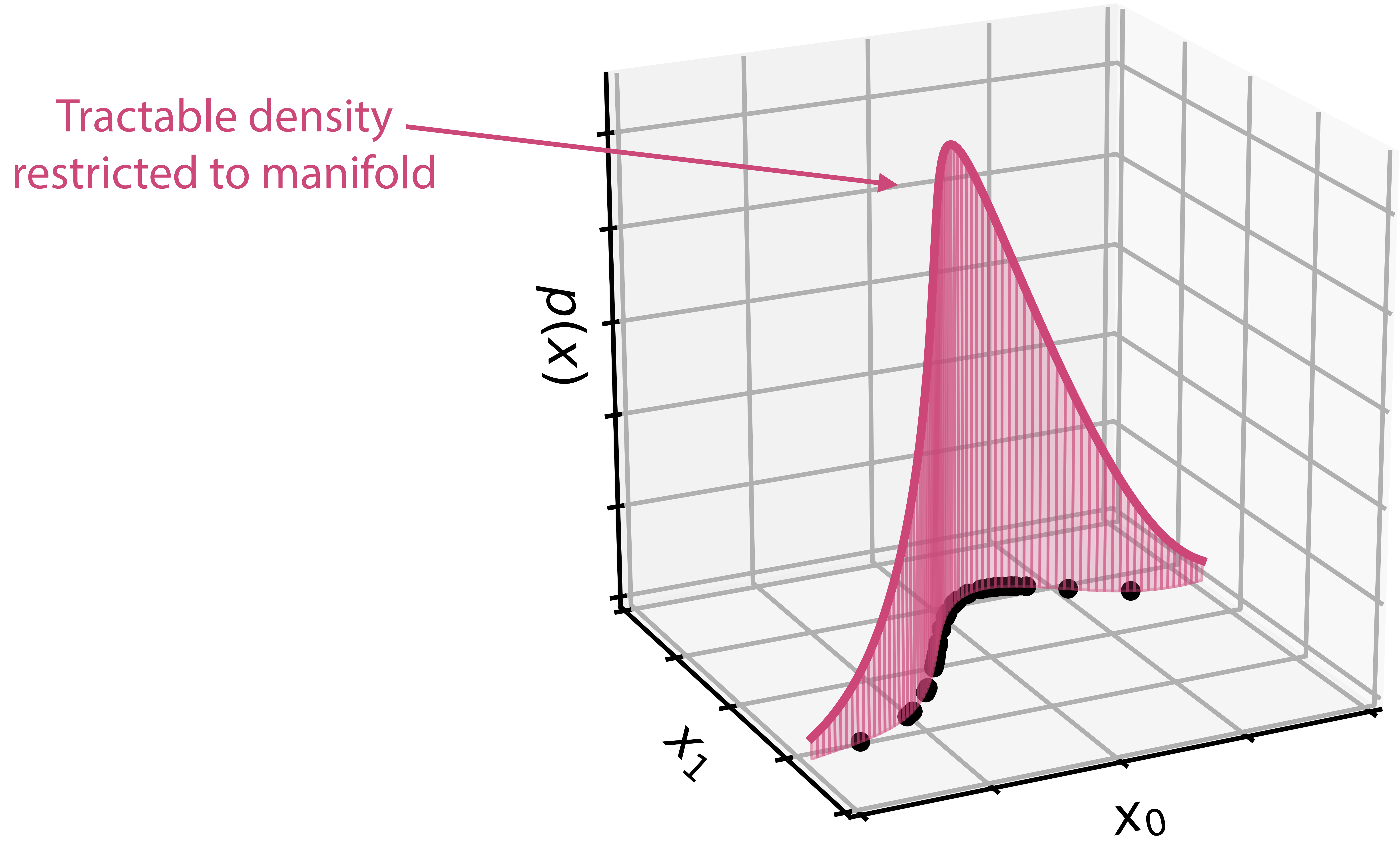


Density has support
off the data manifold

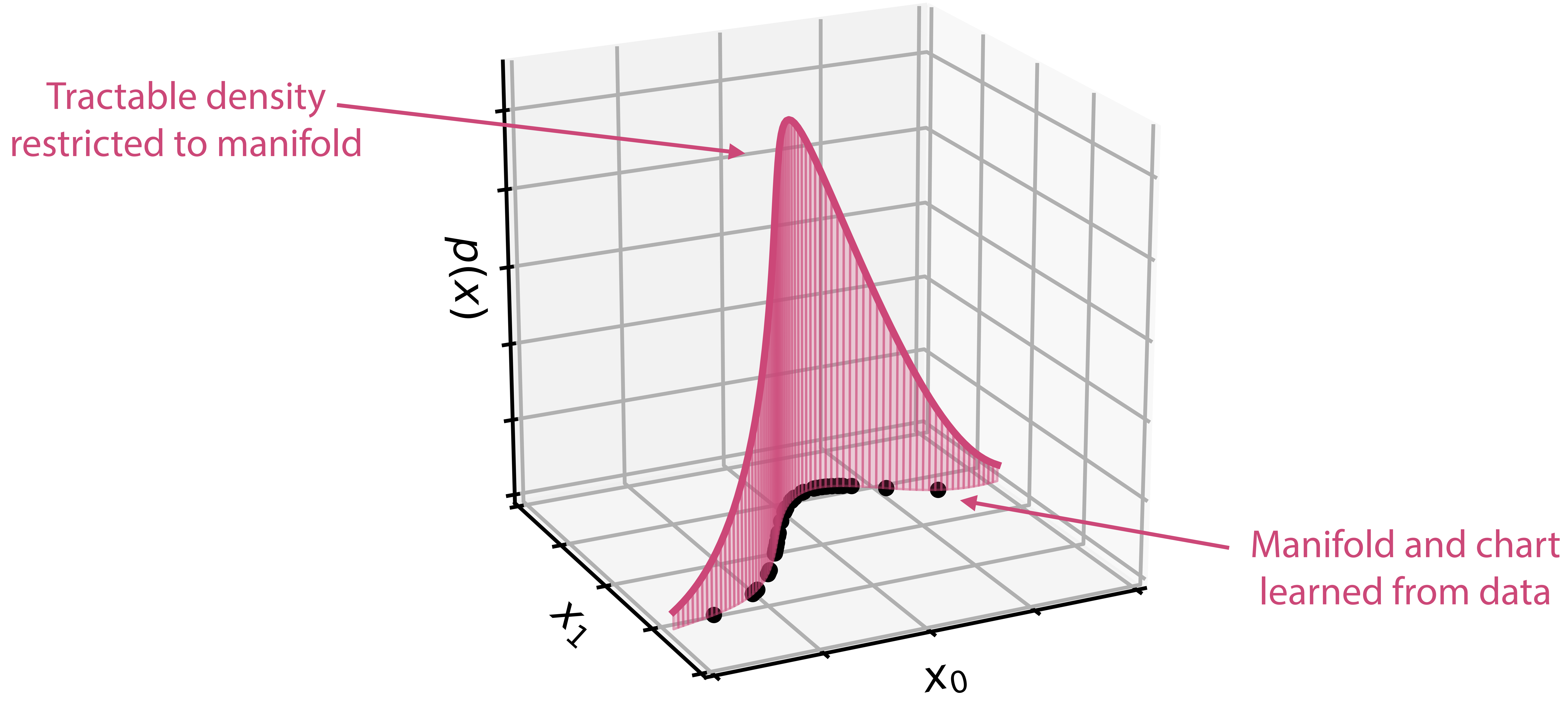
$\mathcal{M}$-flow:



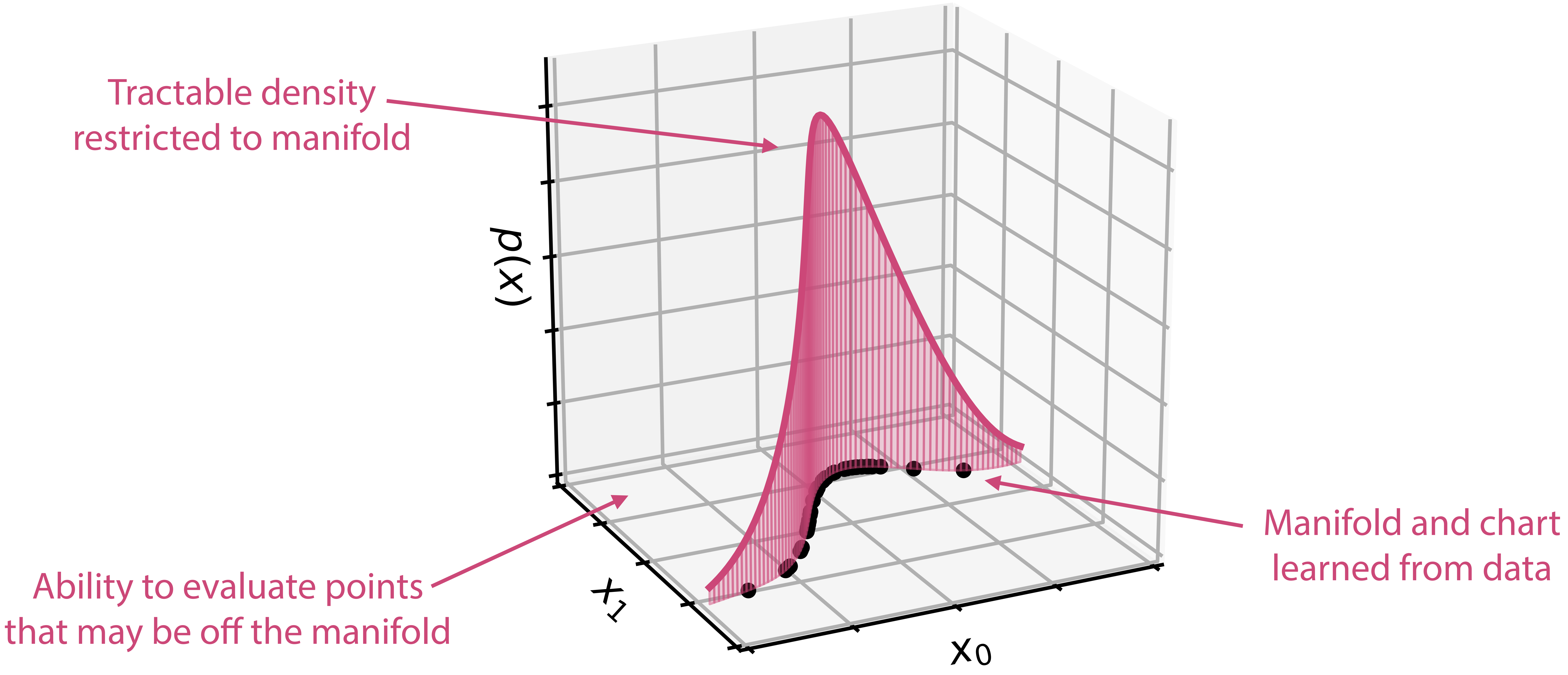
$\mathcal{M}$-flow:

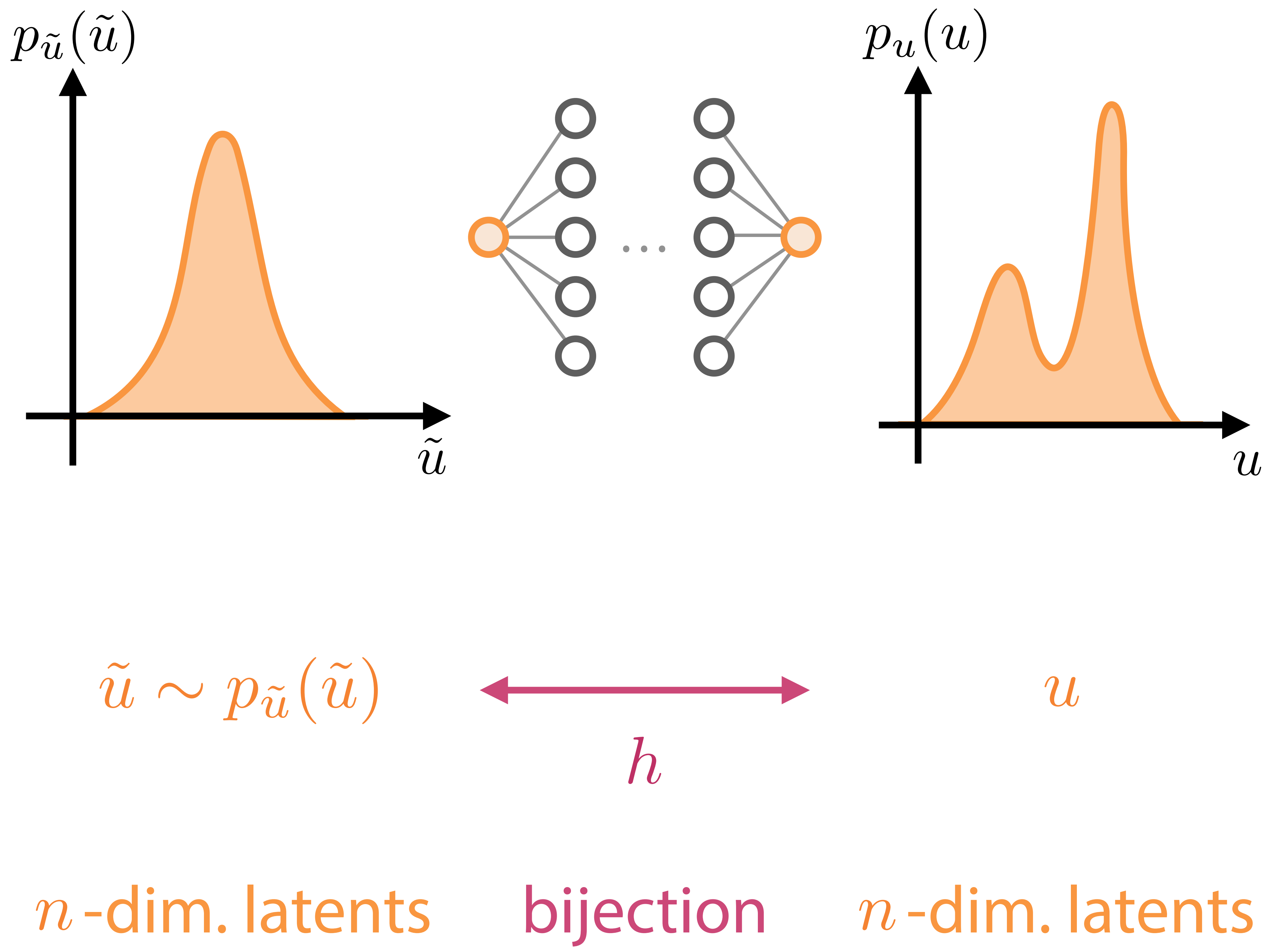


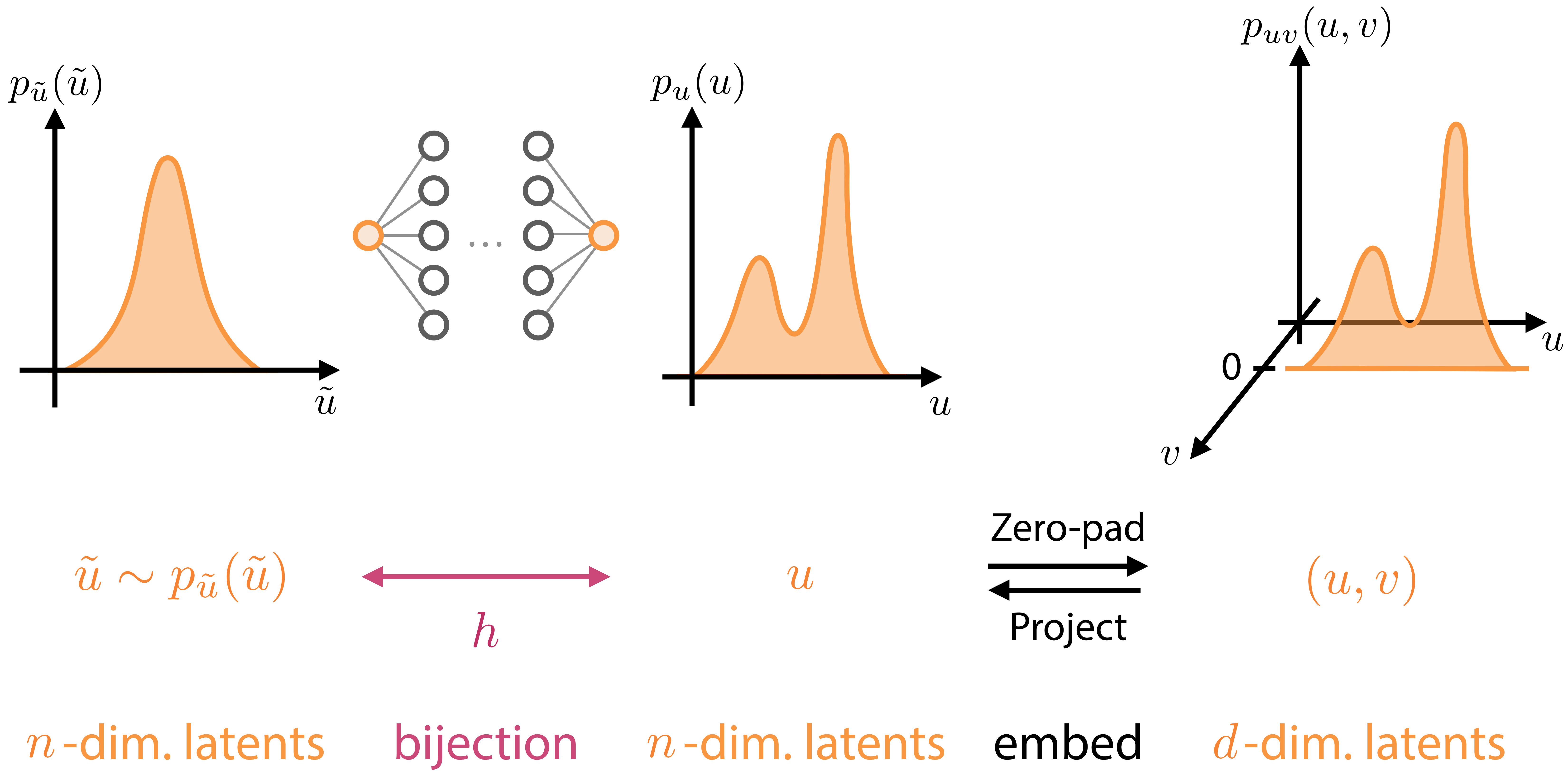
$\mathcal{M}$-flow:



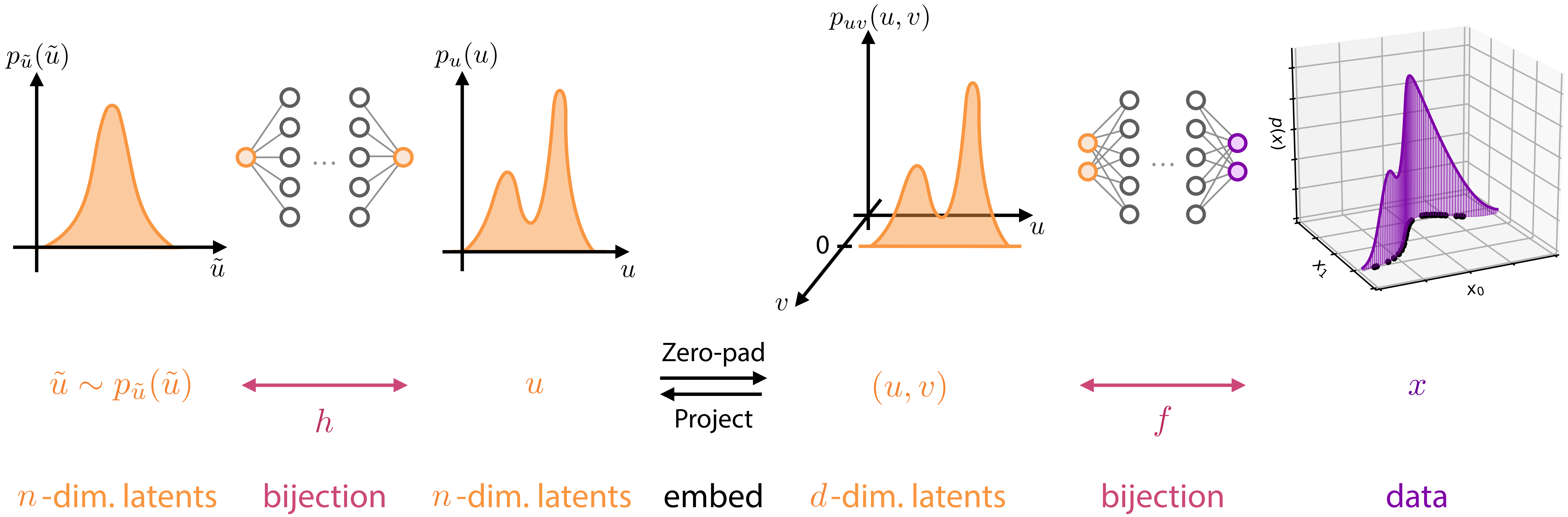
$\mathcal{M}$-flow:



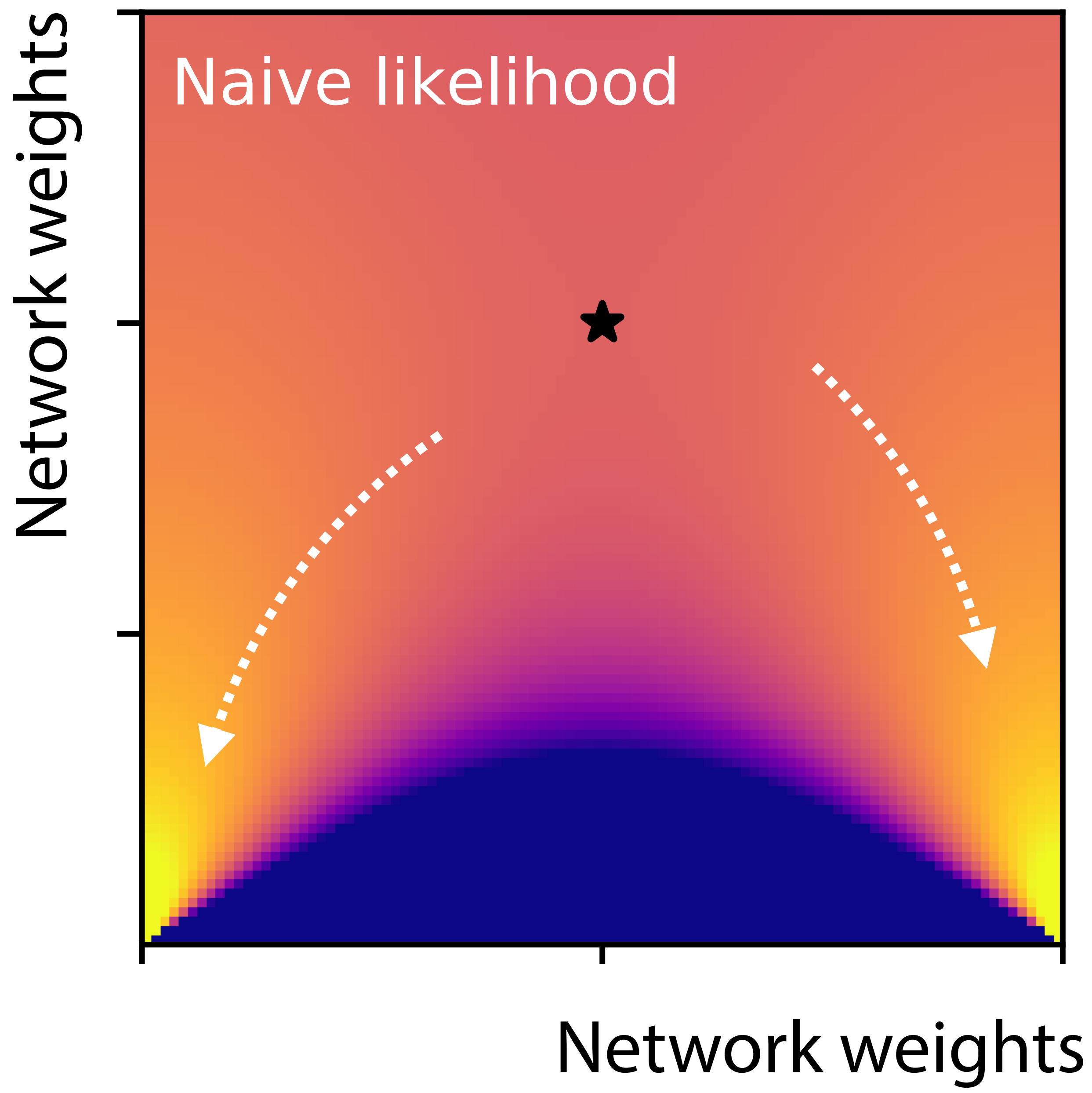

$p_{\tilde{u}}(\tilde{u})$
$\tilde{u}$
$p_u(u)$
u
$\tilde{u} \sim p_{\tilde{u}}(\tilde{u})$
h
u
n-dim. latents
bijection
n-dim. latents



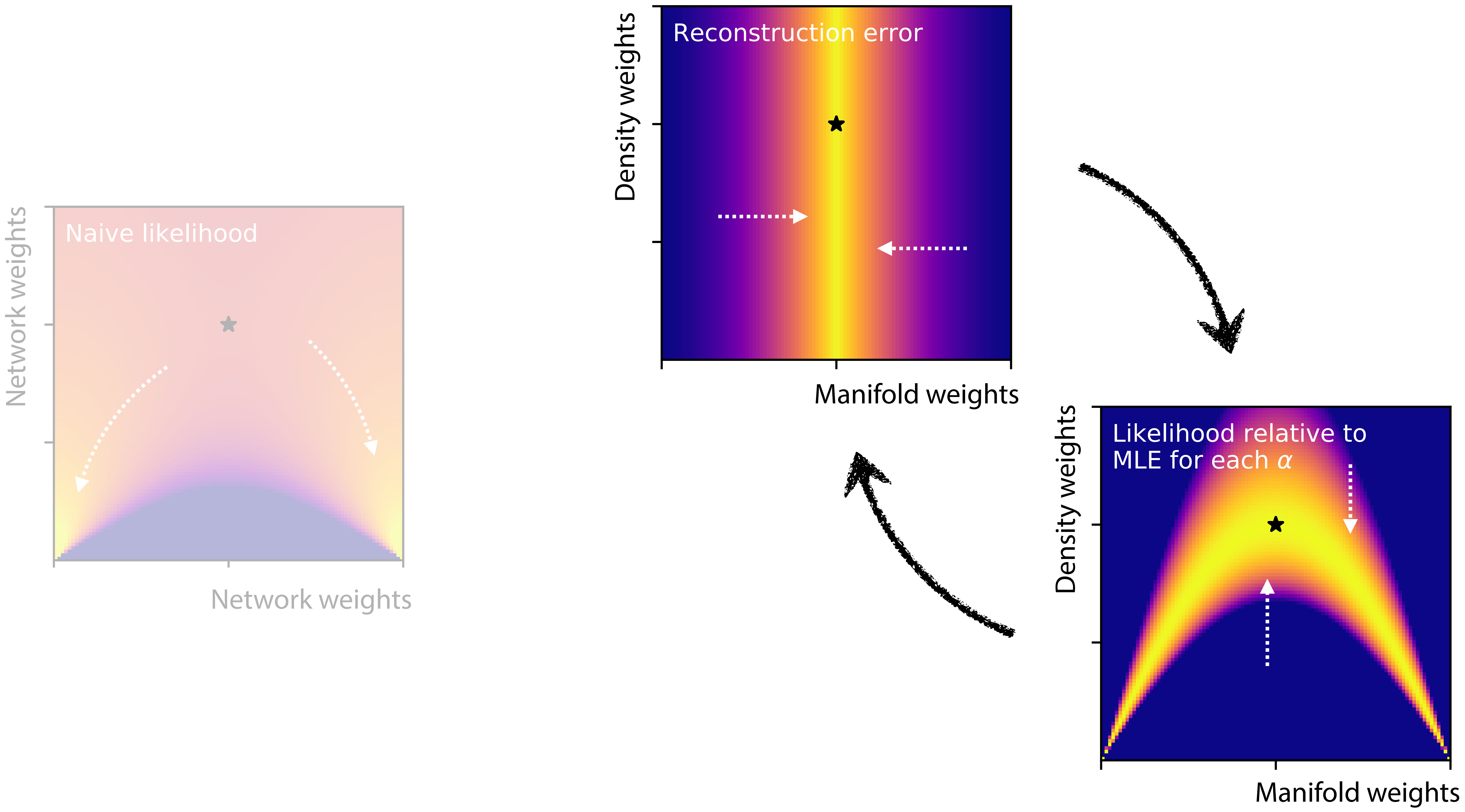
$p_{\tilde{u}}(\tilde{u})$
$\tilde{u}$
$p_u(u)$
u
$p_{uv}(u, v)$
u
0
v
$\tilde{u} \sim p_{\tilde{u}}(\tilde{u})$
h
u
Zero-pad
Project
(u, v)
n-dim. latents
bijection
n-dim. latents
embed
d-dim. latents



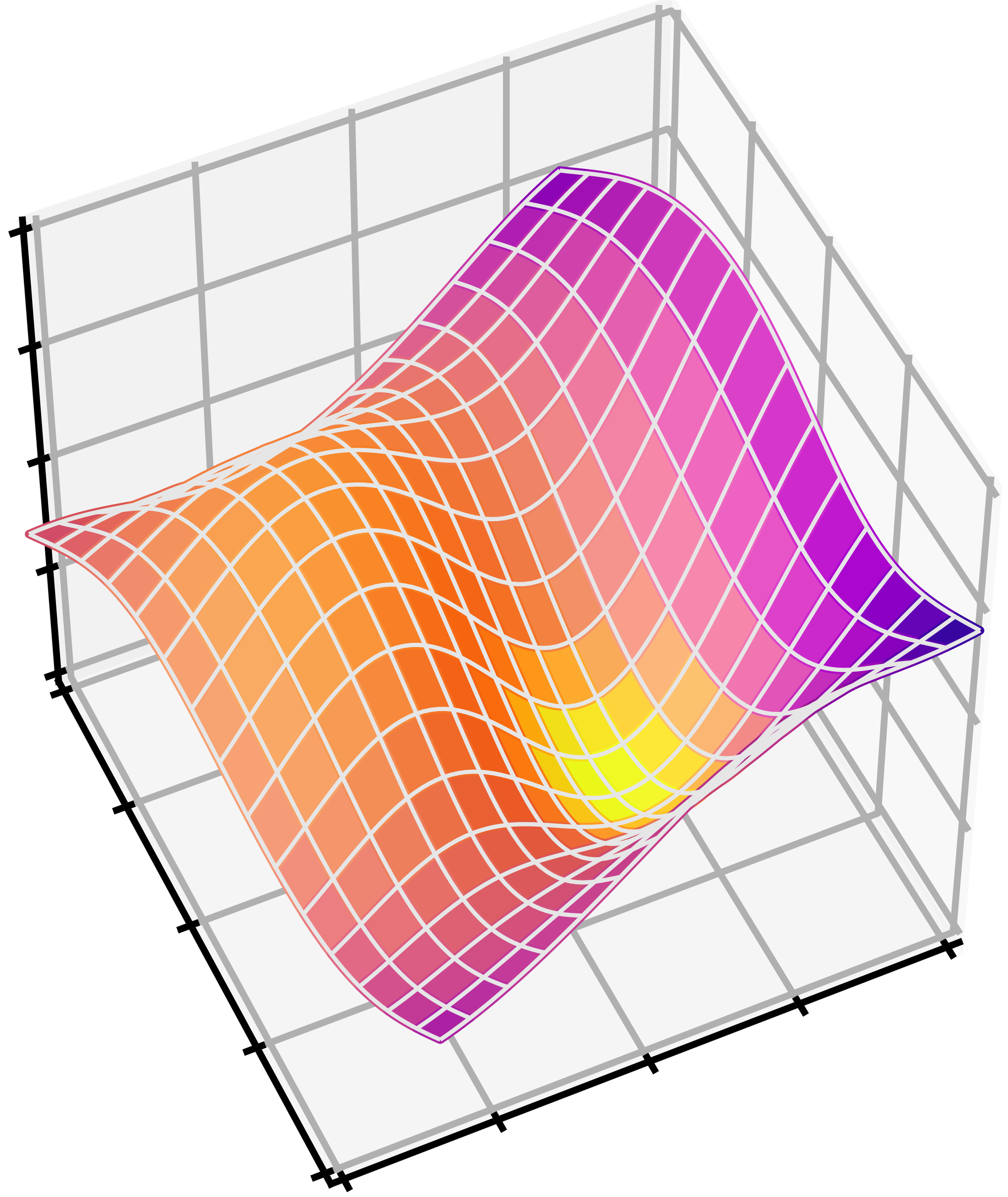
$p_{\tilde{u}}(\tilde{u})$
$\tilde{u}$
$p_u(u)$
u
$p_{uv}(u, v)$
u
0
v
p(x)
x_1
x_0
Zero-pad
Project
$\tilde{u} \sim p_{\tilde{u}}(\tilde{u})$
h
u
(u, v)
f
x
n-dim. latents
bijection
n-dim. latents
embed
d-dim. latents
bijection
data



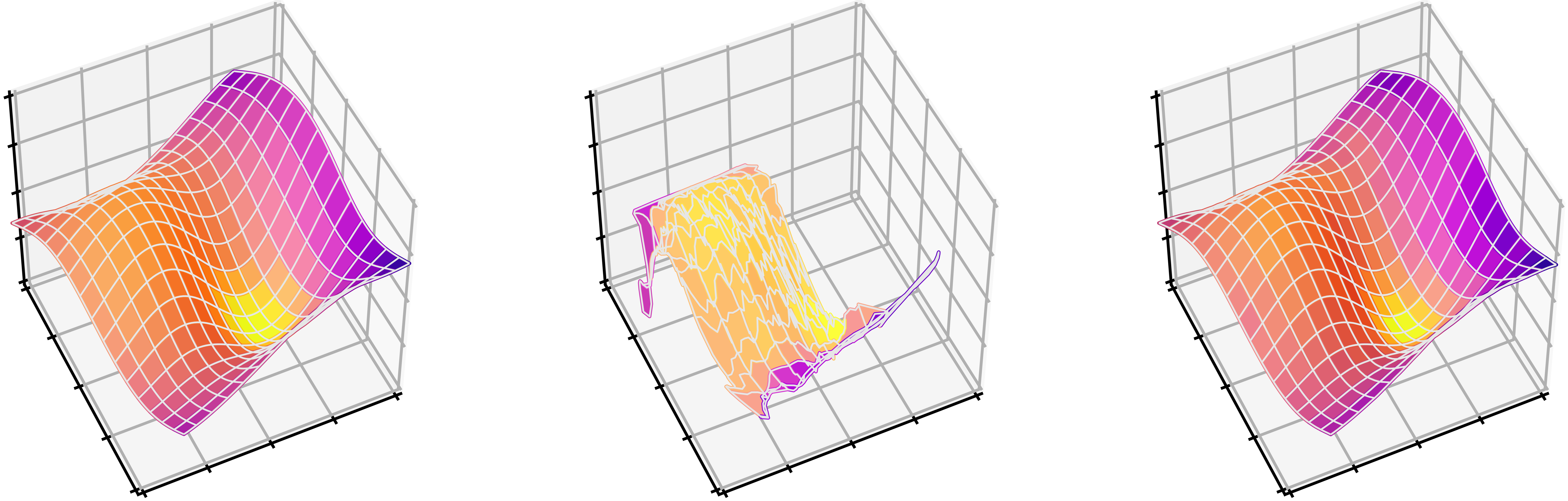
Naive likelihood
Network weights
Network weights

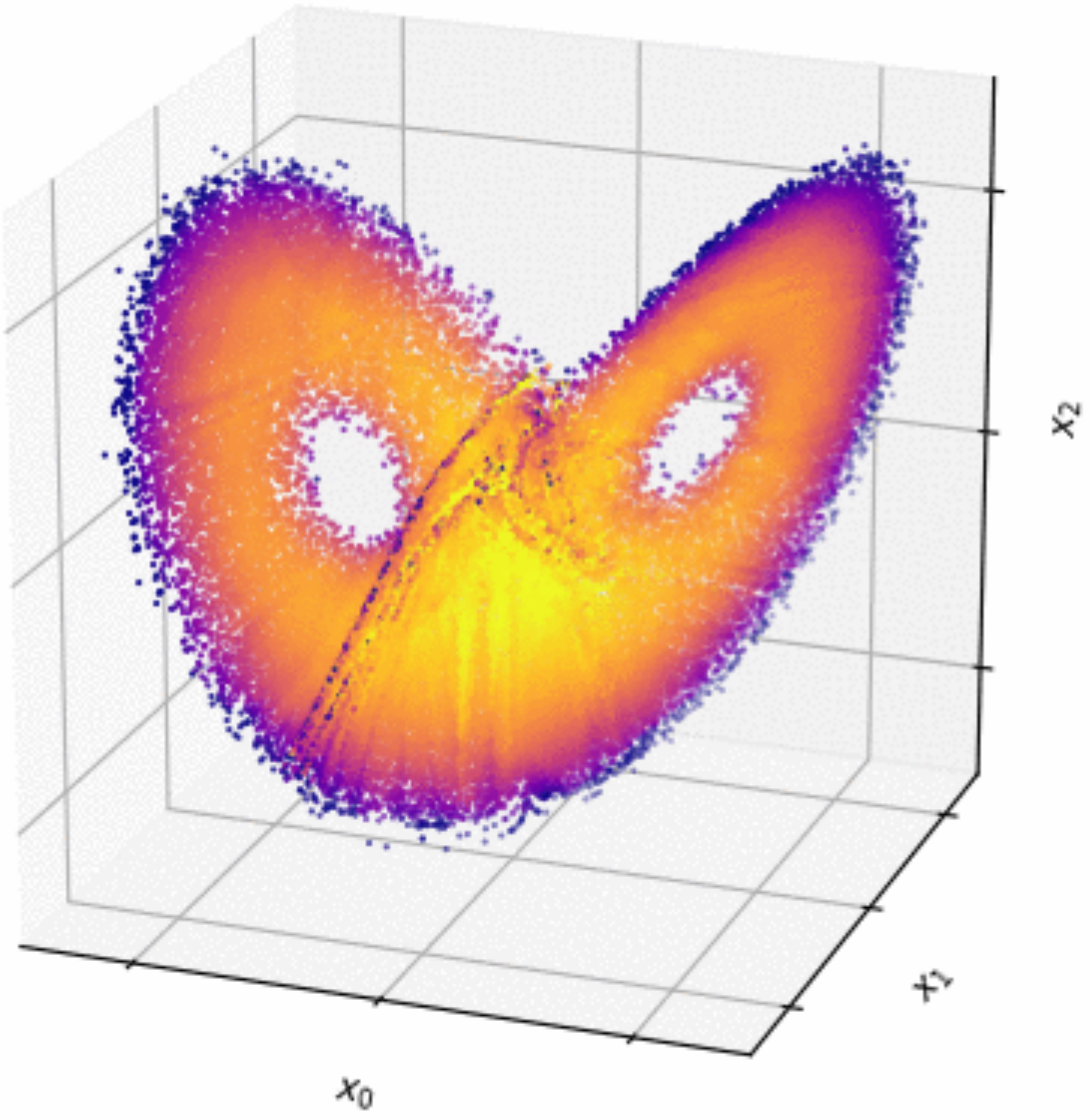

Naive likelihood
Network weights
Network weights
Reconstruction error
Density weights
Manifold weights
Likelihood relative to
MLE for each α
Density weights
Manifold weights

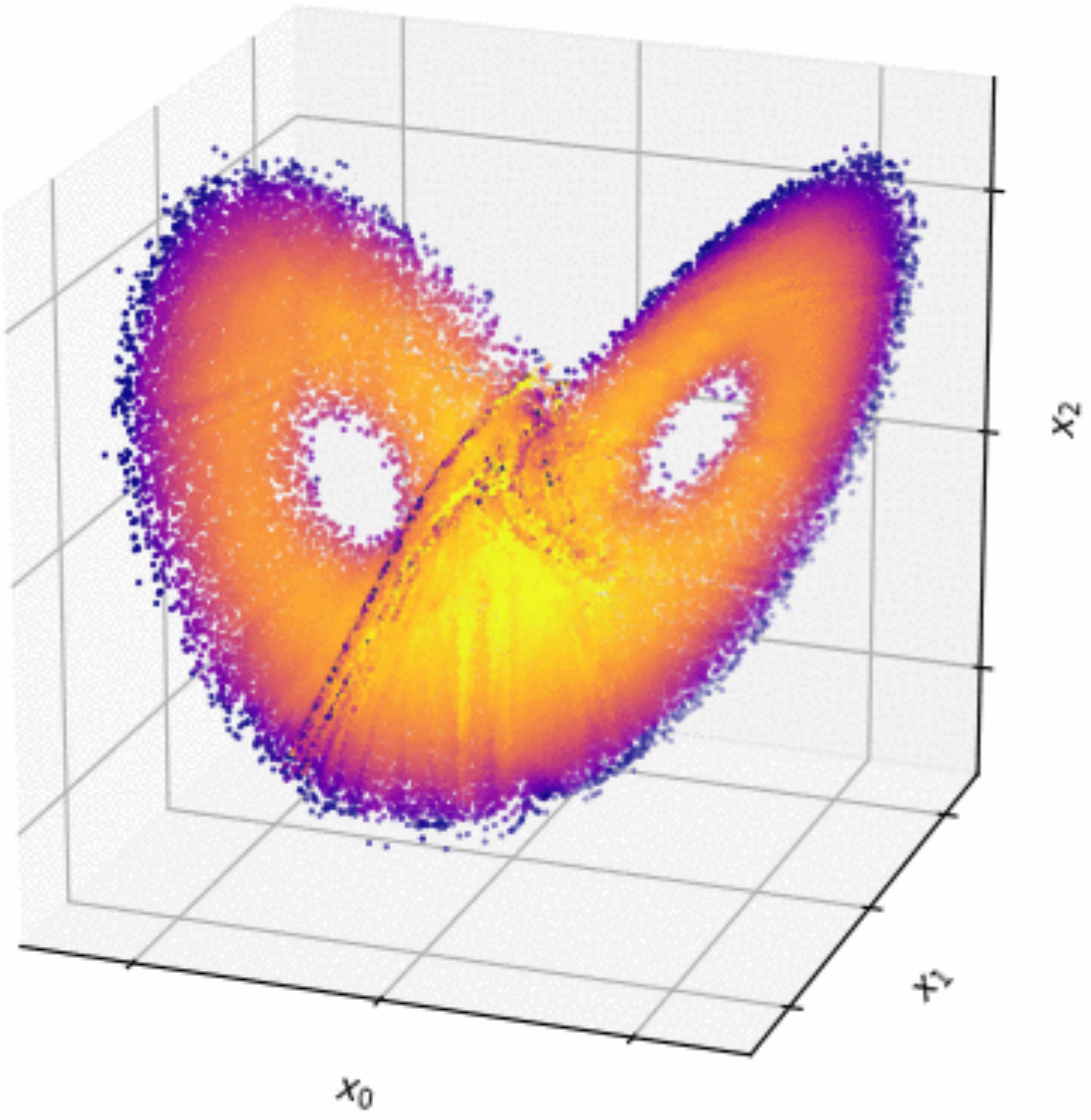
Ground truth, $\theta = 0$



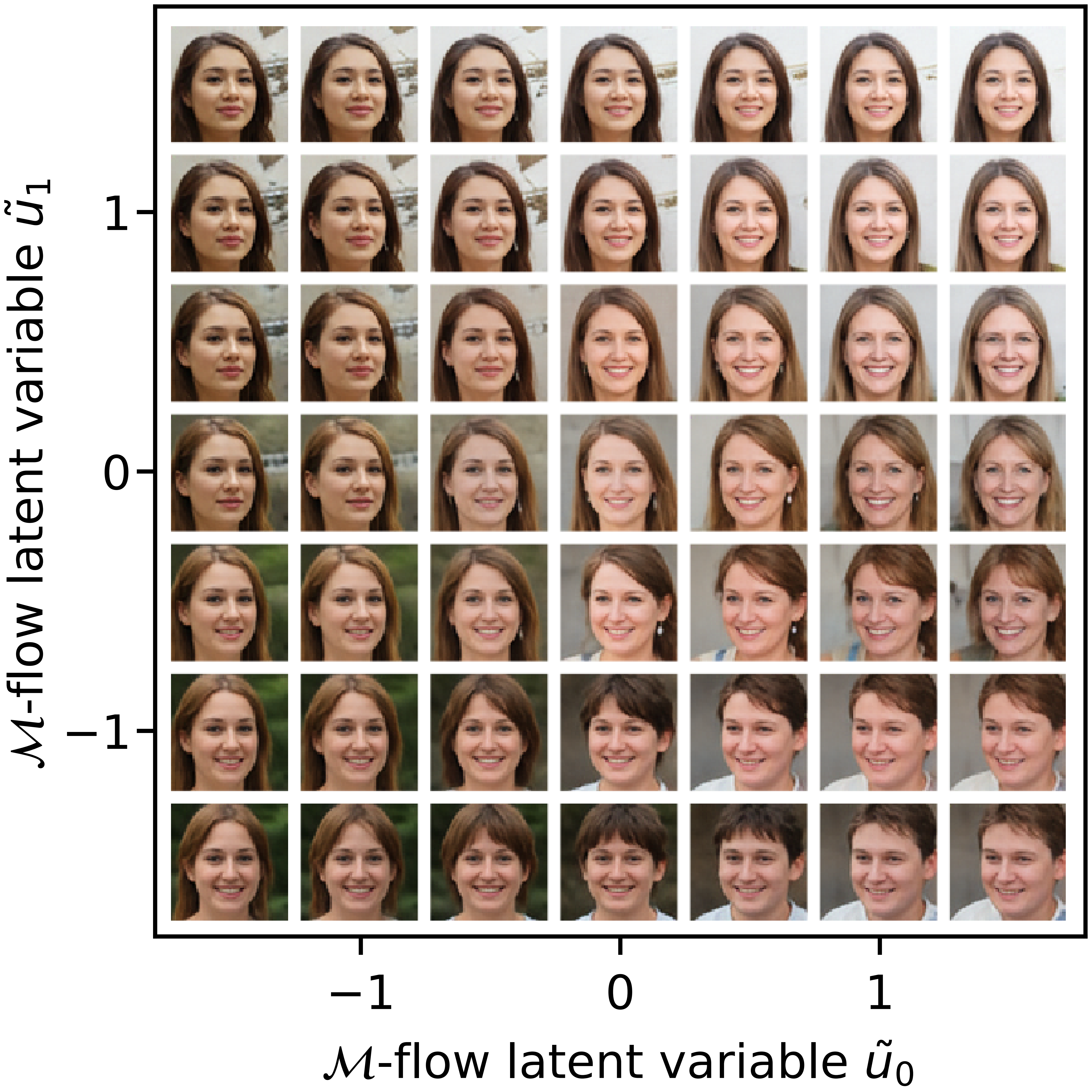
Ground truth, $\theta = 0$ PIE, $\theta = 0$ $\mathcal{M}$-flow (M/D), $\theta = 0$




x_0
x_1
x_2


x_0
x_1
x_2

[CMS collaboration]



1
0
−1
−1
0
1
$\mathcal{M}$-flow latent variable $\tilde{u}_1$
$\mathcal{M}$-flow latent variable $\tilde{u}_0$

Extended version on arXiv: 2003.13913

Flows for simultaneous manifold learning and density estimation

Johann Brehmer[a,b,1] **and Kyle Cranmer**[a,b]

[a]Center for Data Science, New York University, USA; [b]Center for Cosmology and Particle Physics, New York University, USA

June 16, 2020

We introduce manifold-learning flows ($\mathcal{M}$-flows), a new class of generative models that simultaneously learn the data manifold as well as a tractable probability density on that manifold. Combining aspects of normalizing flows, GANs, autoencoders, and energy-based models, they have the potential to represent datasets with a manifold structure more faithfully and provide handles on dimensionality reduction, denoising, and out-of-distribution detection. We argue why such models should not be trained by maximum likelihood alone and present a new training algorithm that separates manifold and density updates. In a range of experiments we demonstrate how $\mathcal{M}$-flows learn the data manifold and allow for better inference than standard flows in the ambient data space.

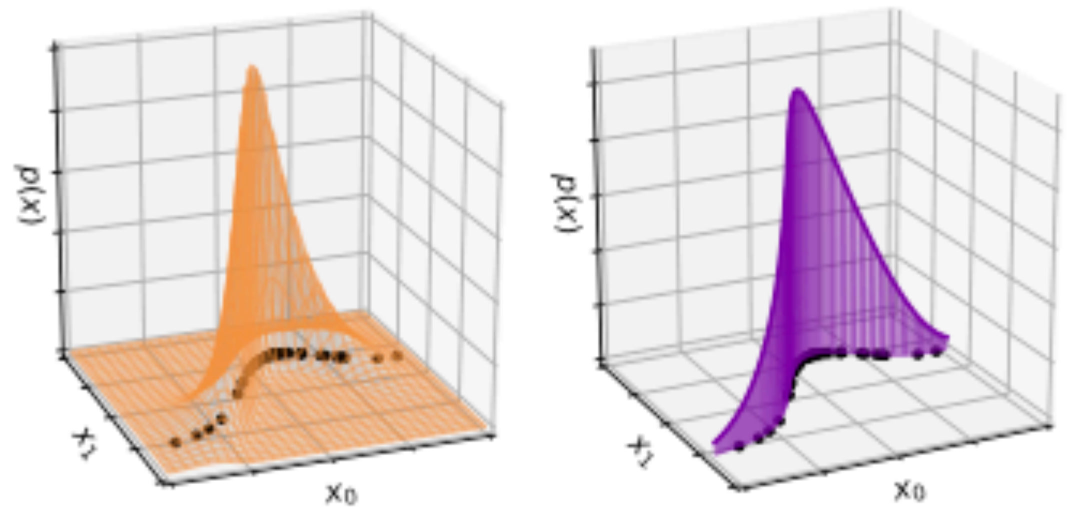


Fig. 1. Sketch of how a standard normalizing flow in the ambient data space (left, orange surface) and an $\mathcal{M}$-flow (right, purple) model data (black dots).

Contents

1. Introduction

Inferring a probability distribution from example data is a common problem that is increasingly tackled with deep generative models. Generative adversarial networks (GANs) (1) and variational autoencoders (VAEs) (2) are both based on a lower-dimensional latent space and a learnable mapping from