

# Flows for simultaneous manifold learning and density estimation

#notagan

Johann Brehmer

Work done at New York University

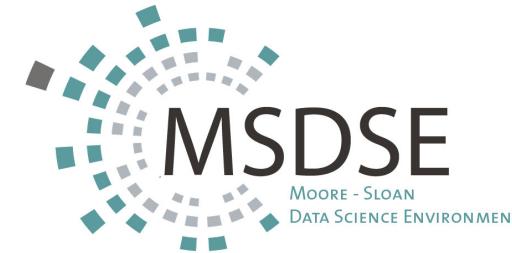
AI and Physics seminar at CMU  
April 28, 2021

# Flows for simultaneous manifold learning and density estimation

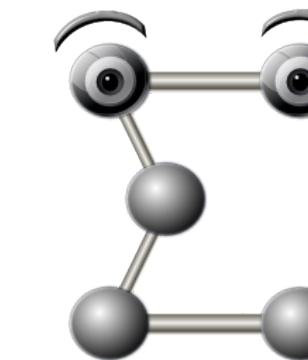
#notagan

Johann Brehmer

Work done at New York University



The SCAILFIN Project  
[scailfin.github.io](https://scailfin.github.io)

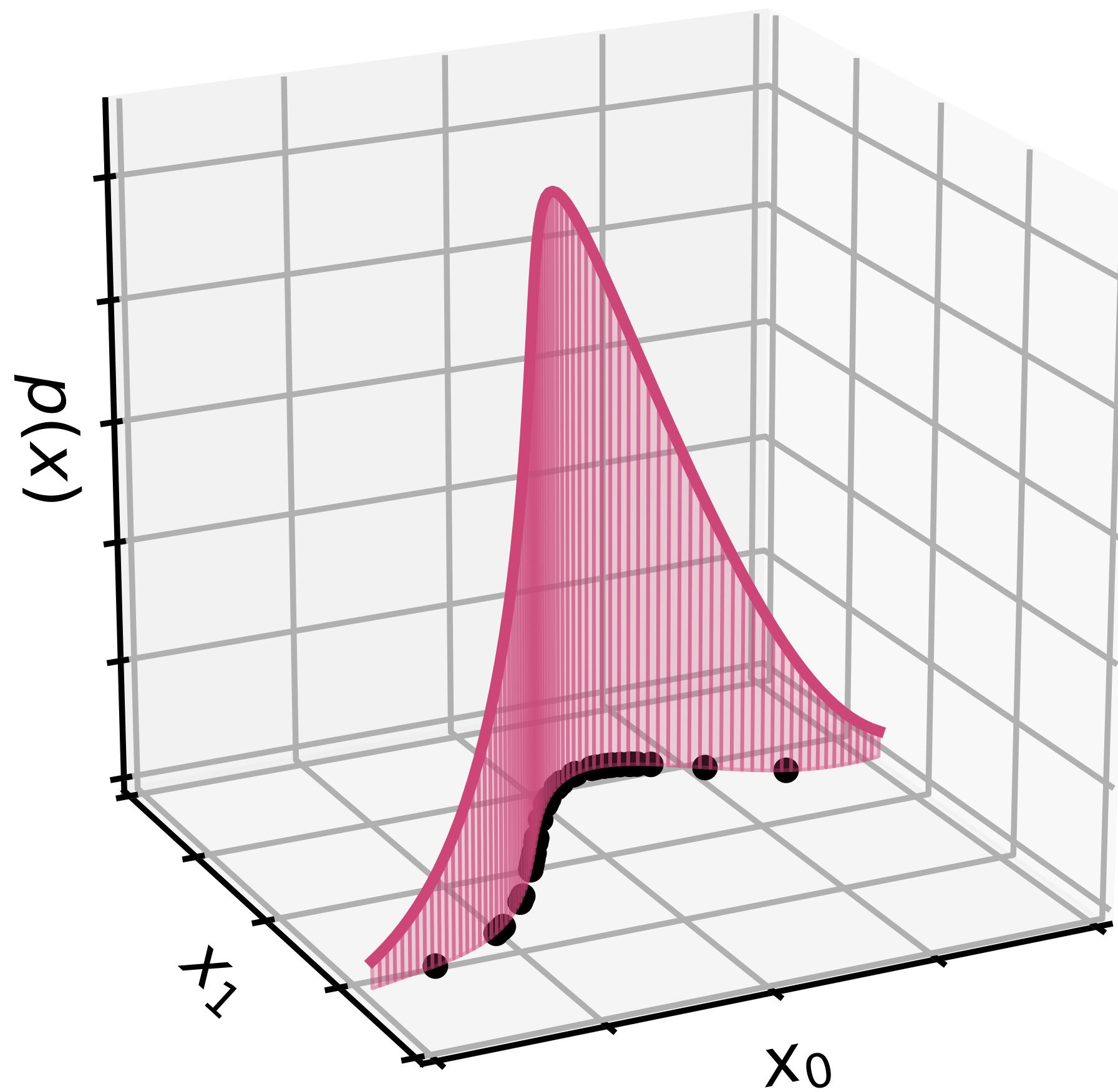


**BROOKHAVEN**  
NATIONAL LABORATORY | Scientific Data and Computing Center

## $\mathcal{M}$ -flow:

A normalizing flow that

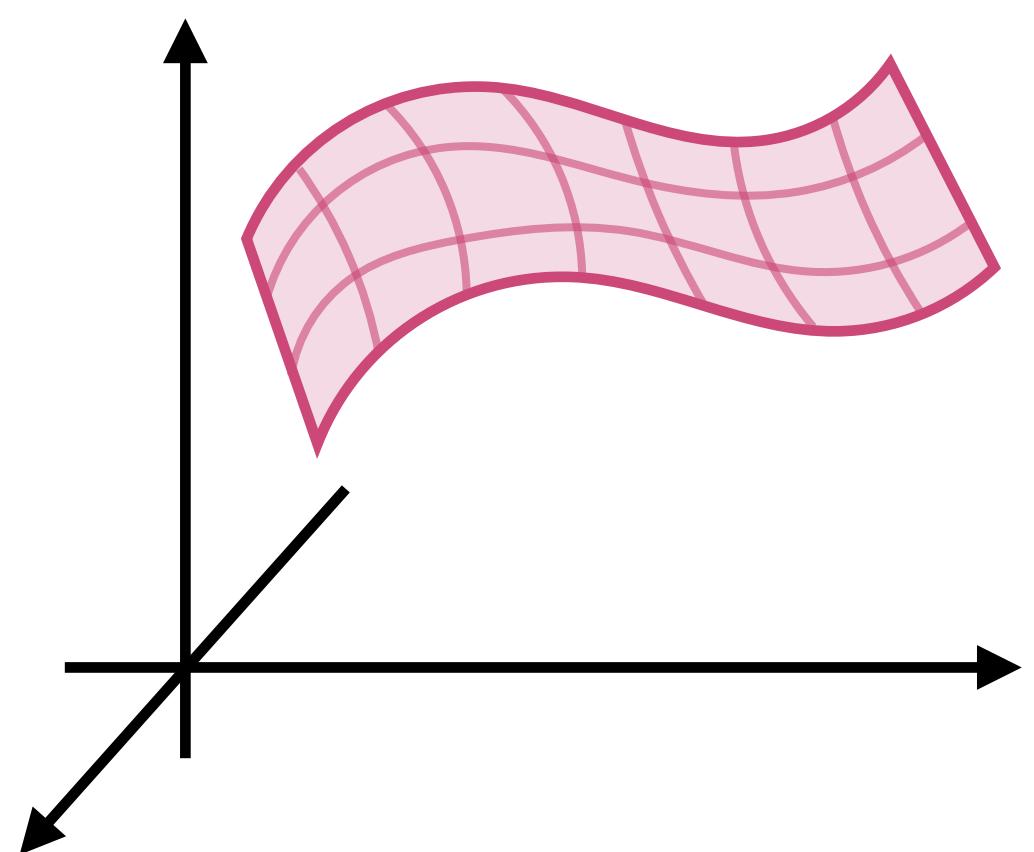
- describes data as a probability density over a lower-dimensional manifold
- learns manifold and density from data
- has a tractable density



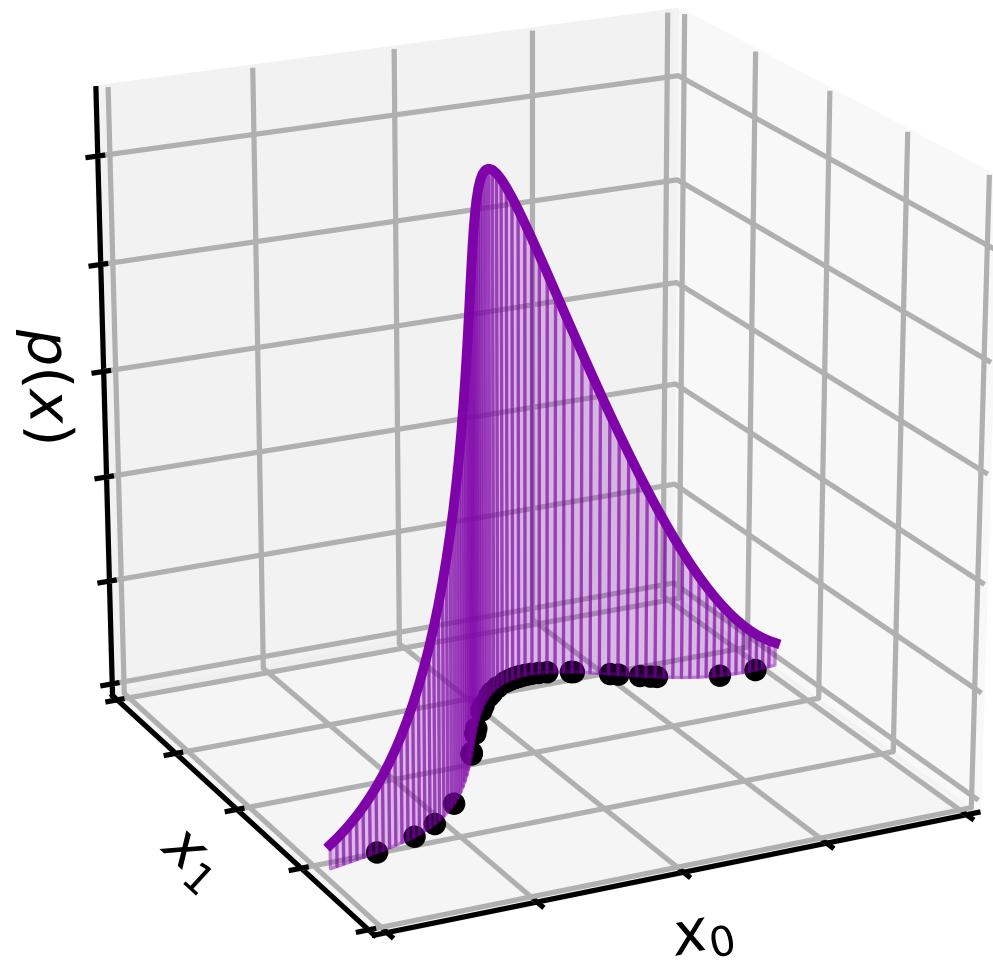
m-flo:



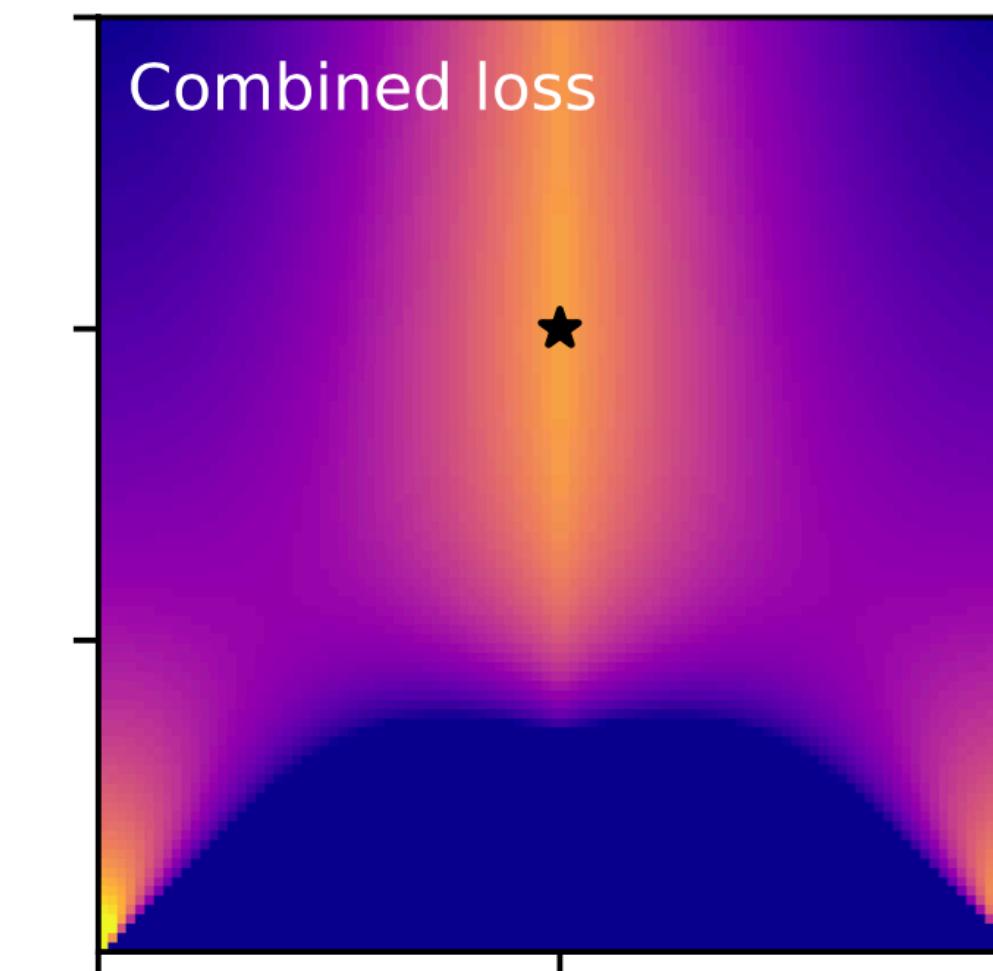
[m-flo / Rhythm Zone]



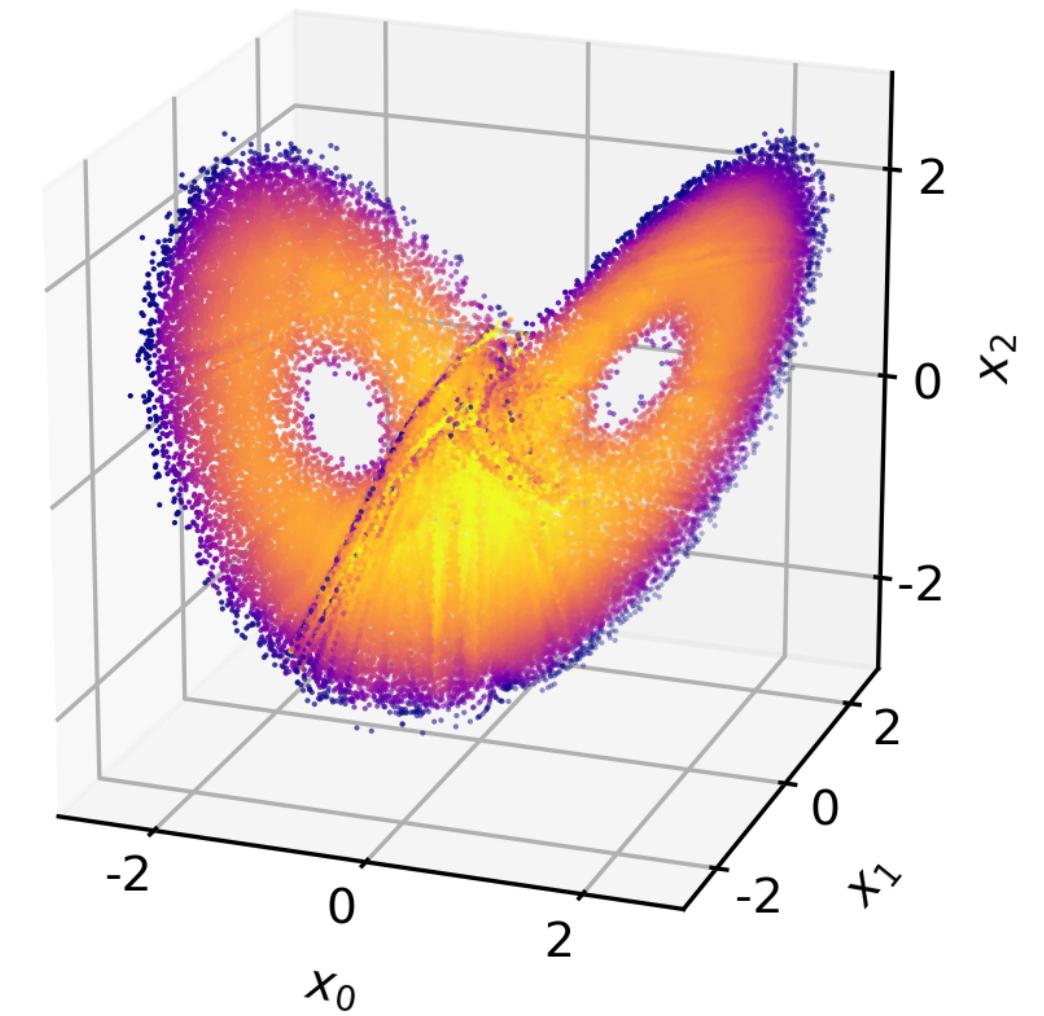
1. Generative models  
and the data manifold



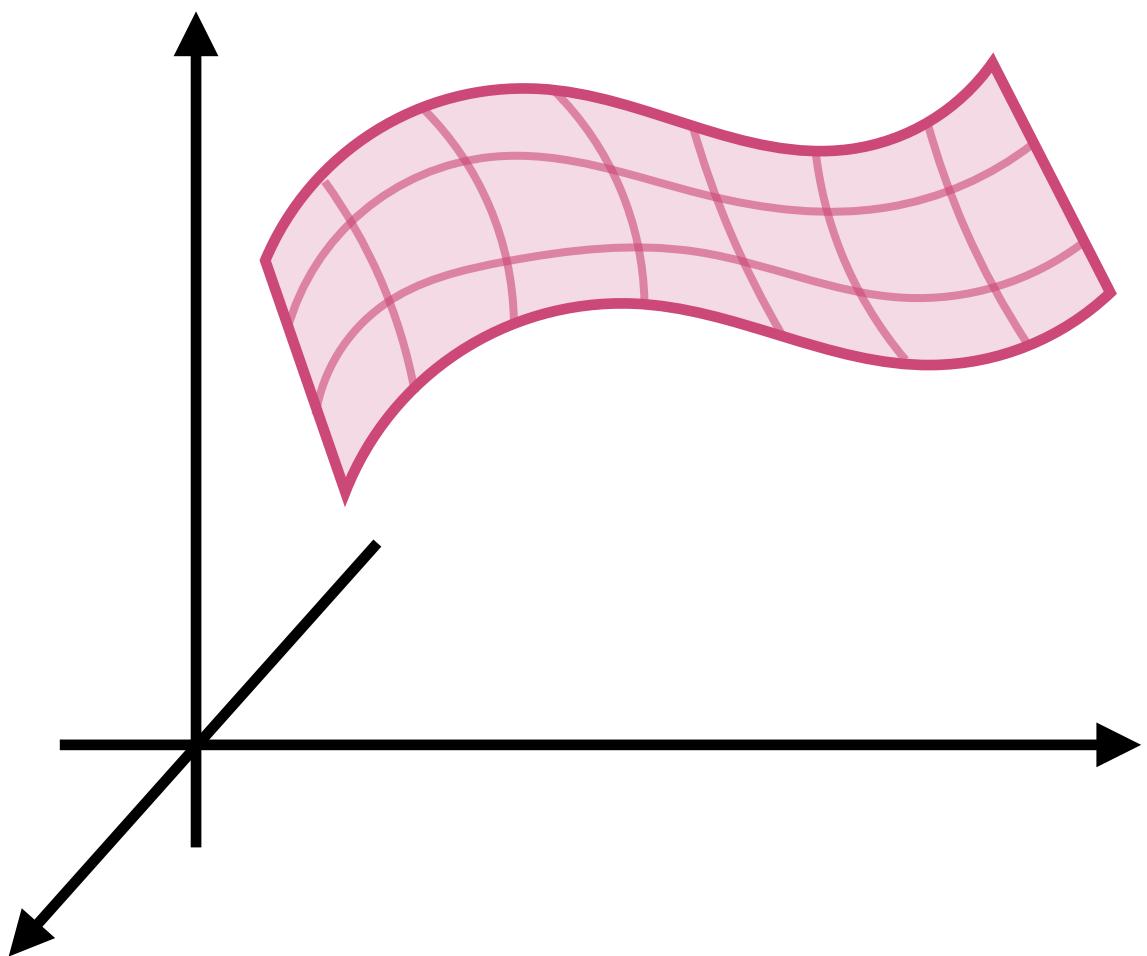
2.  $\mathcal{M}$ -flows



3. Training  $\mathcal{M}$ -flows



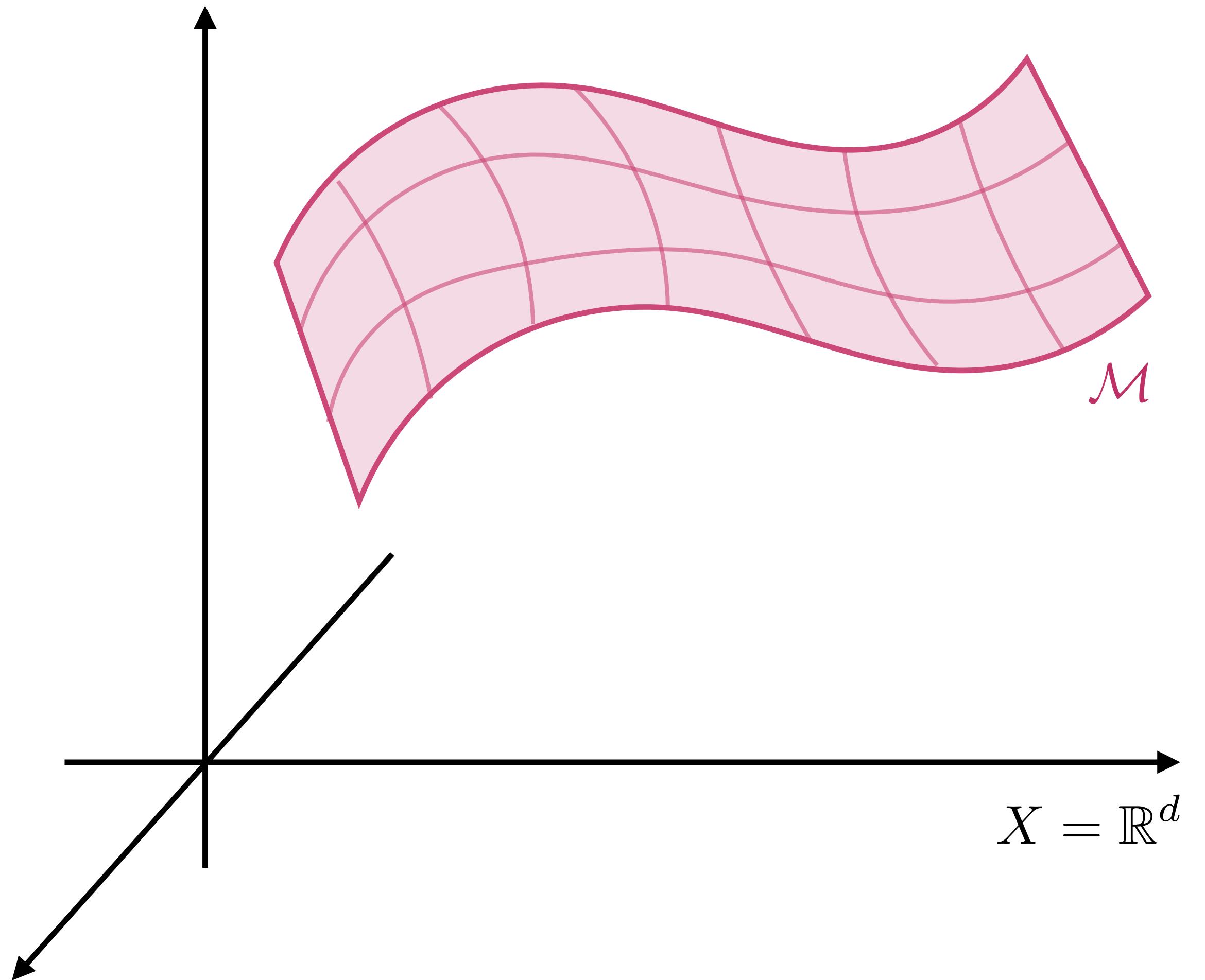
4. Experiments



Generative models  
and the data manifold

# The manifold hypothesis

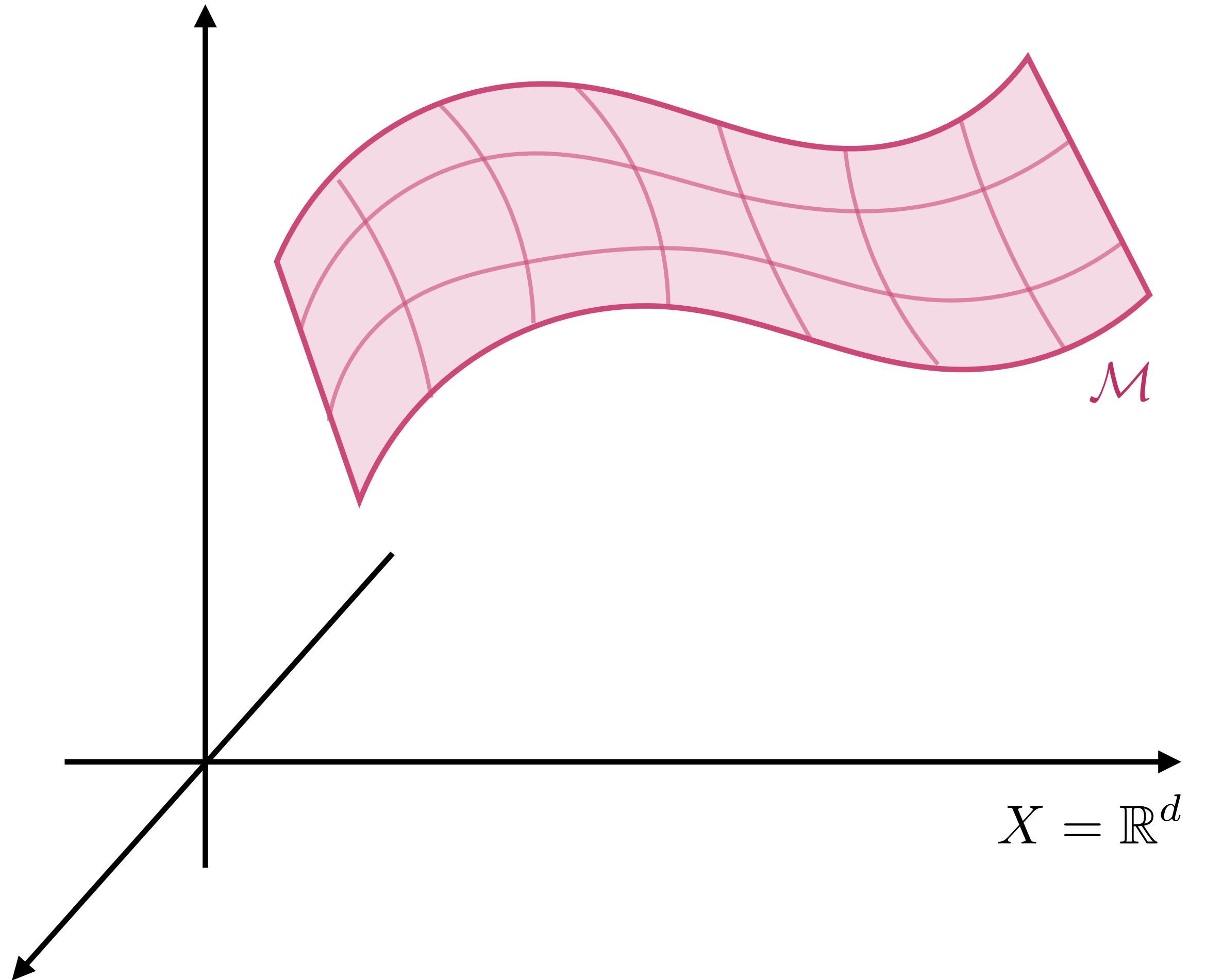
Data often live on a  $n$ -dimensional manifold  
embedded in the  $d$ -dimensional ambient space



# The manifold hypothesis

Data often live on a *n*-dimensional manifold embedded in the *d*-dimensional ambient space

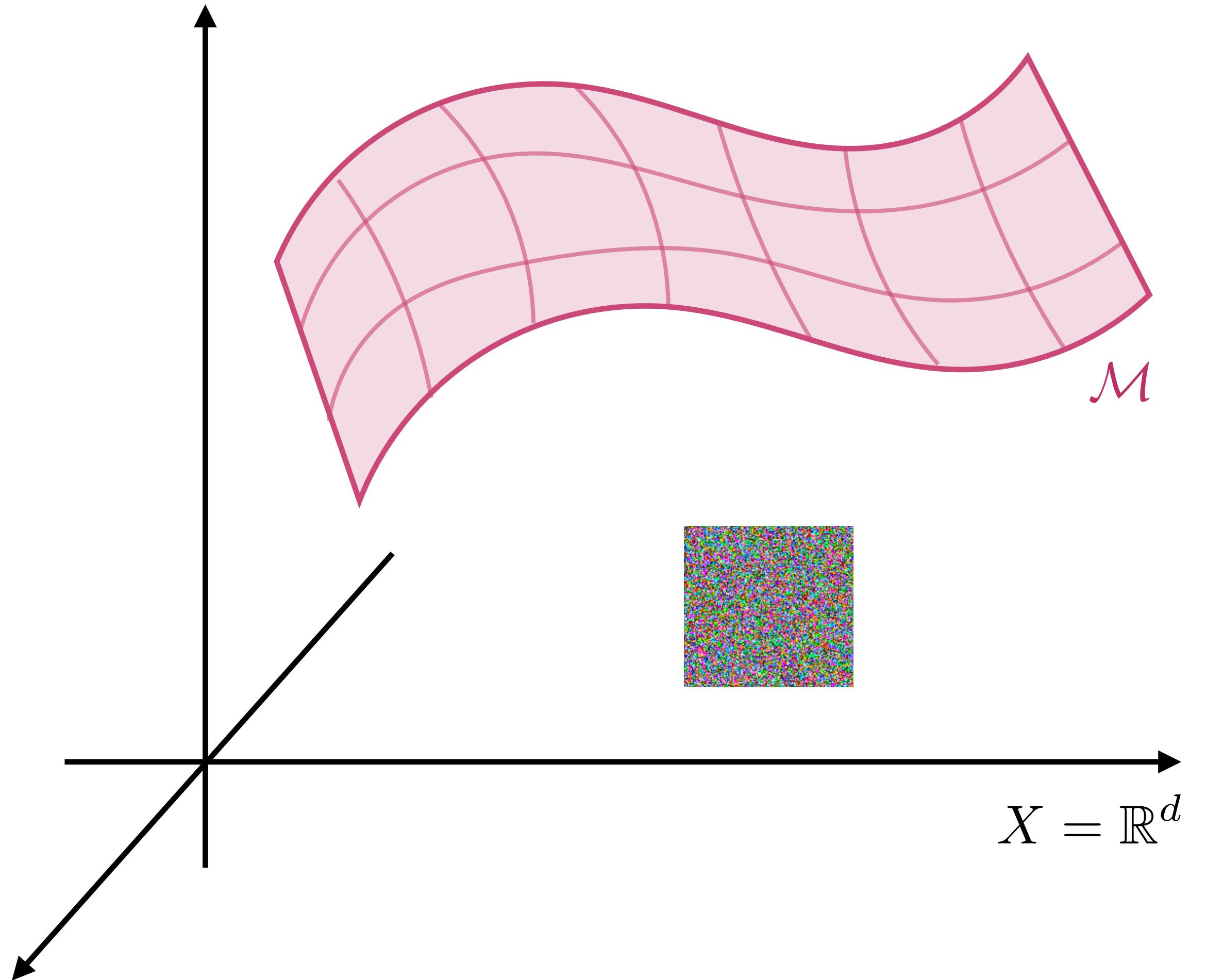
- Robot arms, molecules: limited degrees of freedom
- Particle physics: energy-momentum conservation, on-shell conditions, redundant observables
- Many other high-dimensional datasets (e.g. images): empirical evidence for (approximate) data manifold  
[L. Cayton 2005; ...]



# The manifold hypothesis

Data often live on a *n*-dimensional manifold embedded in the *d*-dimensional ambient space

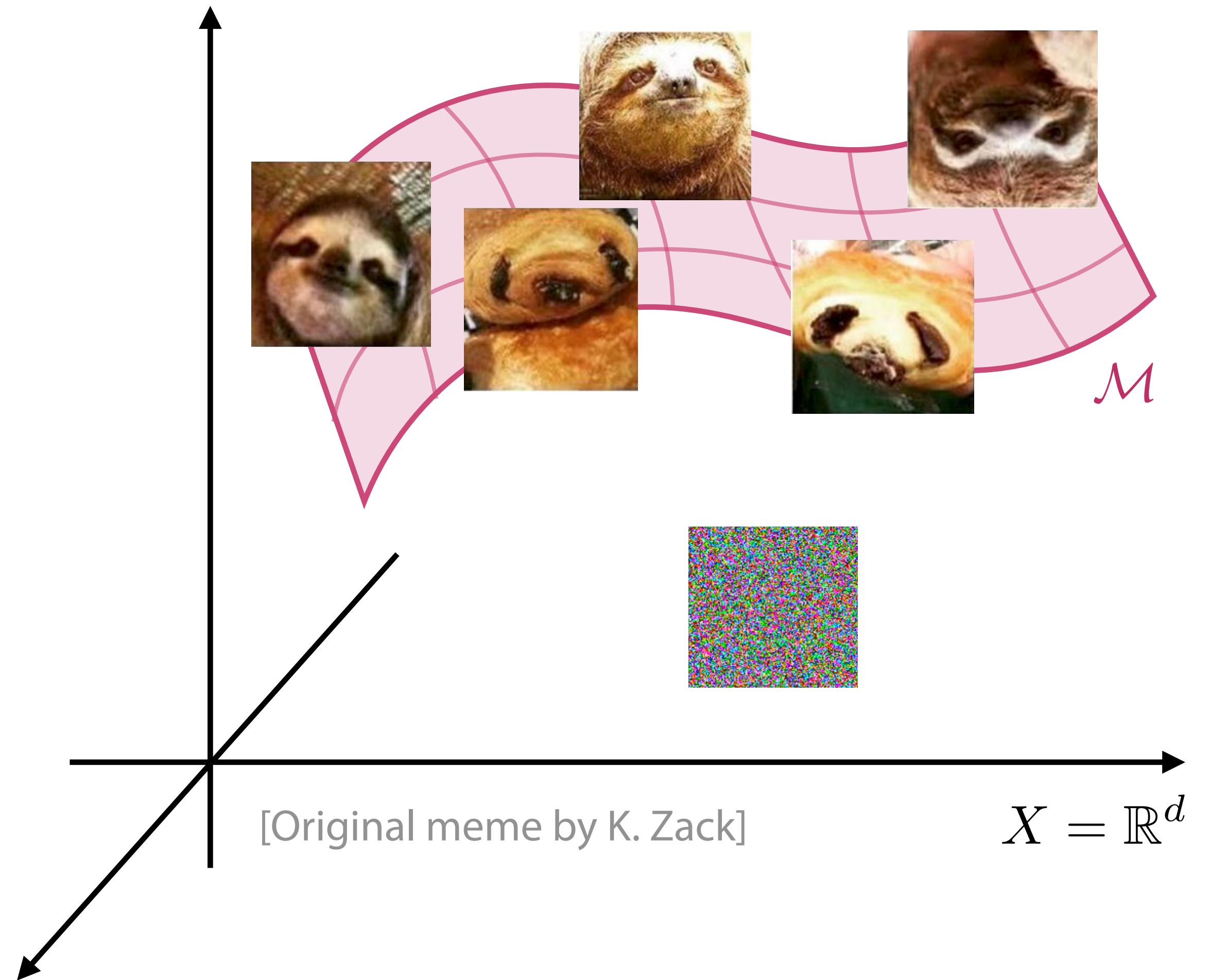
- Robot arms, molecules: limited degrees of freedom
- Particle physics: energy-momentum conservation, on-shell conditions, redundant observables
- Many other high-dimensional datasets (e.g. images): empirical evidence for (approximate) data manifold  
[L. Cayton 2005; ...]



# The manifold hypothesis

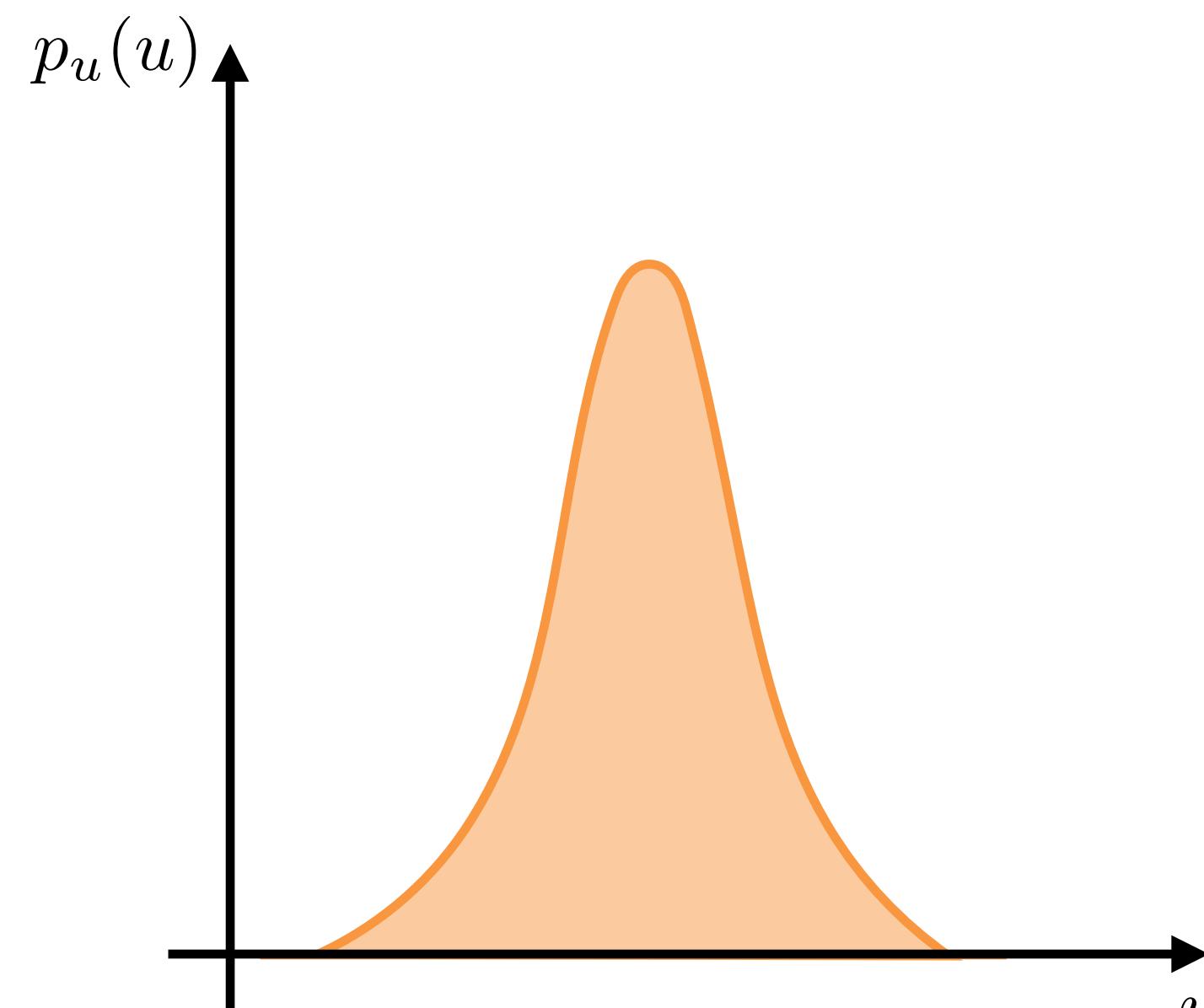
Data often live on a *n*-dimensional manifold embedded in the *d*-dimensional ambient space

- Robot arms, molecules: limited degrees of freedom
- Particle physics: energy-momentum conservation, on-shell conditions, redundant observables
- Many other high-dimensional datasets (e.g. images): empirical evidence for (approximate) data manifold [L. Cayton 2005; ...]



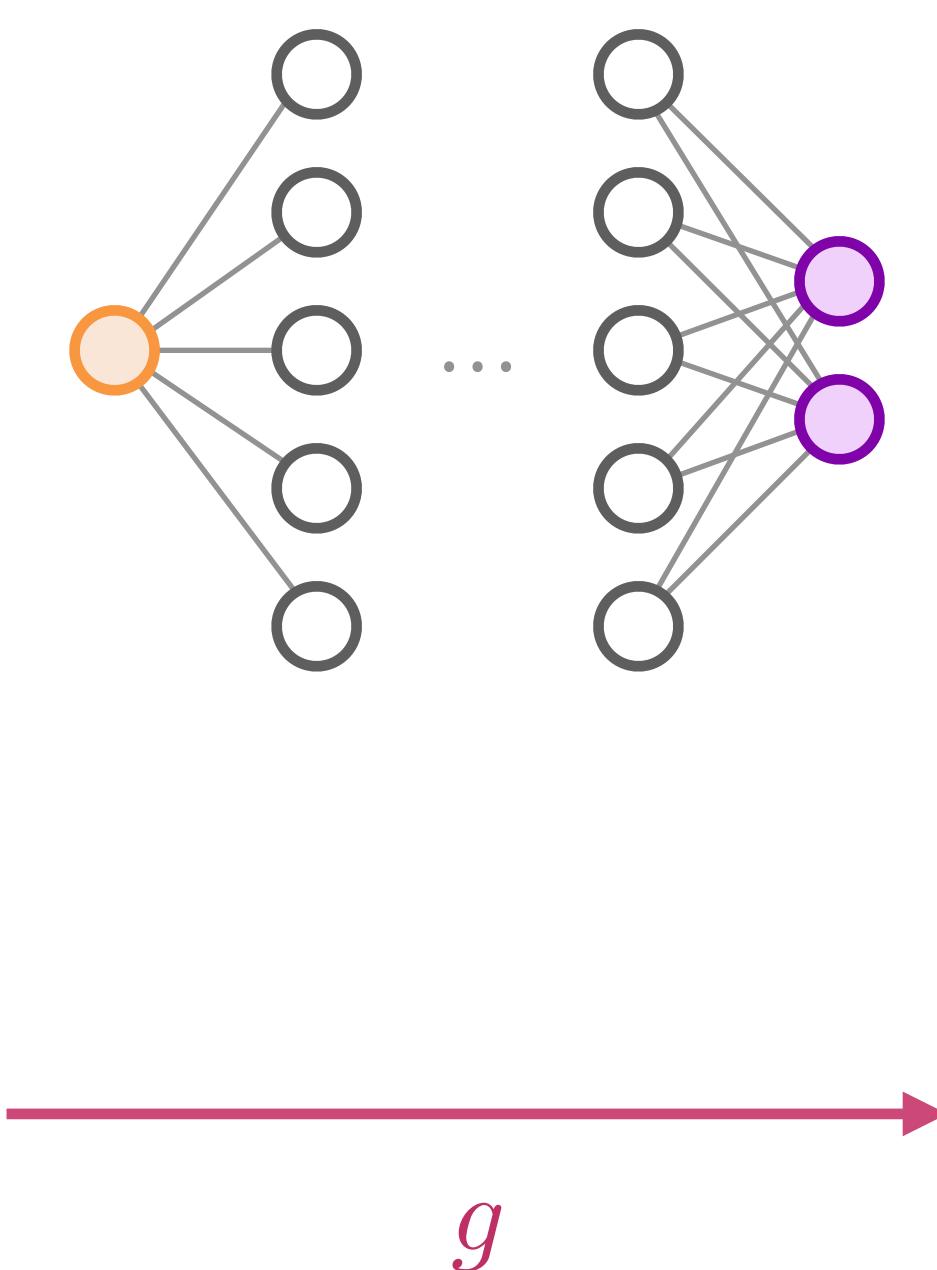
# Generative adversarial networks (GANs)

[I. Goodfellow et al 1406.2661]

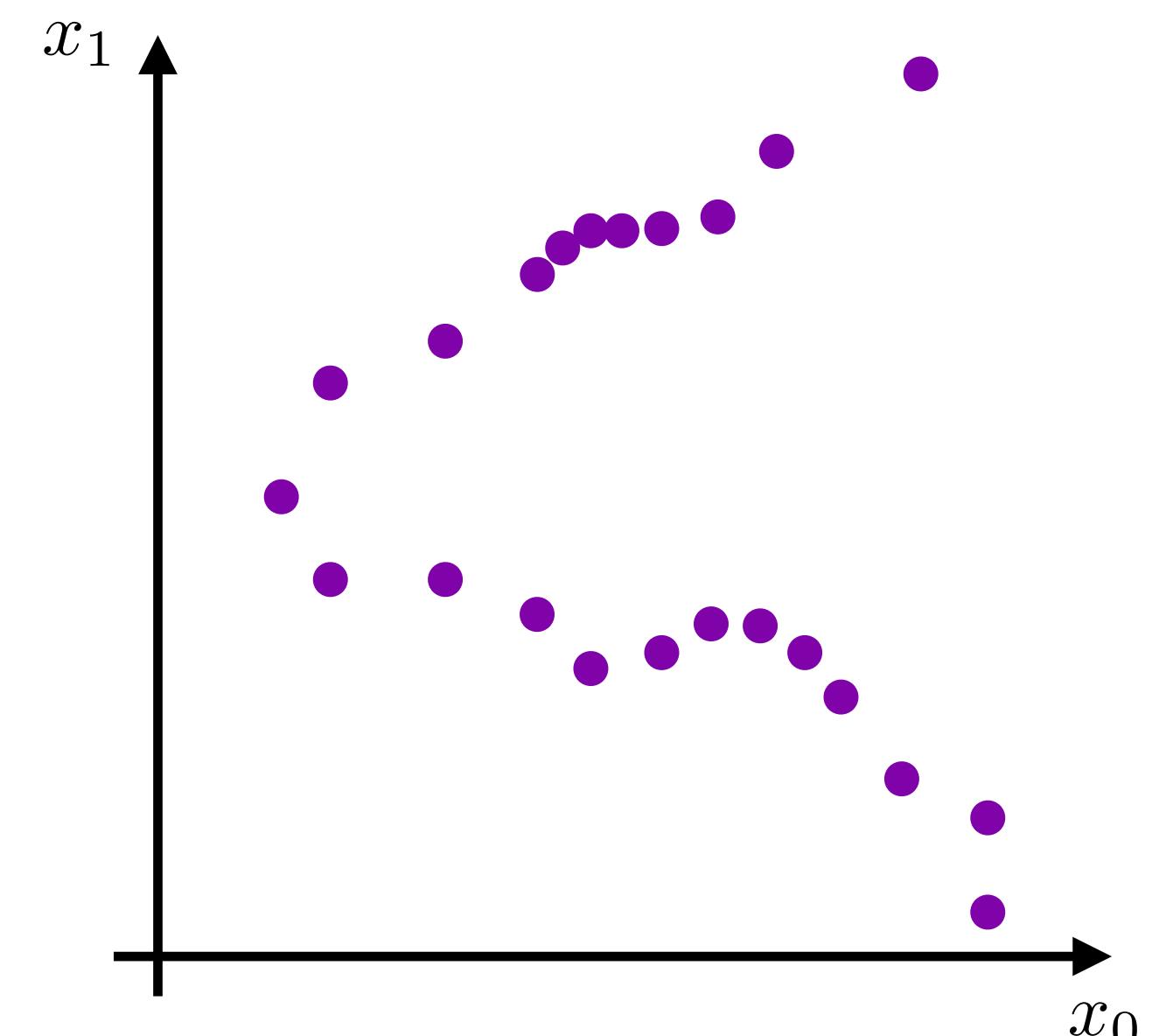


$$u \sim p_u(u)$$

$n$ -dim. latent variables



unconstrained NN



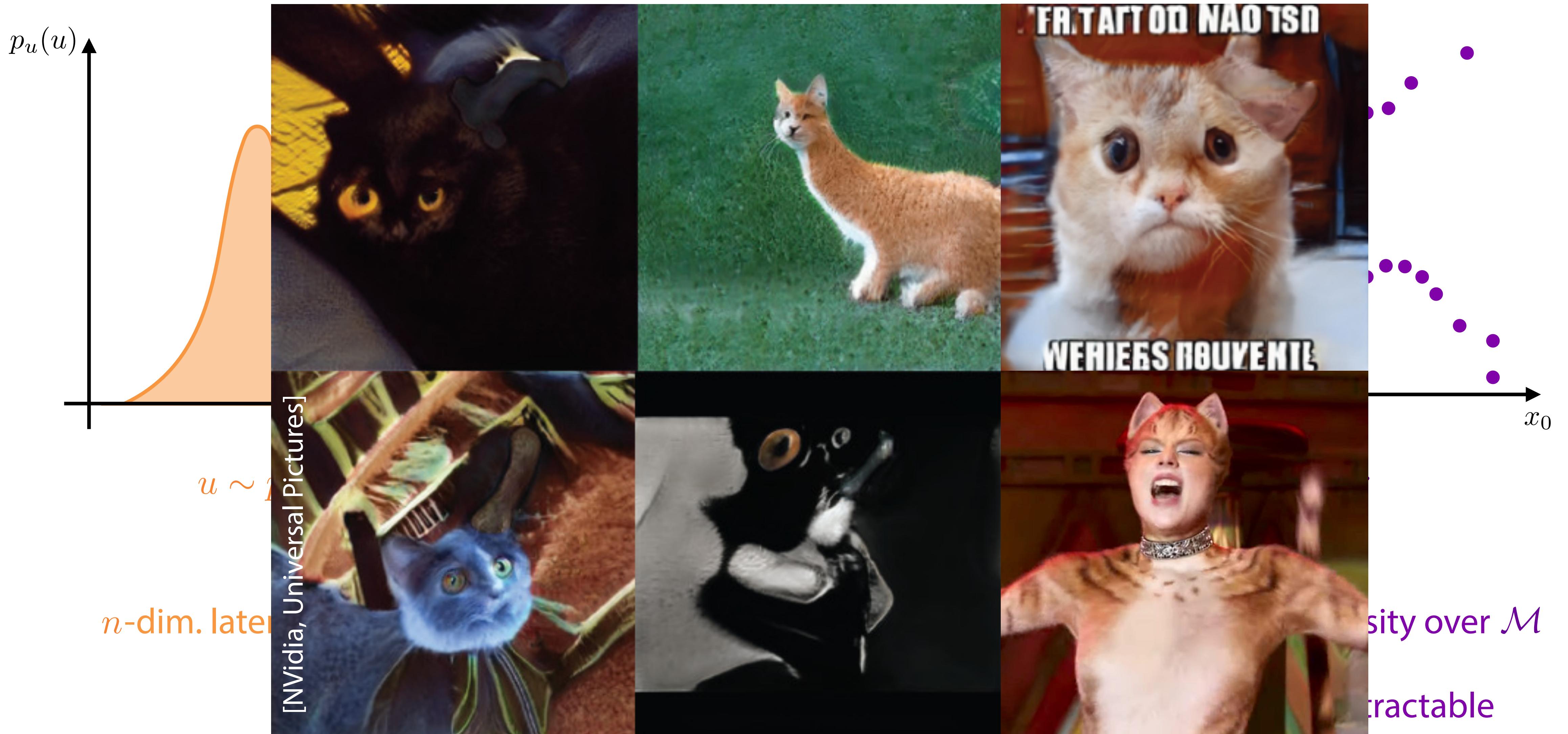
$x$

implicit density over  $\mathcal{M}$

$p_{\mathcal{M}}(x)$  intractable

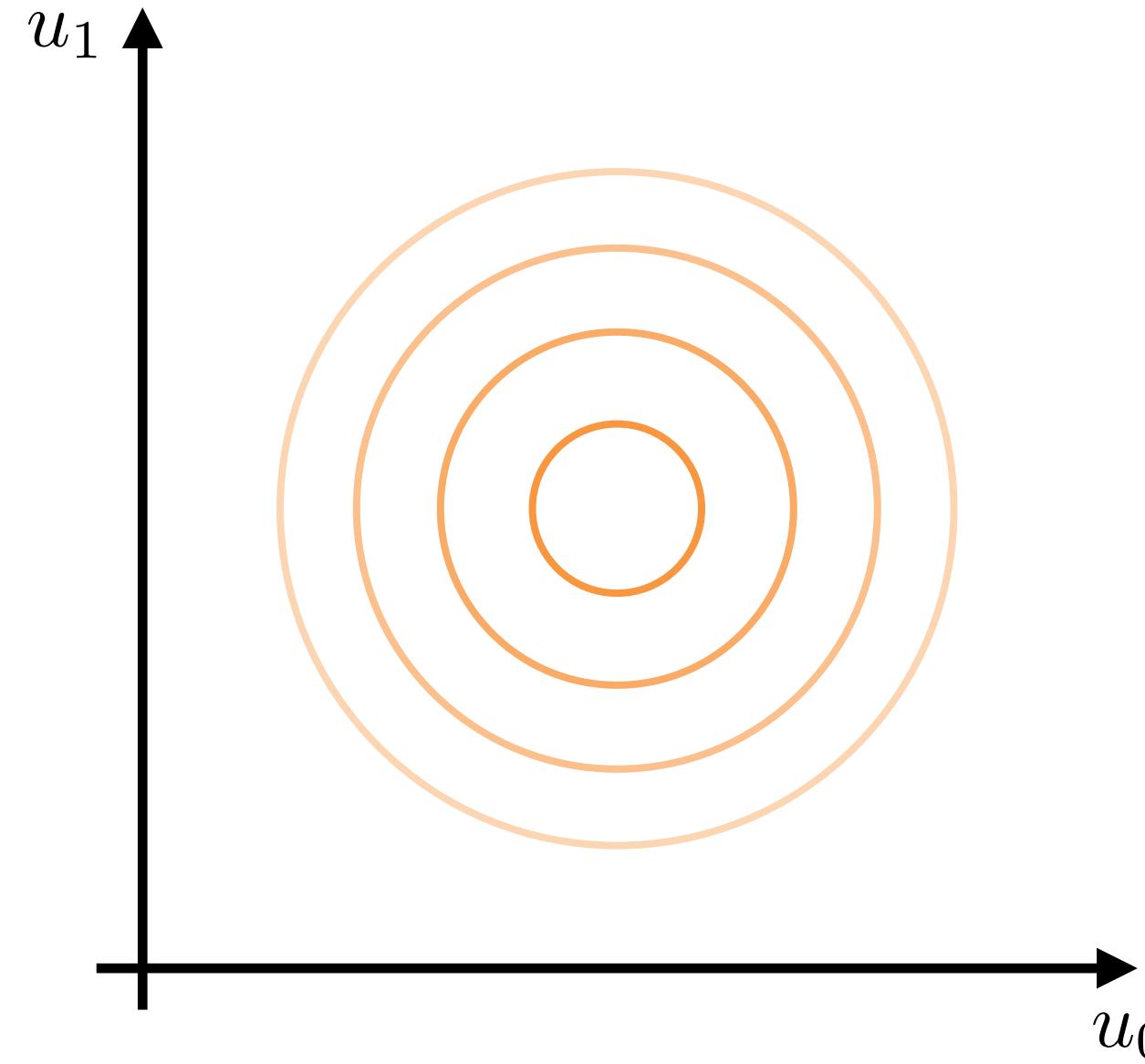
# Generative adversarial networks (GANs)

[I. Goodfellow et al 1406.2661]

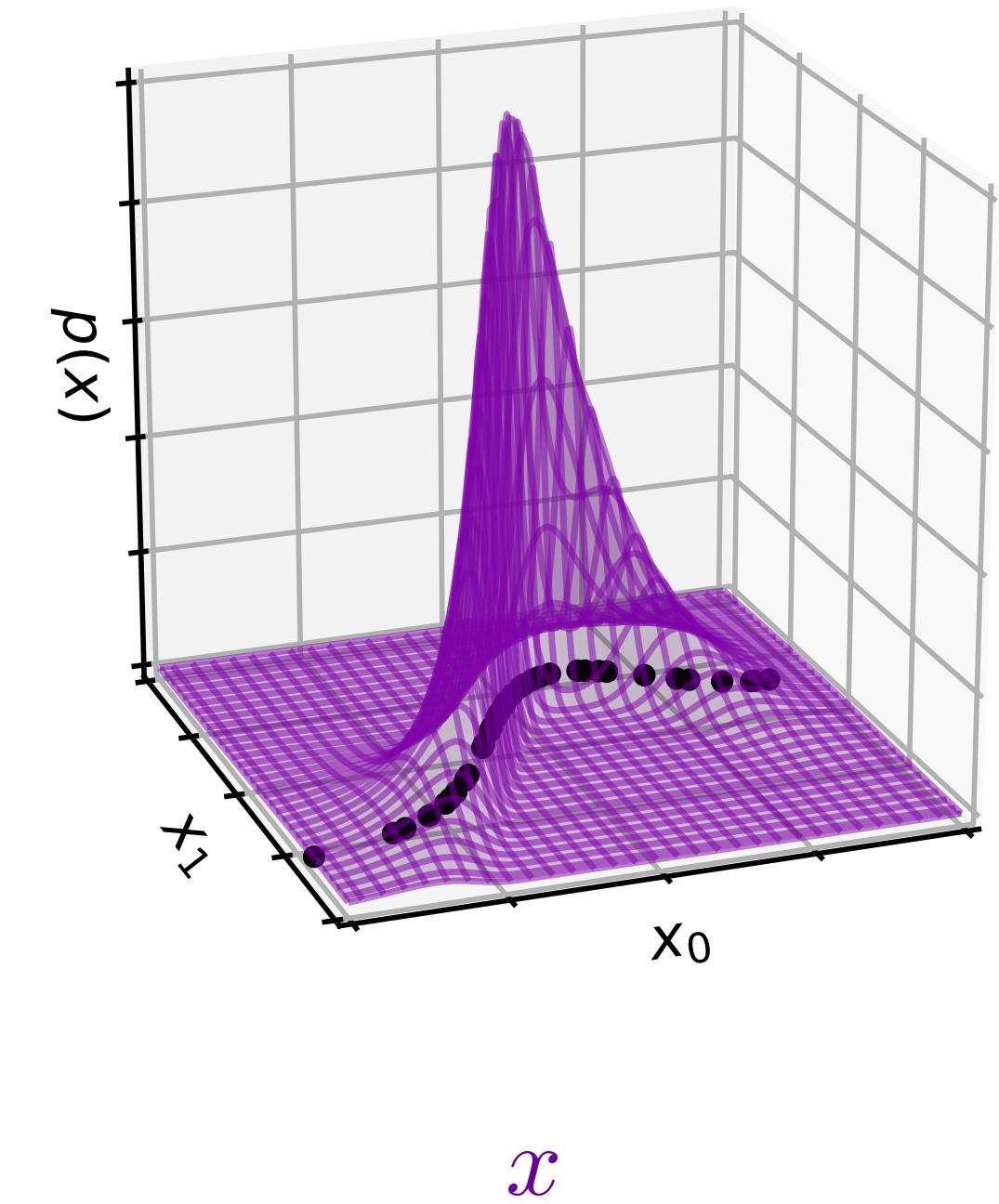
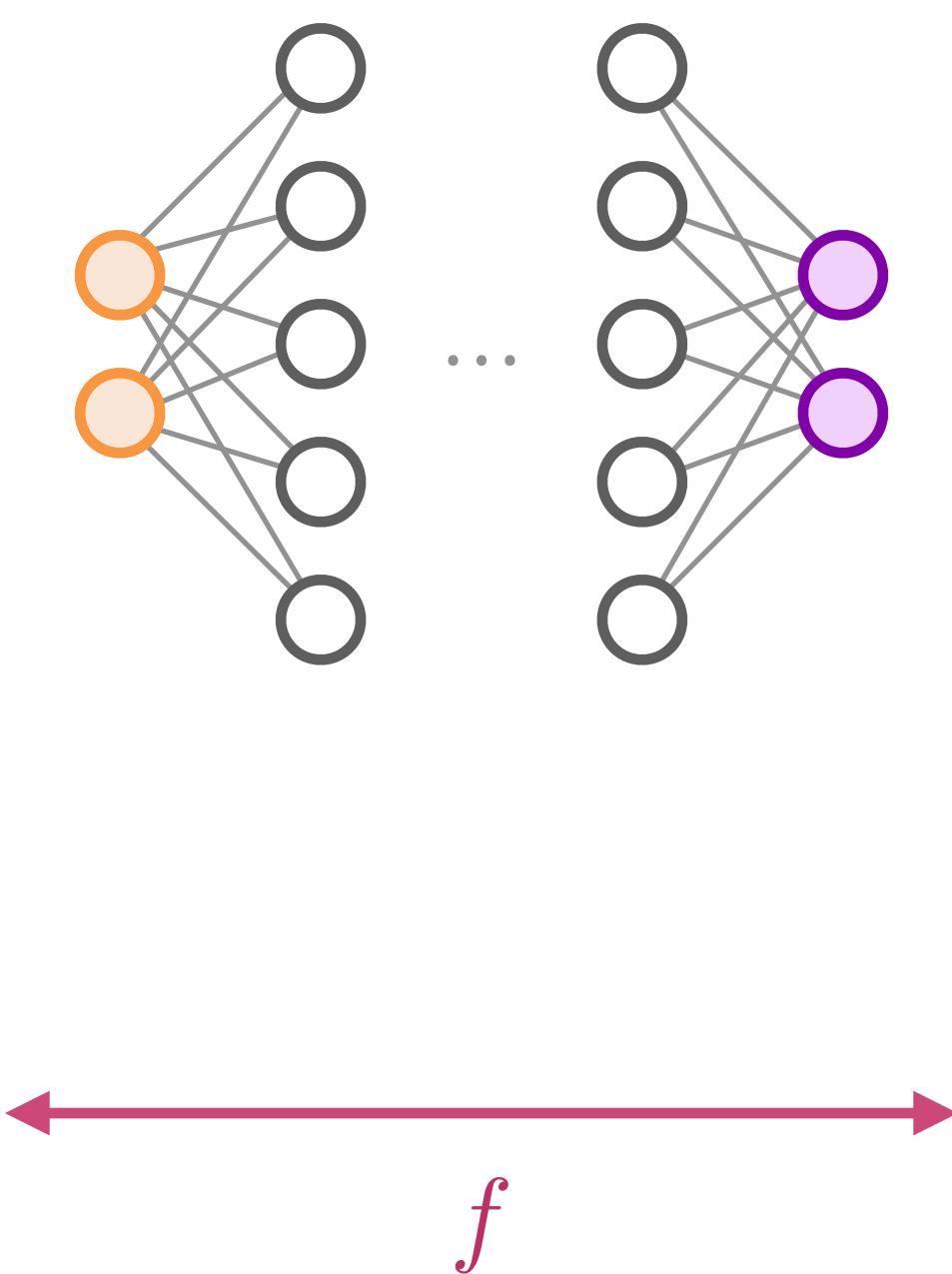


# Normalizing flows in the ambient data space

[L. Dinh et al 1410.8516; ...;  
G. Papamakarios et al 1912.02762]



$d$ -dim. latent variables

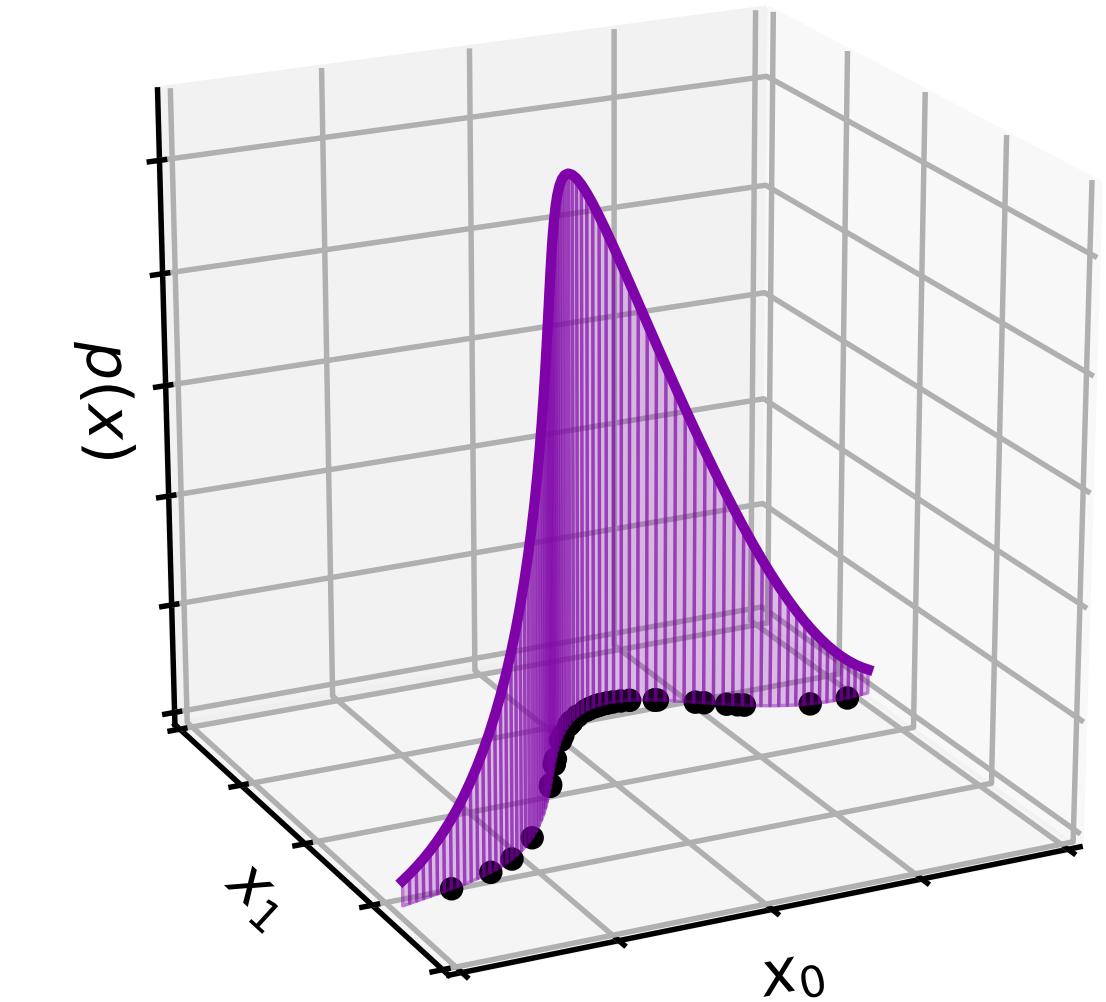
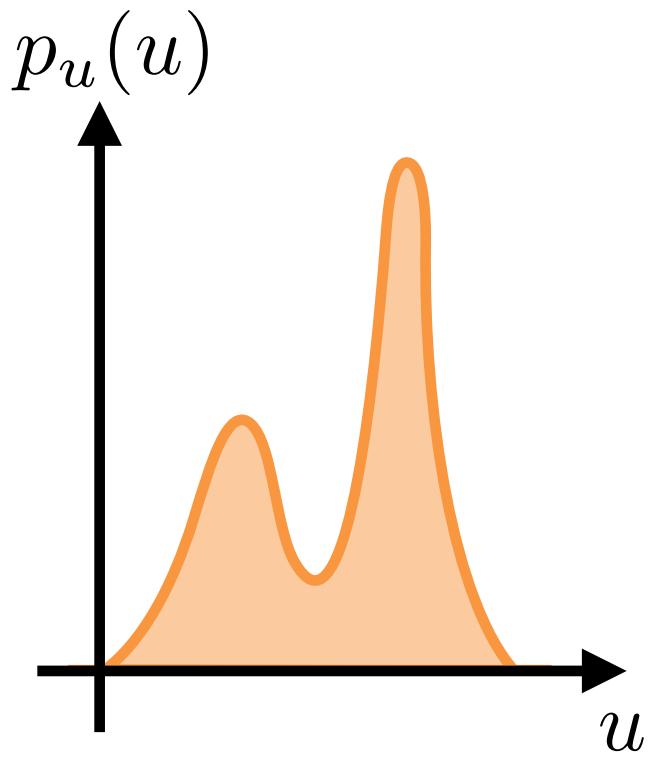
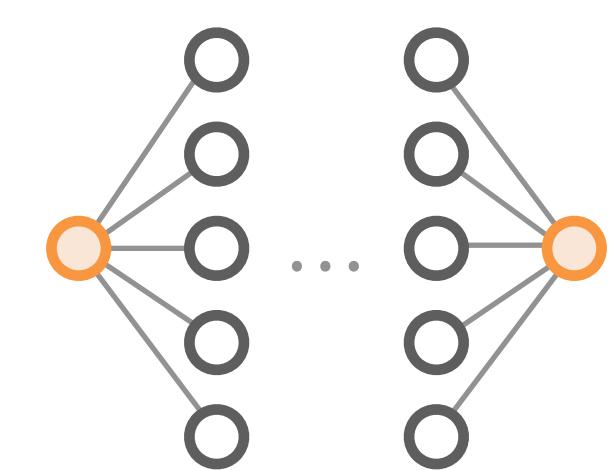
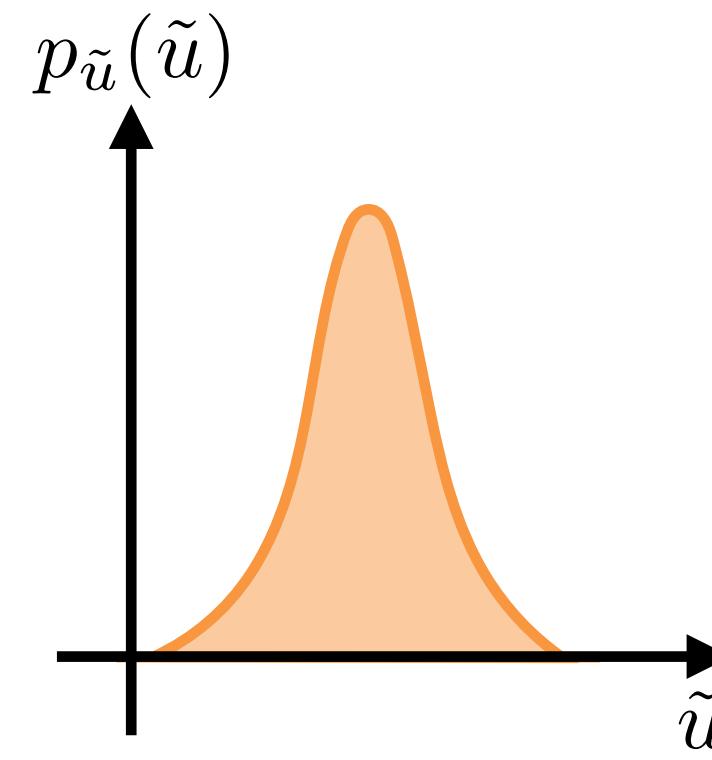


tractable density over  
ambient data space

$$p_x(x) = p_u(f^{-1}(x)) |\det J_f(f^{-1}(x))|^{-1}$$

# Flows on a prescribed manifold

[M. Gemici et al 1611.02304; D. Rezende et al 2002.02428;  
E. Mathieu et al 2006.10605; L. Falorsi et al 2006.06663]



$$\tilde{u} \sim p_{\tilde{u}}(\tilde{u})$$

$$\xleftarrow{h}$$

$$u$$

$$\xleftarrow{g^*}$$

$n$ -dim. latents

invertible NN

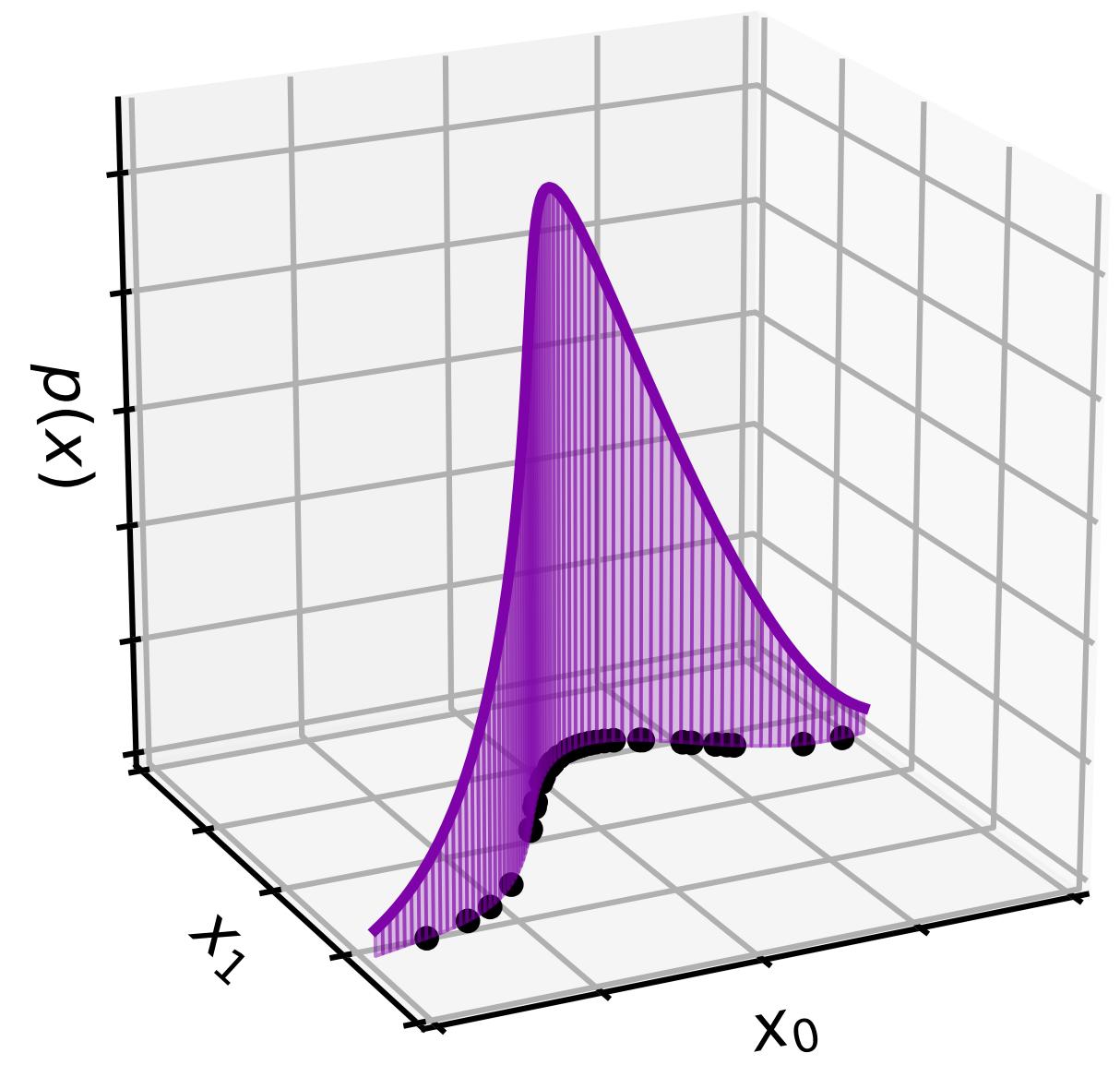
$n$ -dim. latents

prescribed chart

tractable density over  $\mathcal{M}^*$

$$p_{\mathcal{M}^*}(x) = p_{\tilde{u}}(\tilde{u}) |\det J_h(\tilde{u})|^{-1}$$

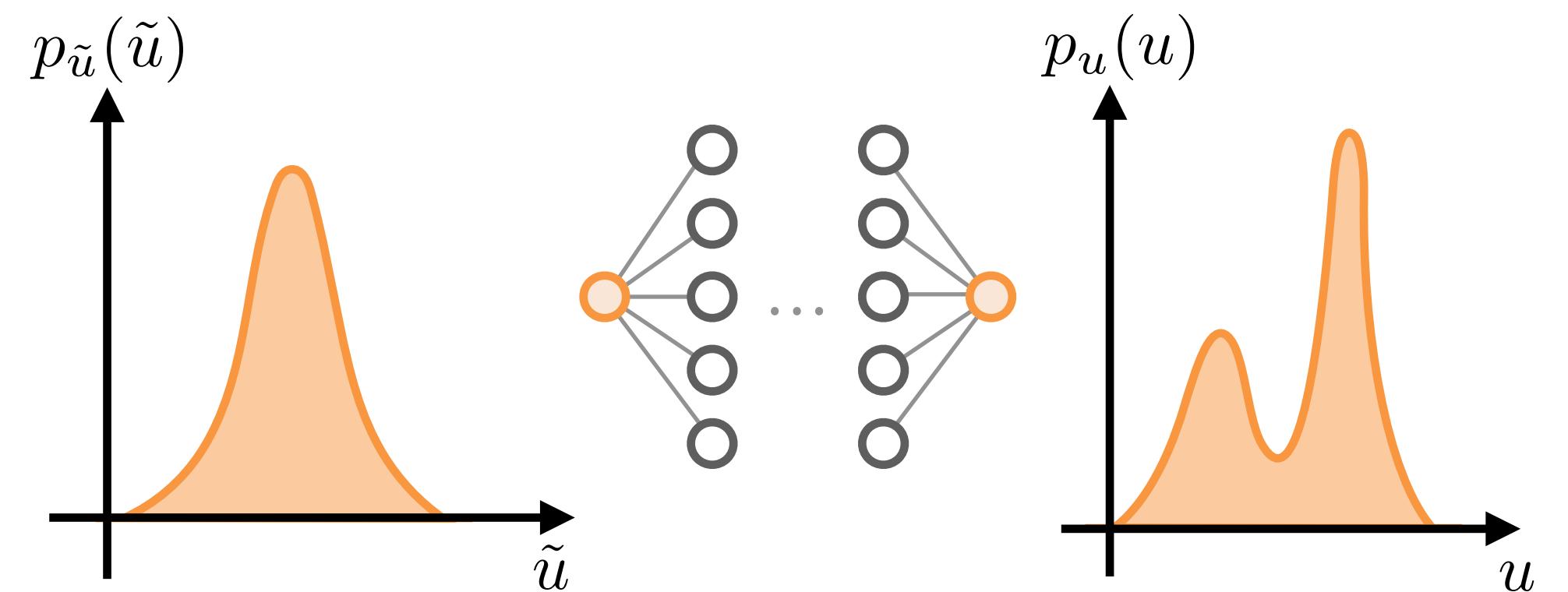
$$\cdot |\det [J_{g^*}^T(u) J_{g^*}(u)]|^{-\frac{1}{2}}$$



$\mathcal{M}$ -flows

# $\mathcal{M}$ -flows

[JB, Kyle Cranmer 2003.13913]

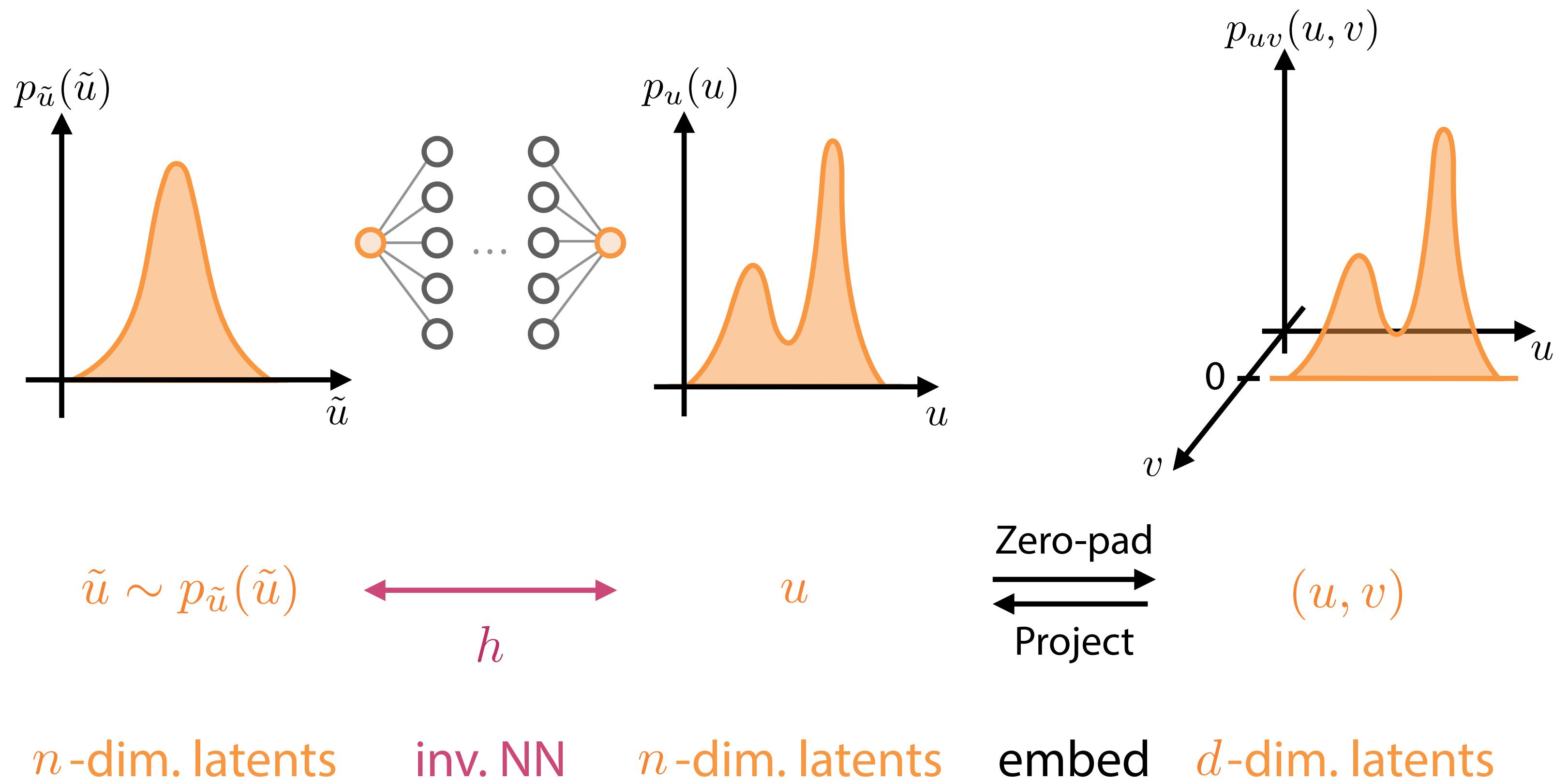


$$\tilde{u} \sim p_{\tilde{u}}(\tilde{u}) \quad \xleftarrow[h]{} \quad u$$

$n$ -dim. latents      inv. NN       $n$ -dim. latents

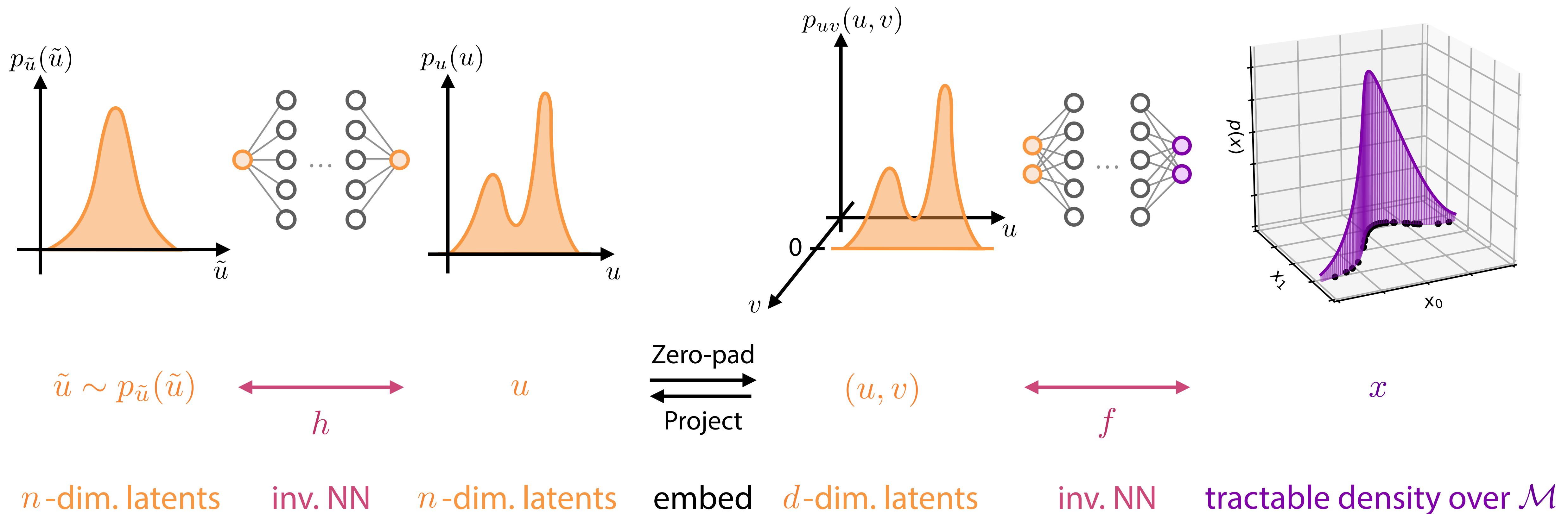
# $\mathcal{M}$ -flows

[JB, Kyle Cranmer 2003.13913]



# $\mathcal{M}$ -flows

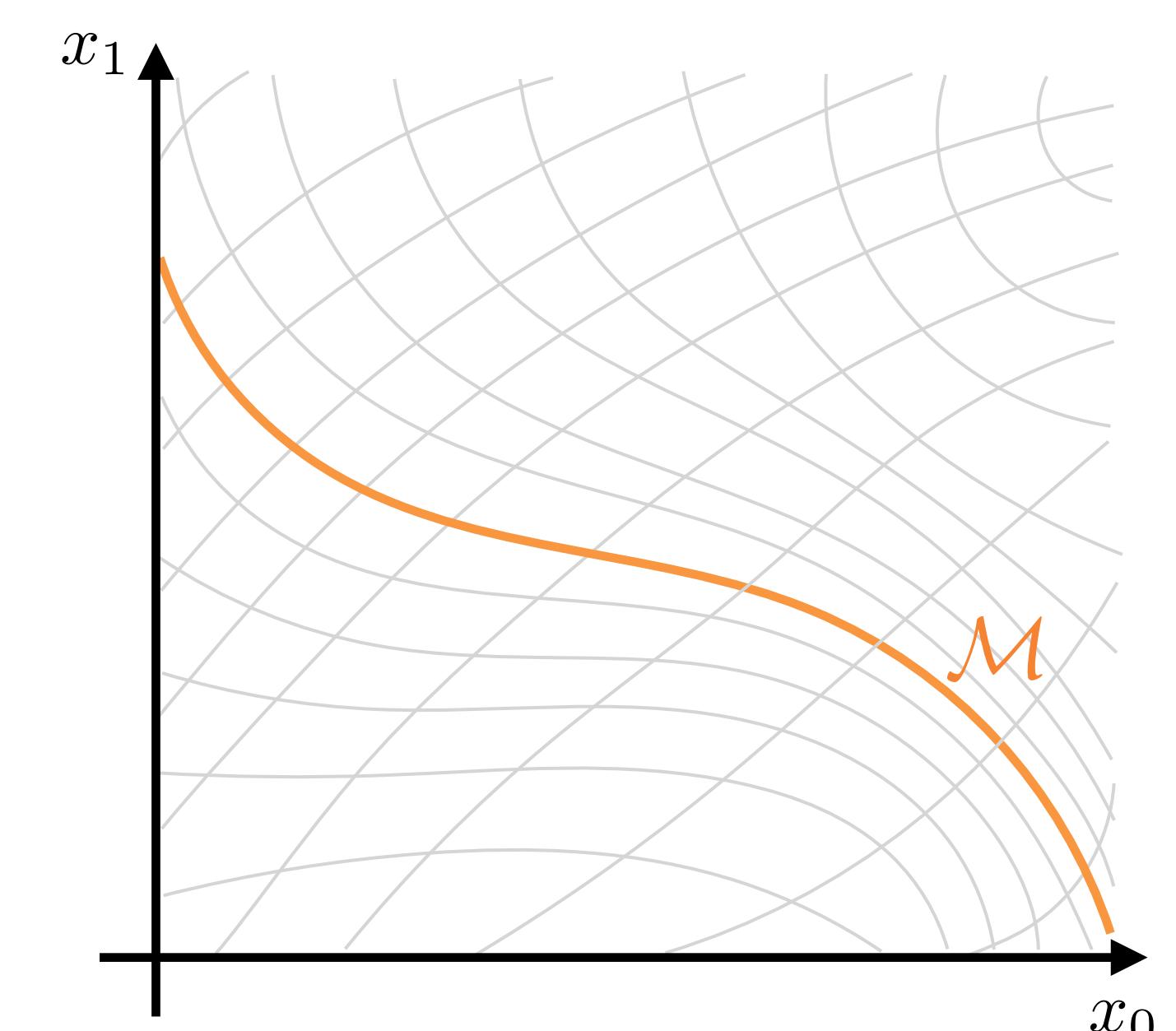
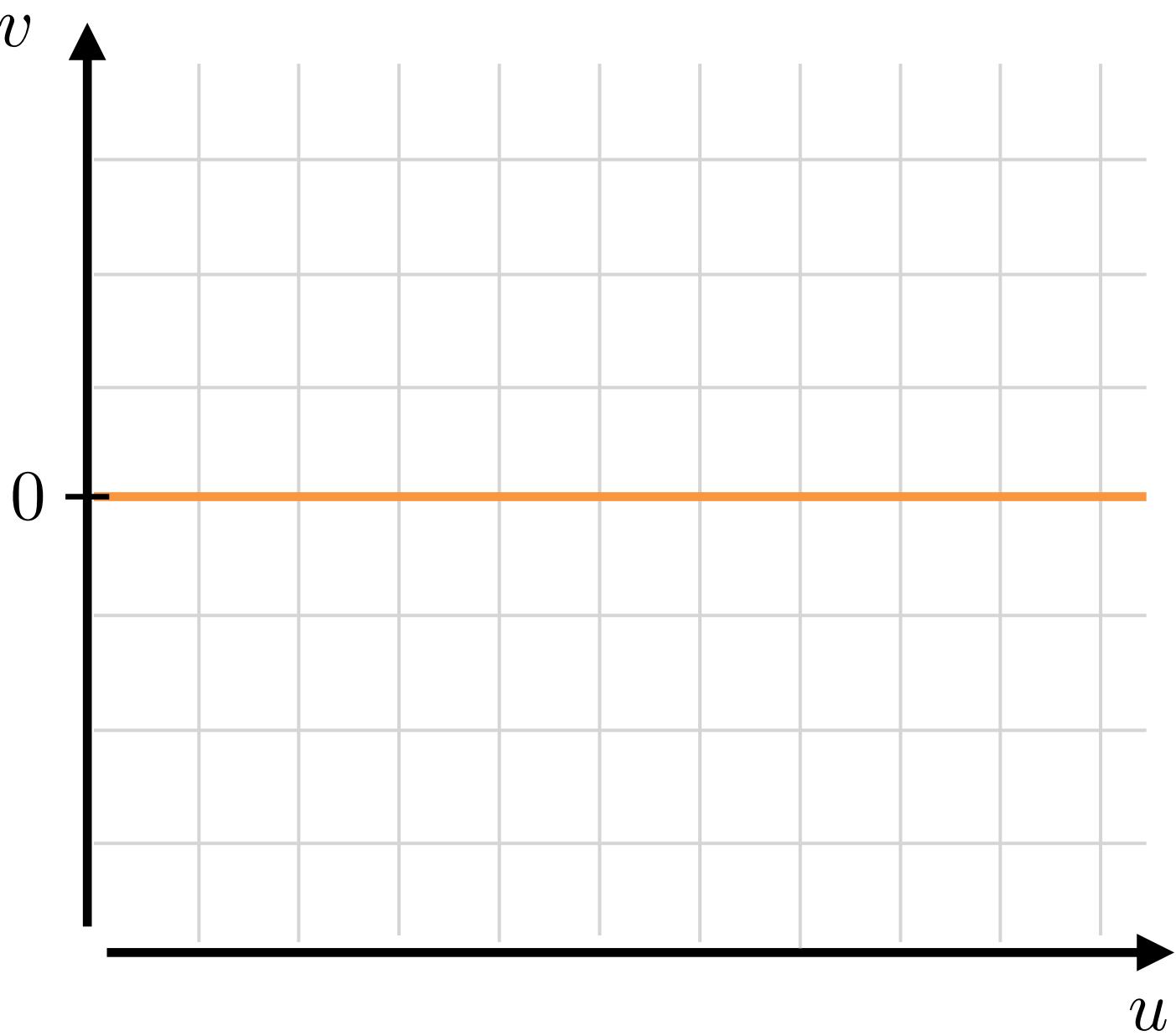
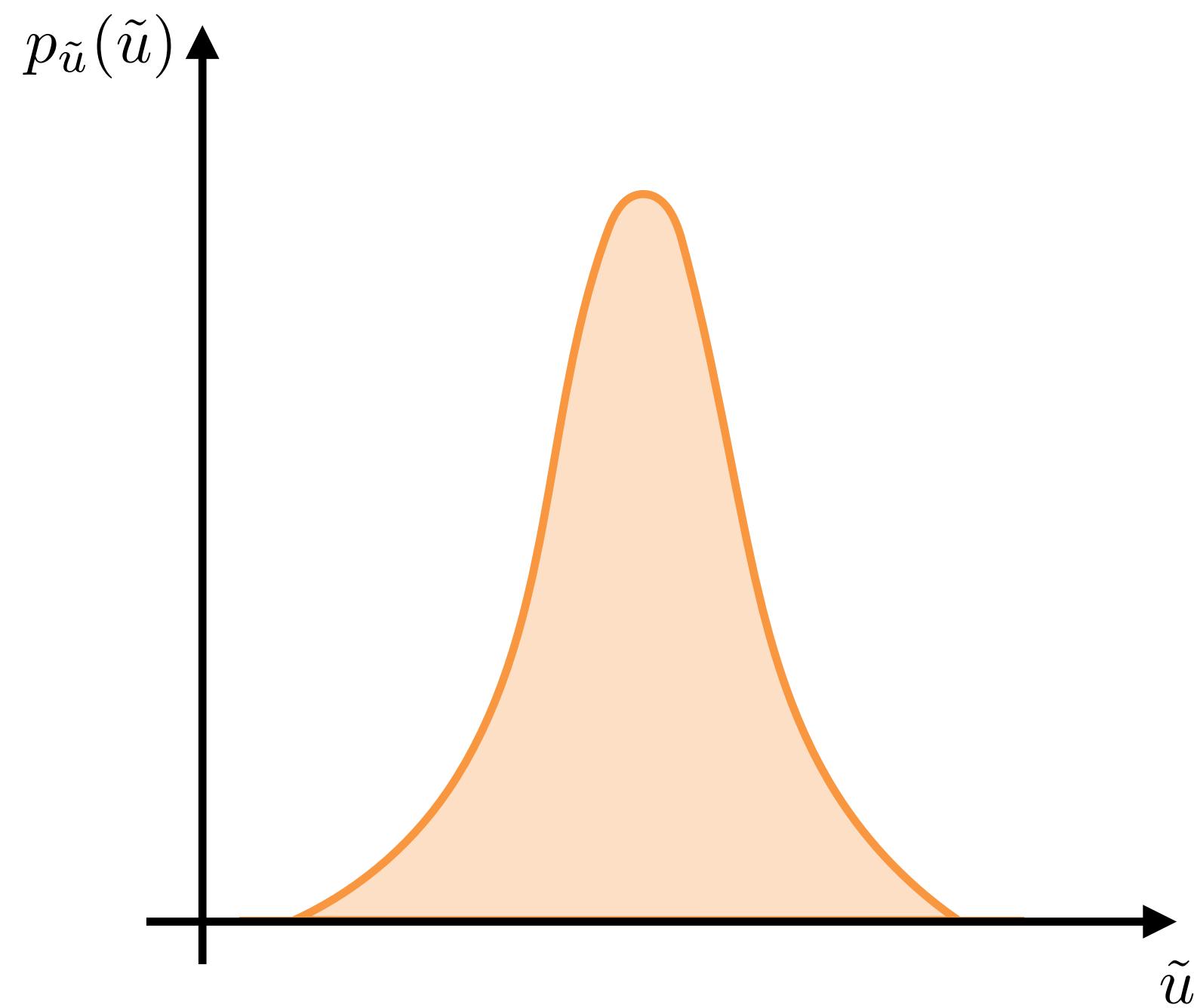
[JB, Kyle Cranmer 2003.13913]



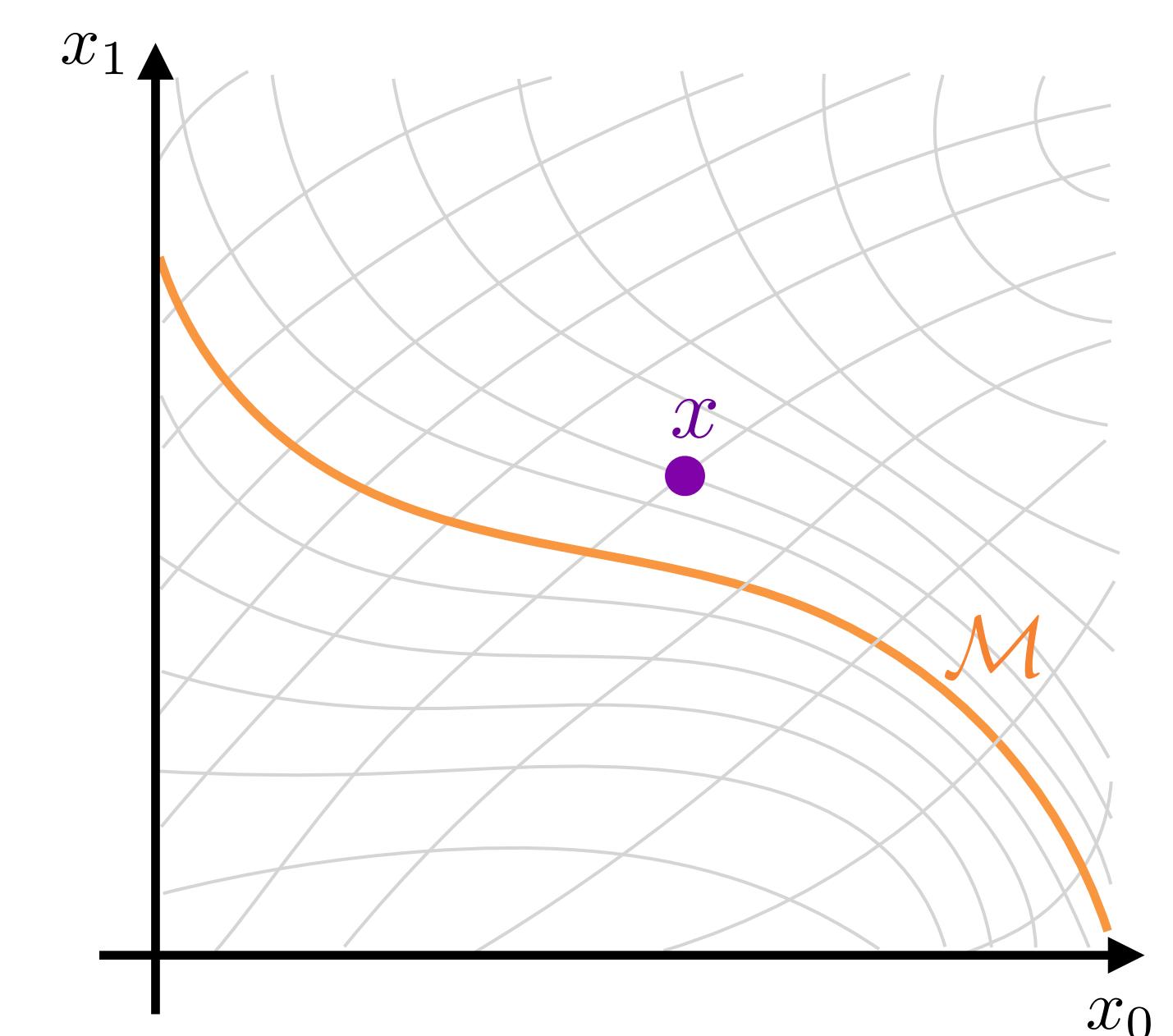
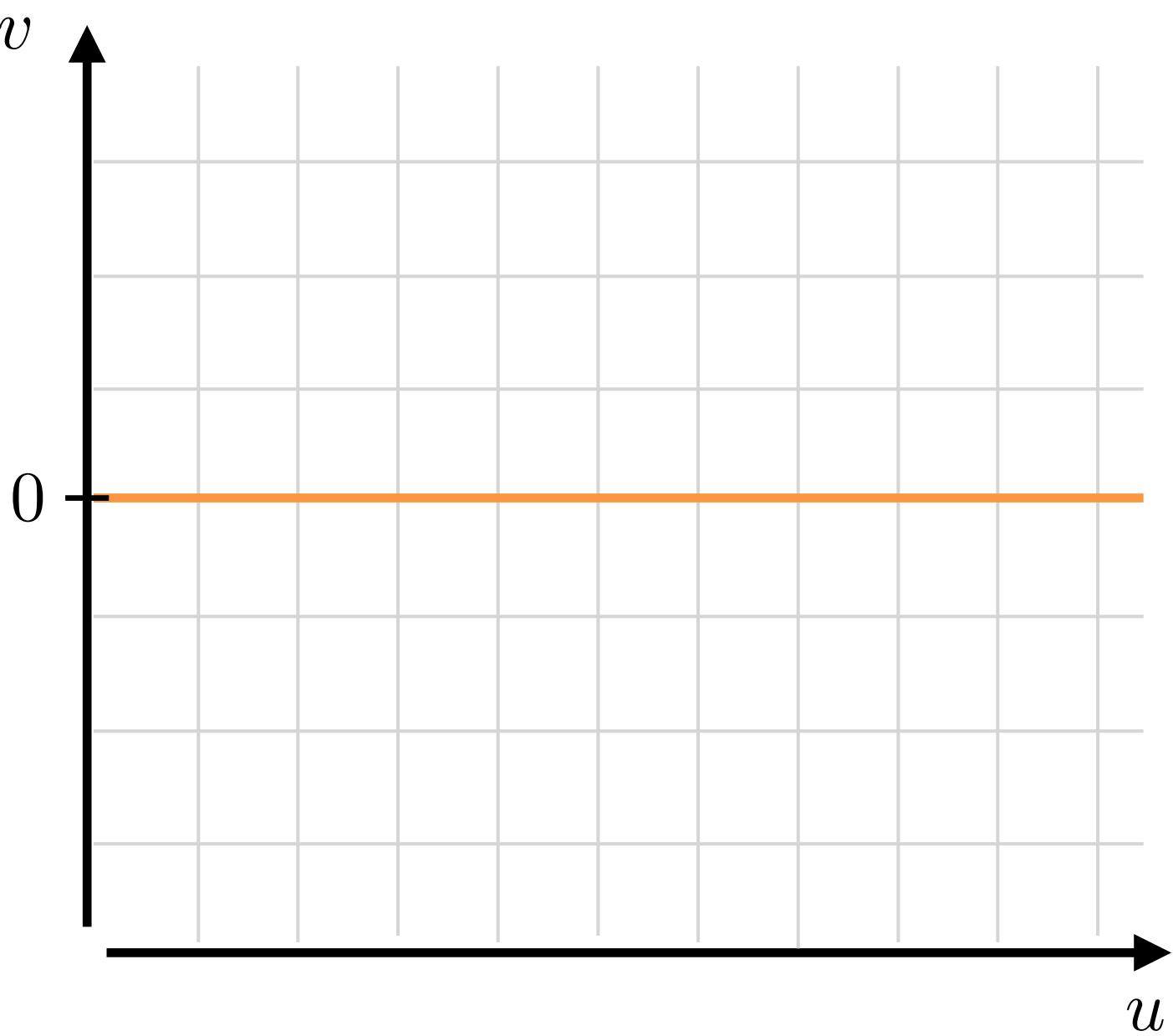
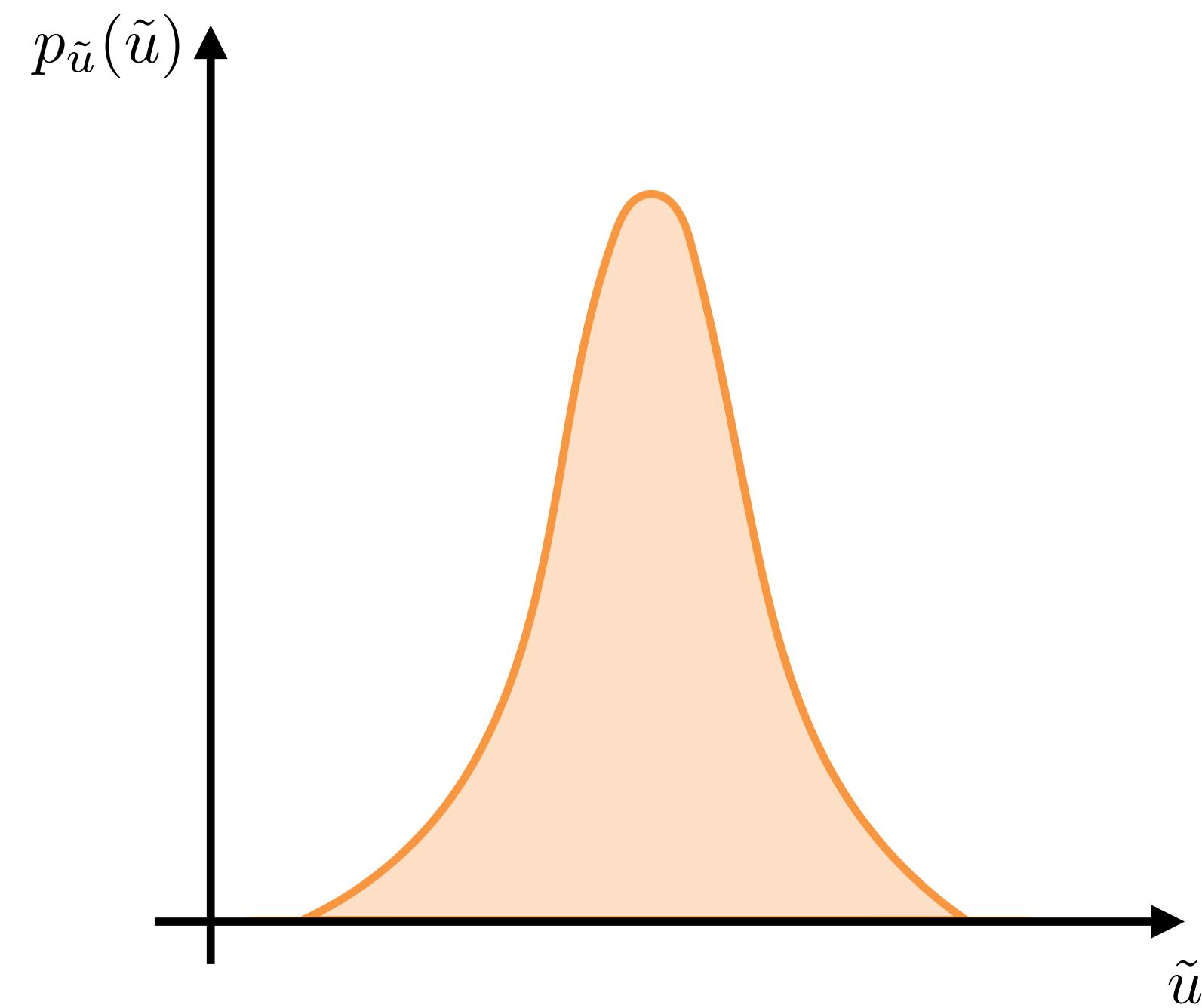
$$p_{\mathcal{M}}(x) = p_{\tilde{u}}(\tilde{u}) |\det J_h(\tilde{u})|^{-1}$$

$$\cdot \left| \det \left[ (\mathbb{1} \ 0) J_f(u)^T J_f(u) \begin{pmatrix} \mathbb{1} \\ 0 \end{pmatrix} \right] \right|^{-\frac{1}{2}}$$

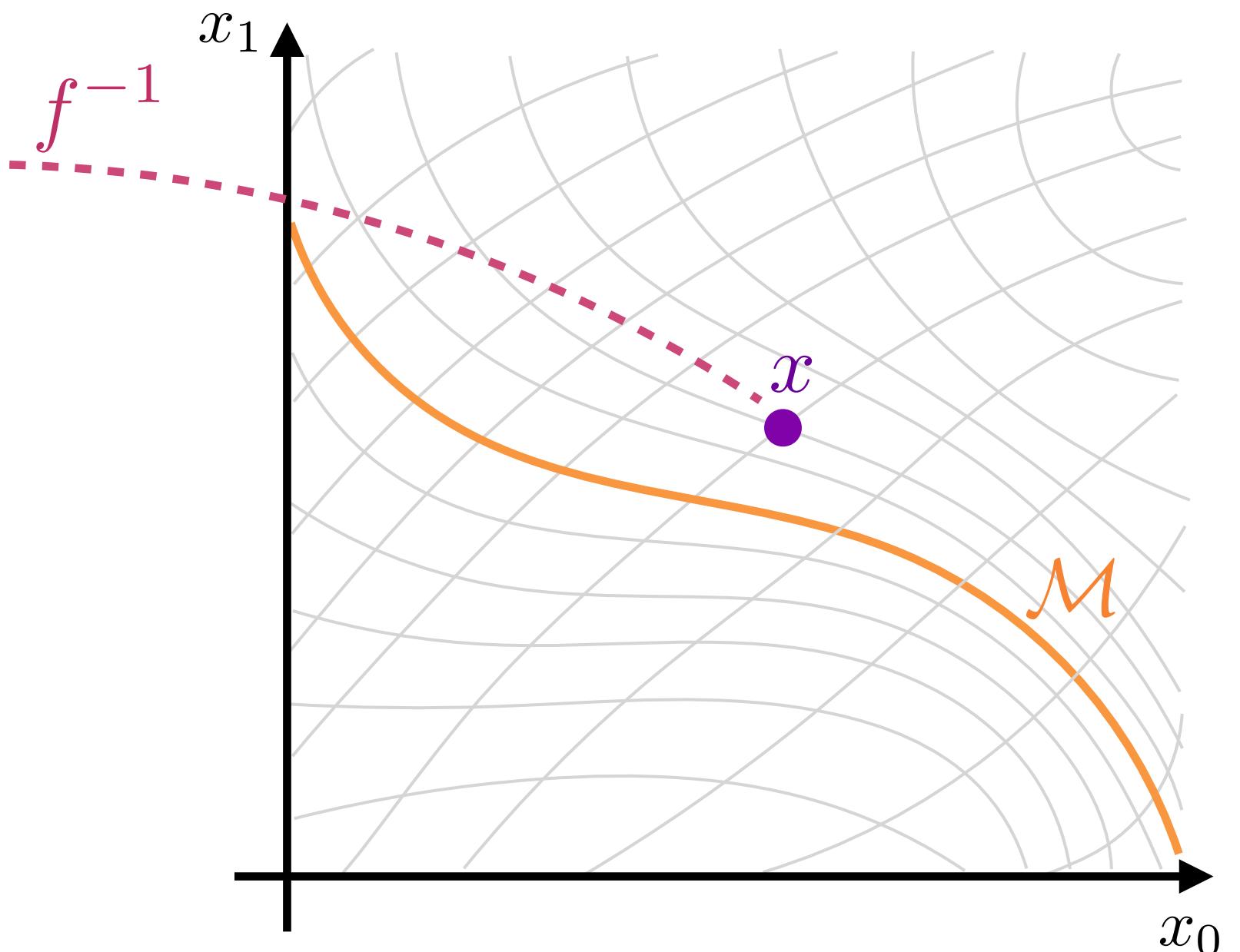
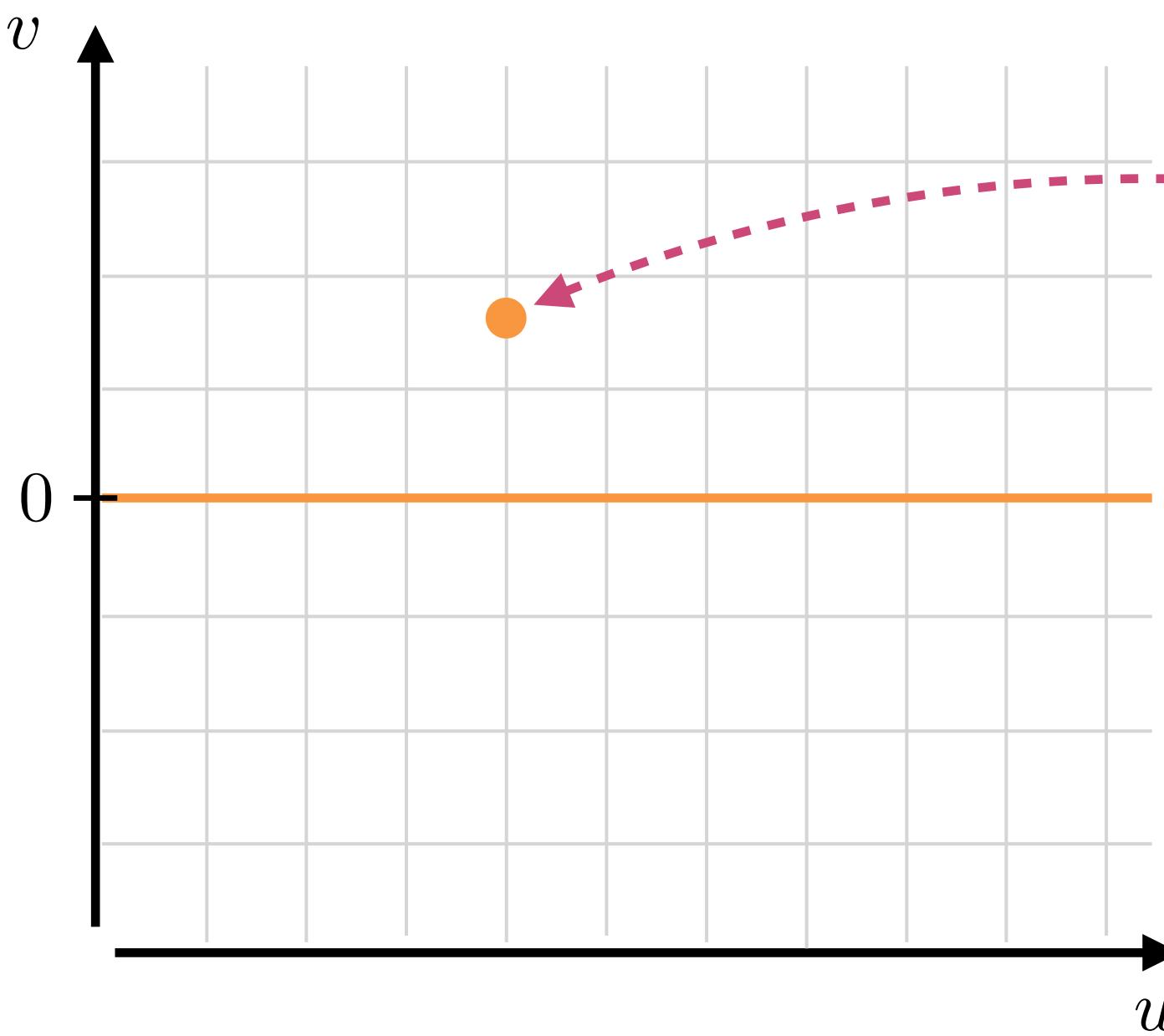
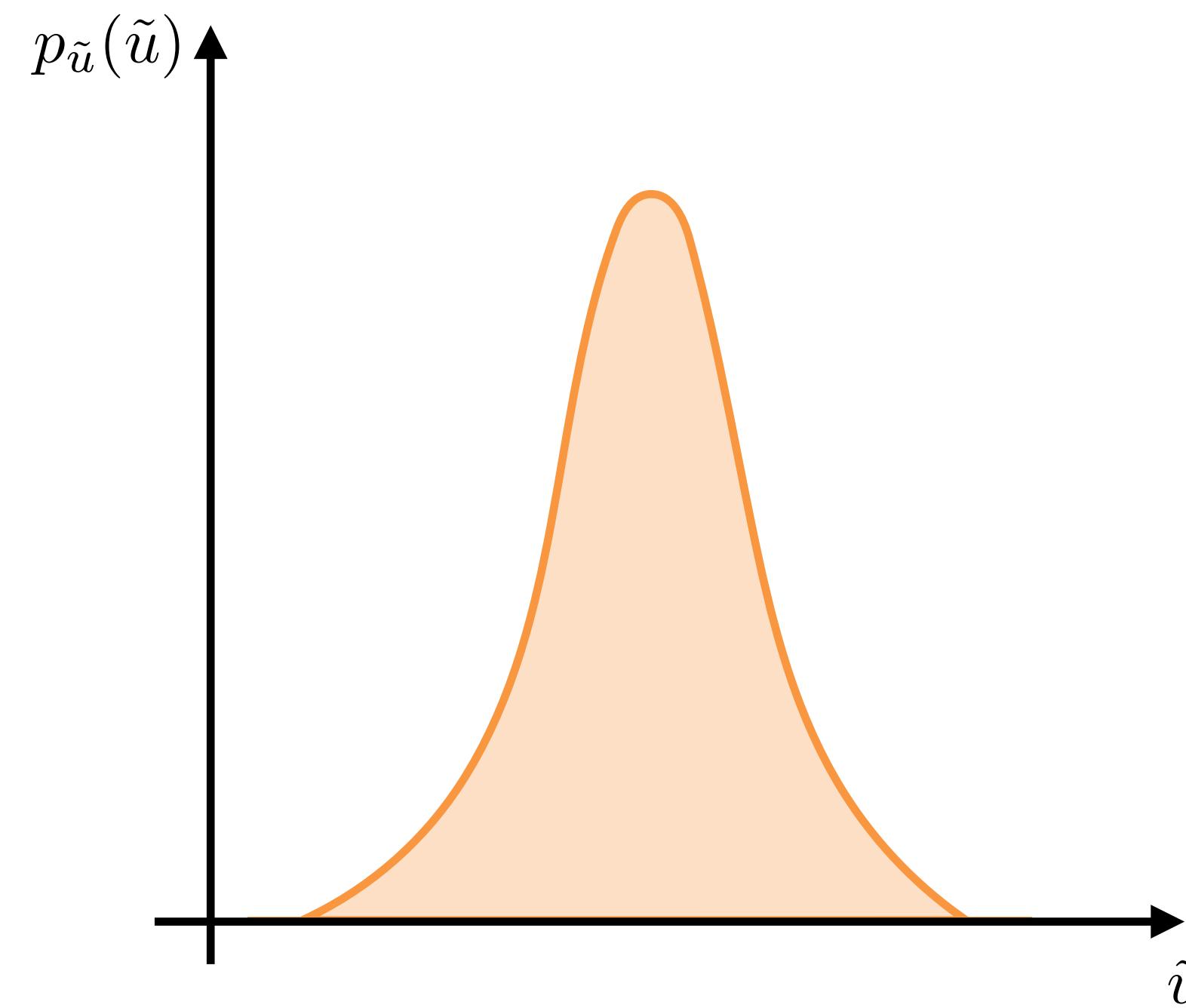
# Evaluating data on or off the manifold



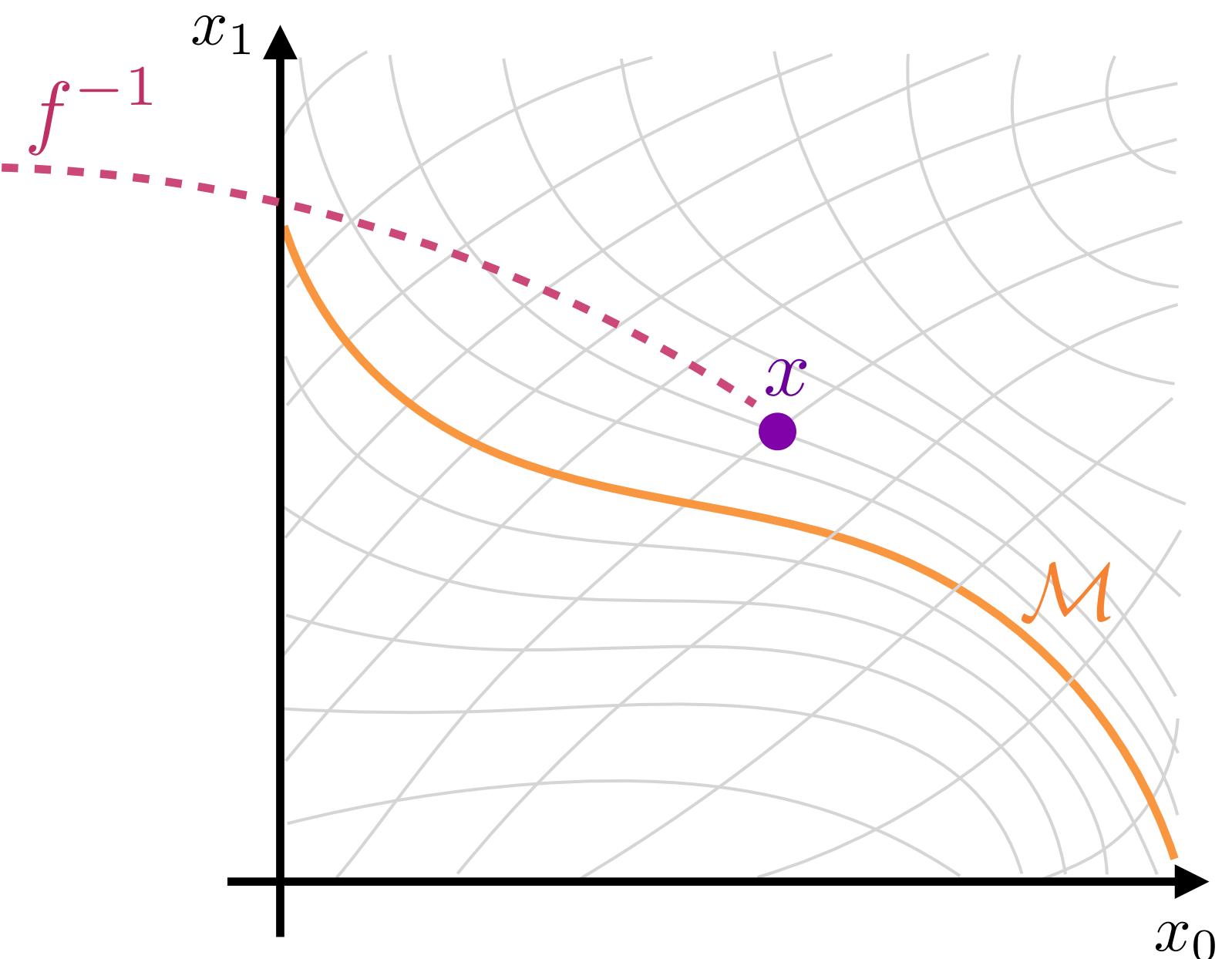
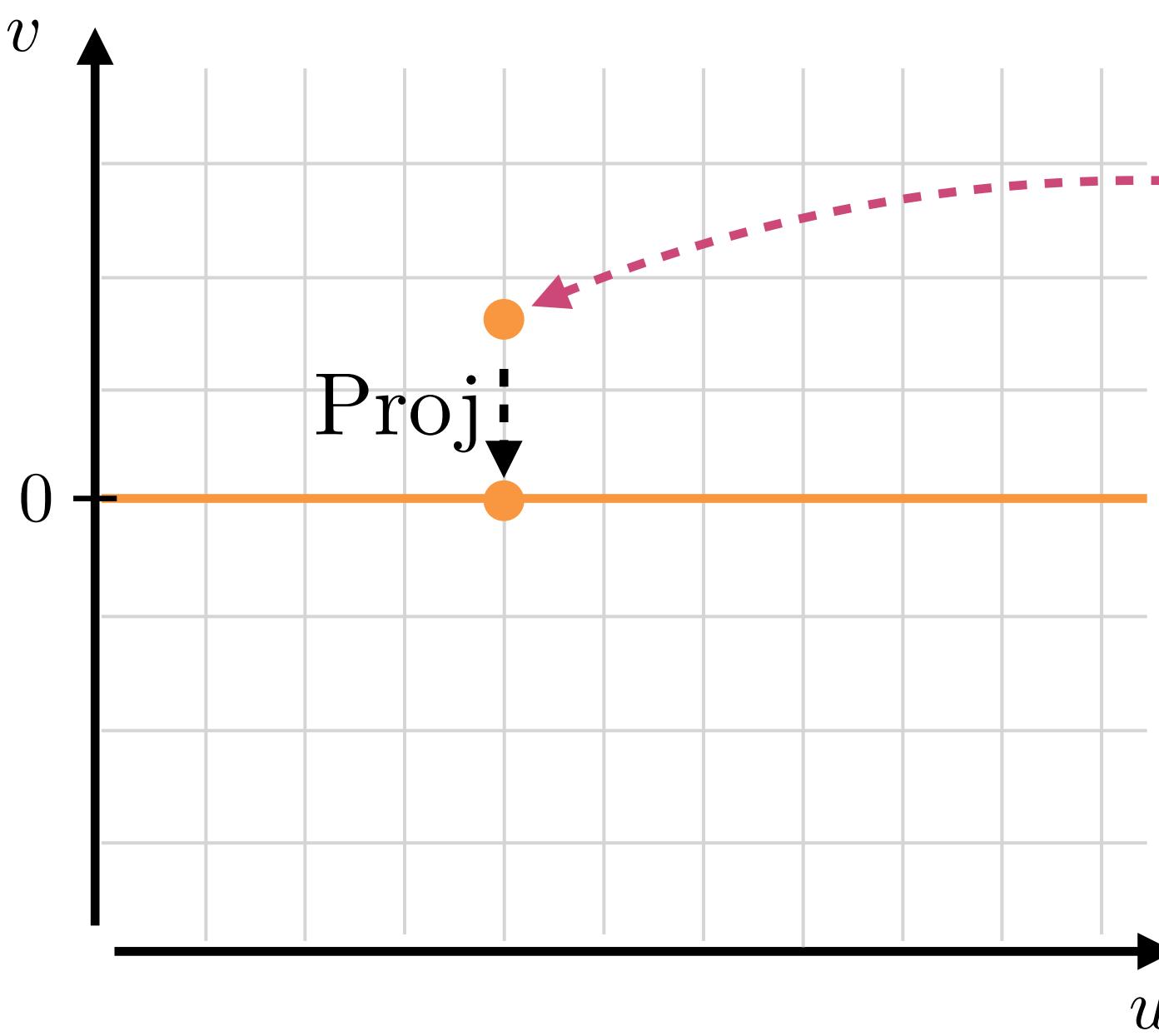
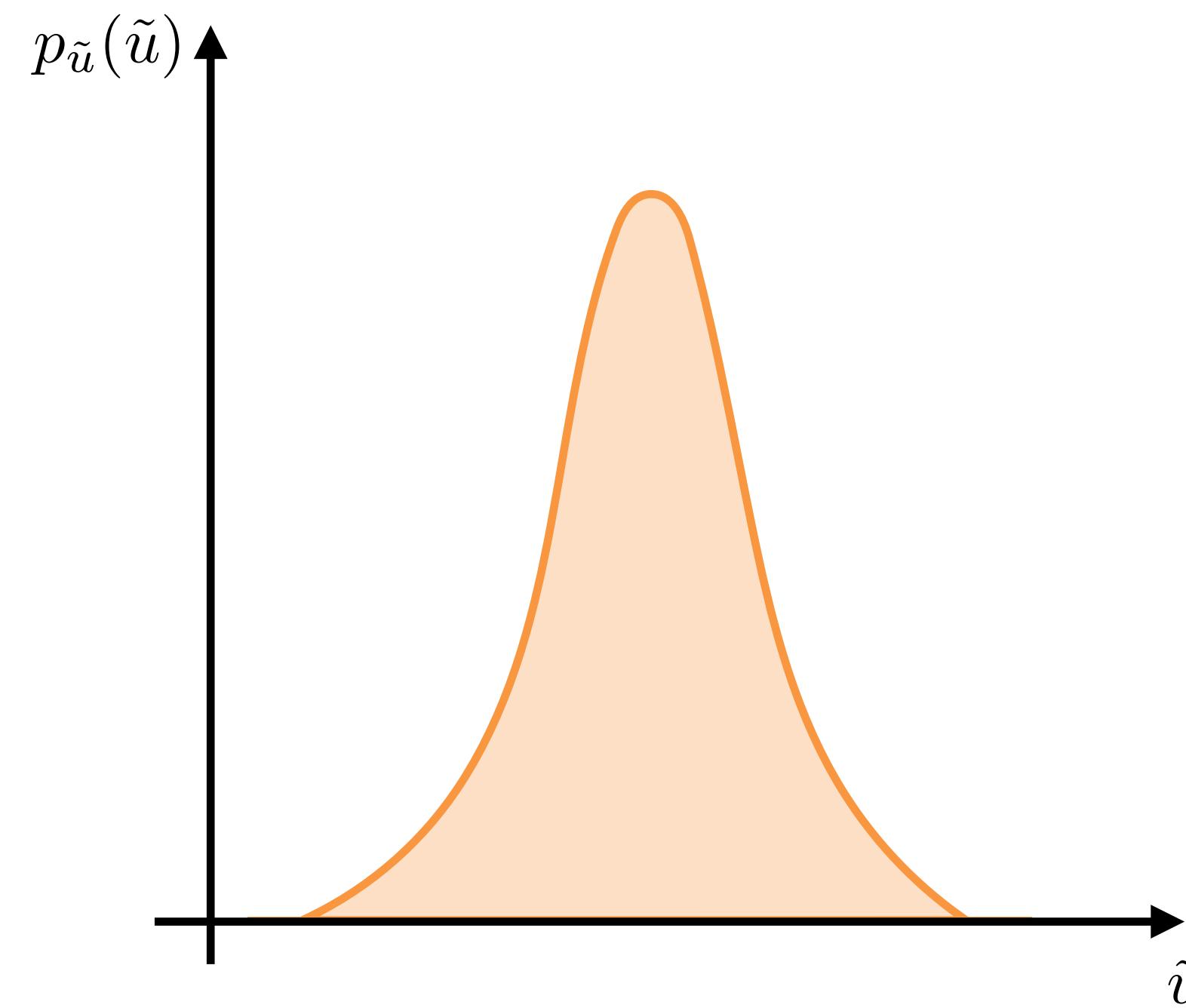
# Evaluating data on or off the manifold



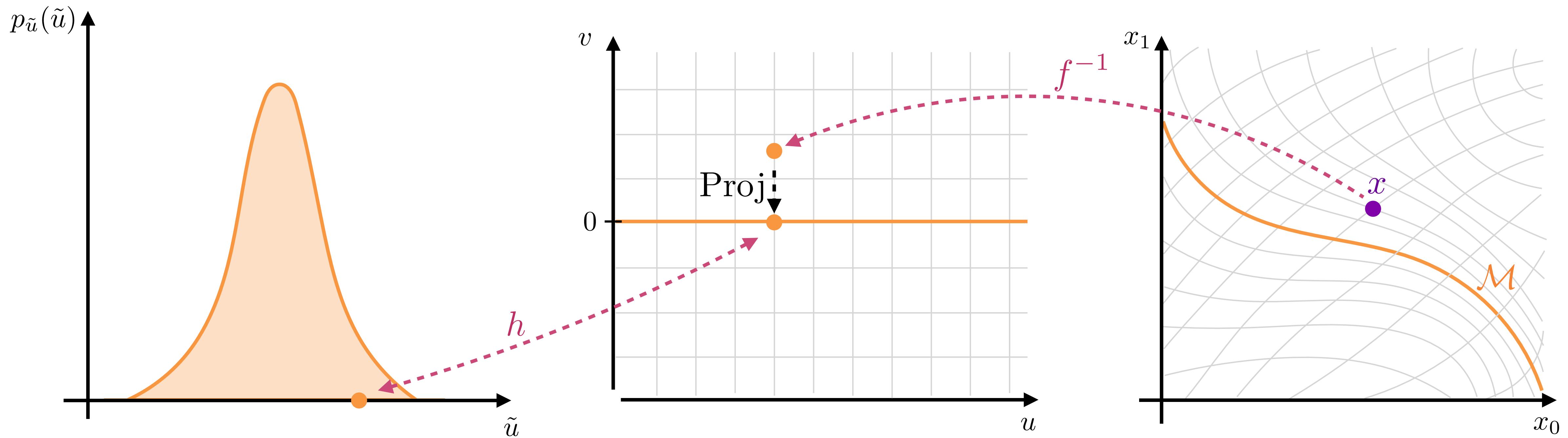
# Evaluating data on or off the manifold



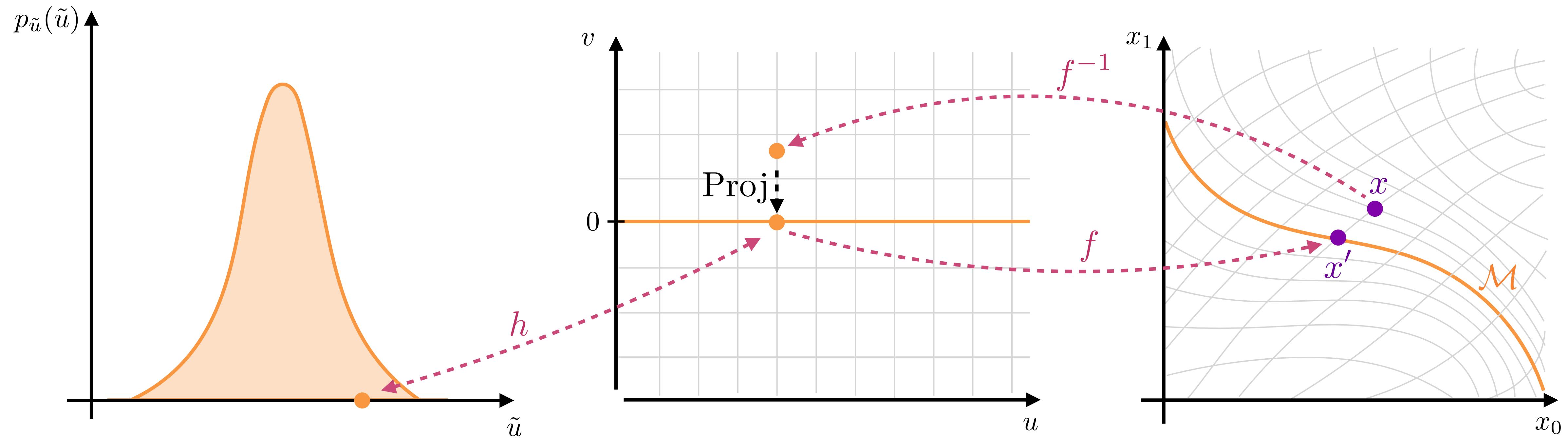
# Evaluating data on or off the manifold



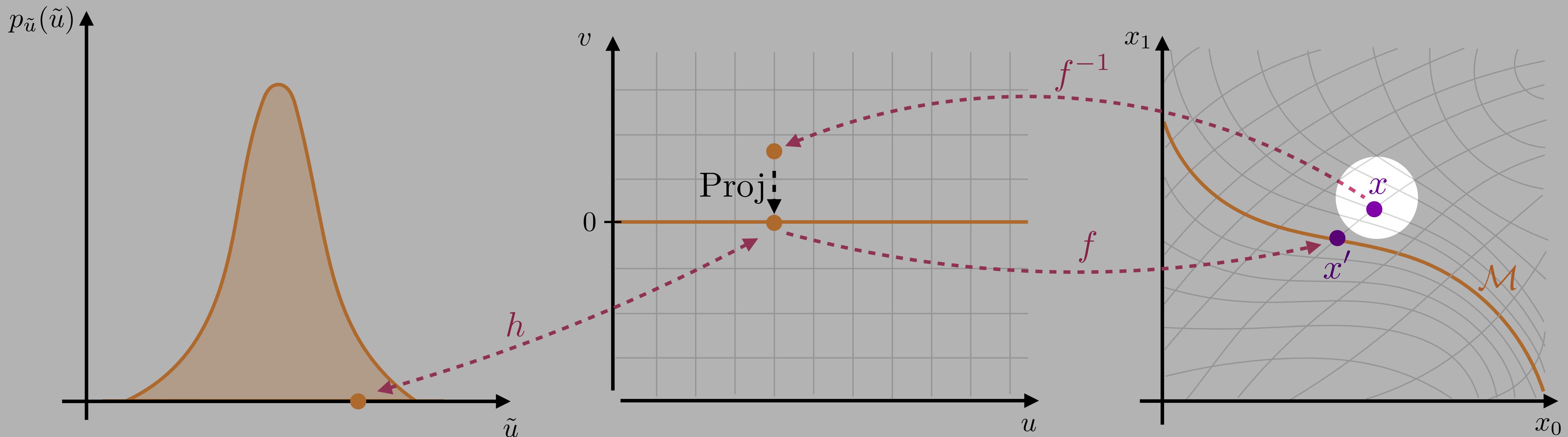
# Evaluating data on or off the manifold



# Evaluating data on or off the manifold

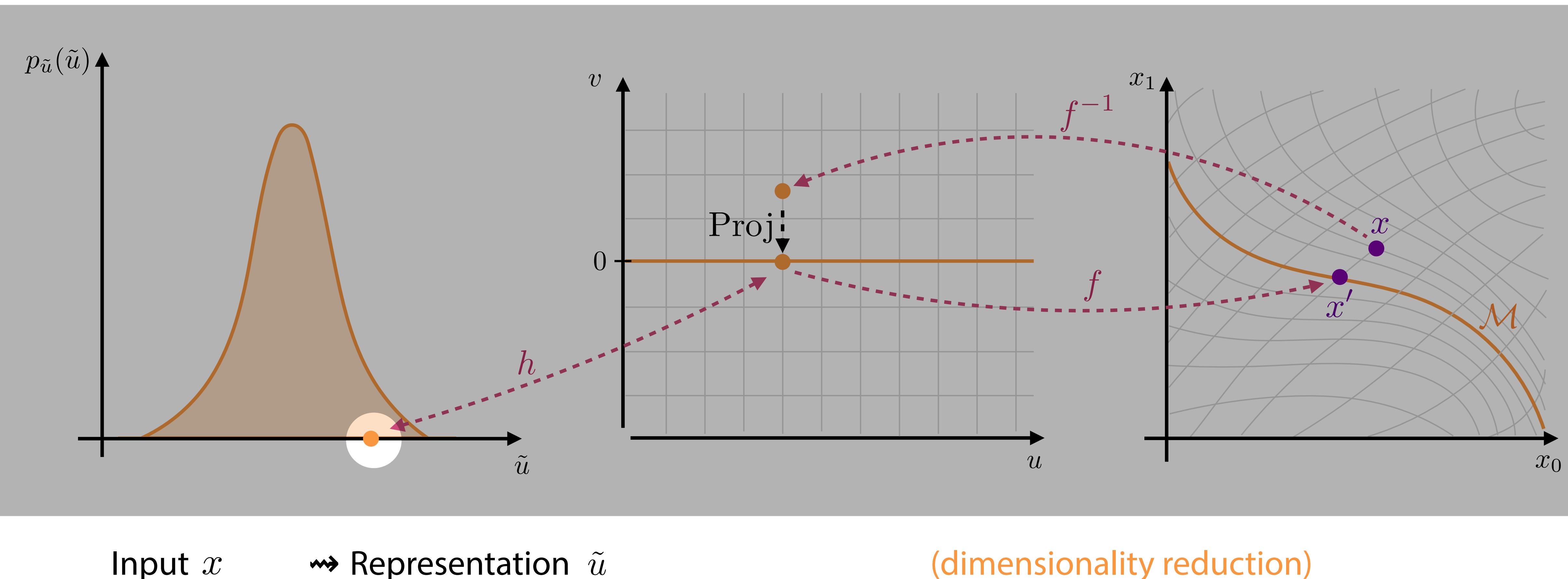


# Evaluating data on or off the manifold

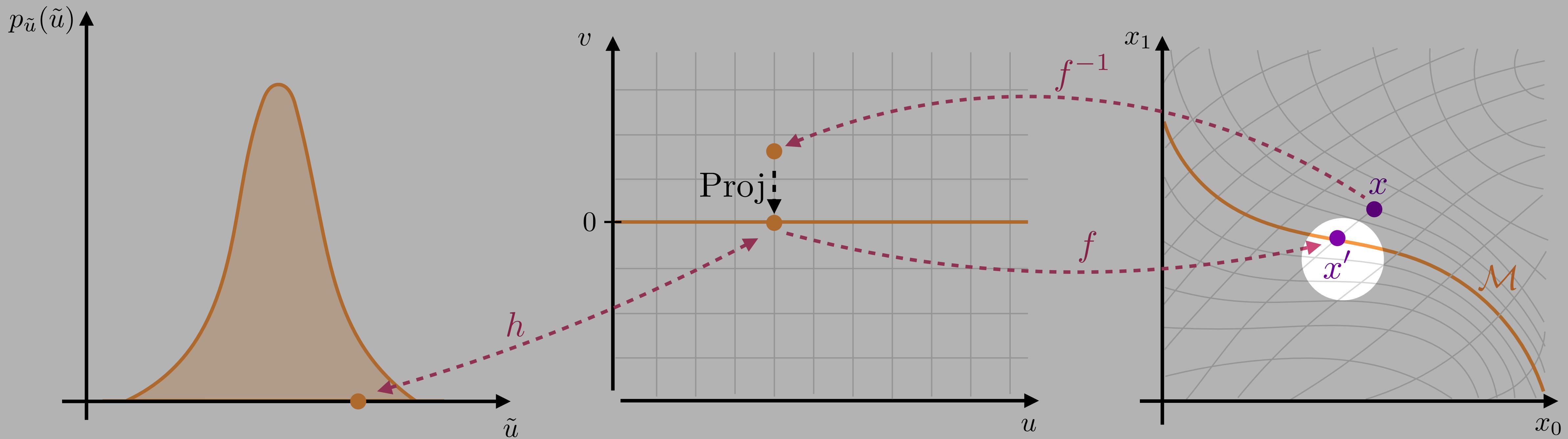


Input  $x$

# Evaluating data on or off the manifold



# Evaluating data on or off the manifold



Input  $x$

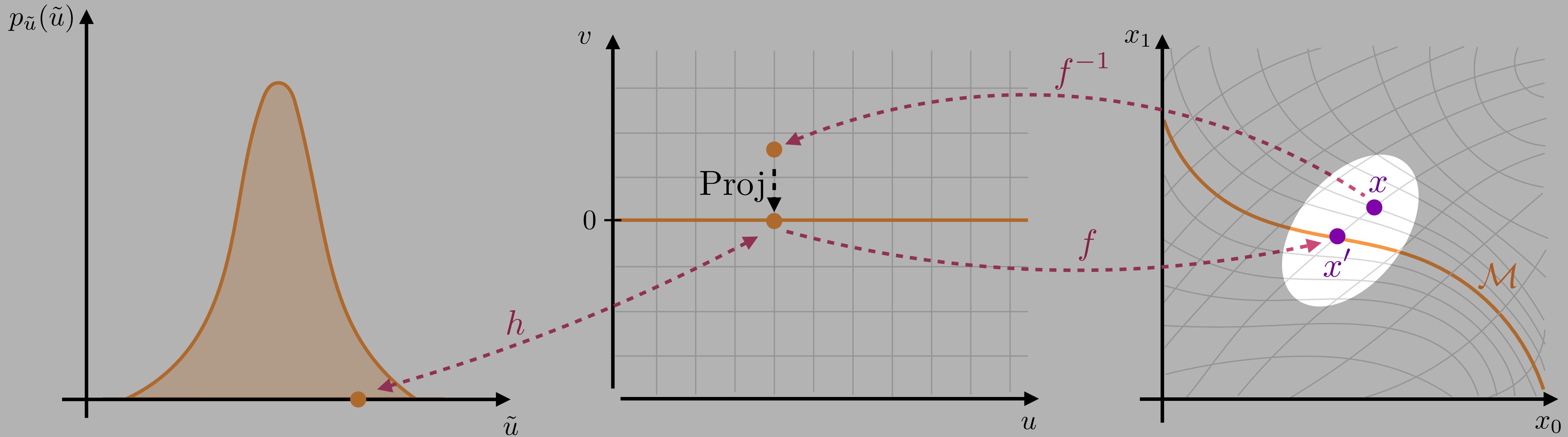
↔ Representation  $\tilde{u}$

↔ Projection to manifold  $x'$

(dimensionality reduction)

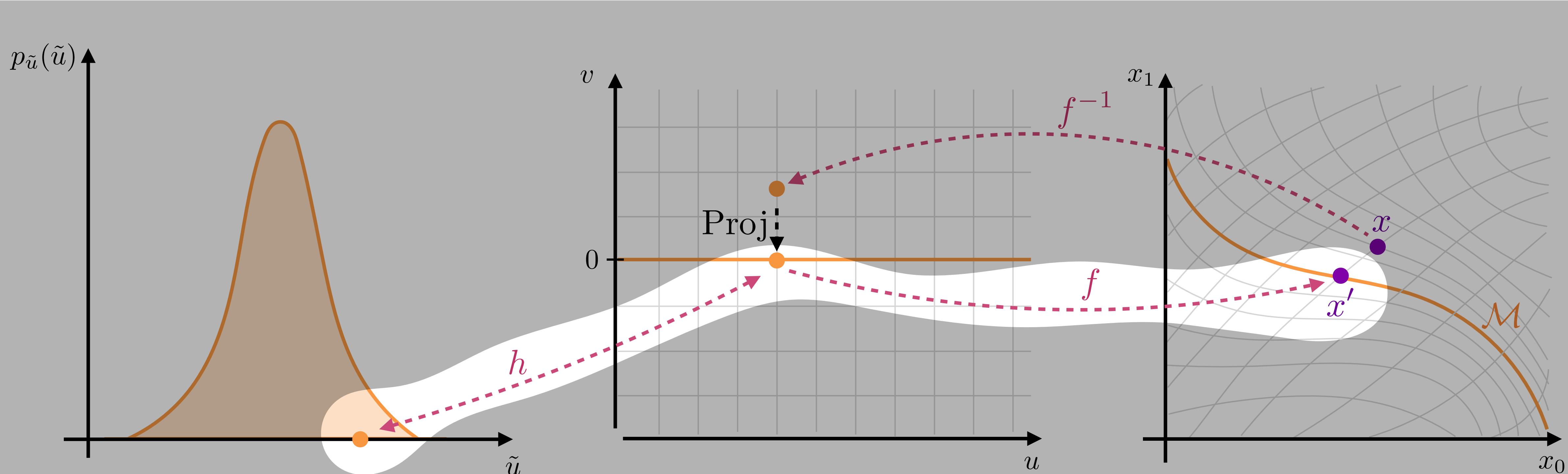
(denoising)

# Evaluating data on or off the manifold



- |           |                                     |                            |
|-----------|-------------------------------------|----------------------------|
| Input $x$ | ↔ Representation $\tilde{u}$        | (dimensionality reduction) |
|           | ↔ Projection to manifold $x'$       | (denoising)                |
|           | ↔ Reconstruction error $\ x - x'\ $ | (training, OOD detection)  |

# Evaluating data on or off the manifold



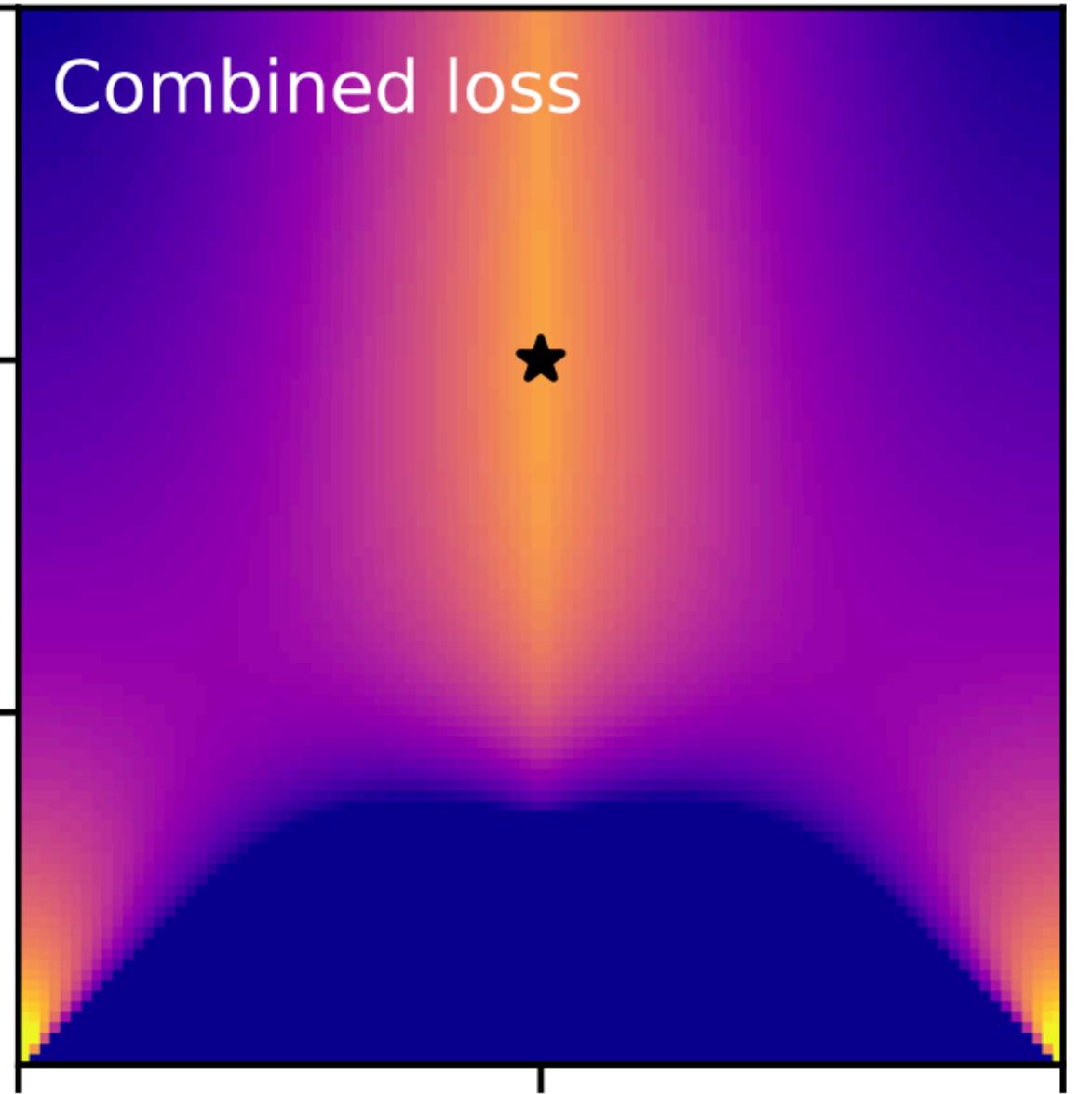
- |           |   |                            |
|-----------|---|----------------------------|
| Input $x$ | ↔ Representation $\tilde{u}$                        | (dimensionality reduction) |
|           | ↔ Projection to manifold $x'$                       | (denoising)                |
|           | ↔ Reconstruction error $\ x - x'\ $                 | (training, OOD detection)  |
|           | ↔ Likelihood after projection $p_{\mathcal{M}}(x')$ | (training, inference)      |

# Related work

- **Pseudo-Invertible Encoder (PIE):**  
add “narrow” base density  $p(v)$  for off-the-manifold latents,  
likelihood over ambient space (inconsistent with sampling)  
[J. J. Beitler, I. Sosnovik, A. Smeulders 2018]
- **Probabilistic Auto-Encoder:**  
classic autoencoder instead of flow  $f$ ,  
likelihood intractable  
[V. Böhm, U. Seljak 2006.05479]
- **Relaxed Injective Probability Flows:**  
classic autoencoder + bounds on Jacobian,  
stochastic lower bound on likelihood  
[A. Kumar, B. Poole, K. Murphy 2002.08927]

# Generative models vs. the data manifold

Model	Manifold	Chart	Generative	Tractable density	Restr. to manifold
Ambient flow (AF)	no	no	✓	✓	no
Flow on prescr. manifold	prescribed	prescribed	✓	✓	✓
GAN	learned	no	✓	no	✓
VAE	learned	no	✓	only ELBO	(no)
$\mathcal{M}$ -flow	learned	learned	✓	✓ (potentially slow)	✓



Training  $\mathcal{M}$ -flows

# Maximum likelihood is not enough

Likelihood defined after projection to  $\mathcal{M}$ ,  
which is defined through NN weights  $\phi_f$

Family of likelihoods  $p_{\phi_f}(x|\phi_h)$   
rather than one likelihood  $p(x|\phi_f, \phi_h)$

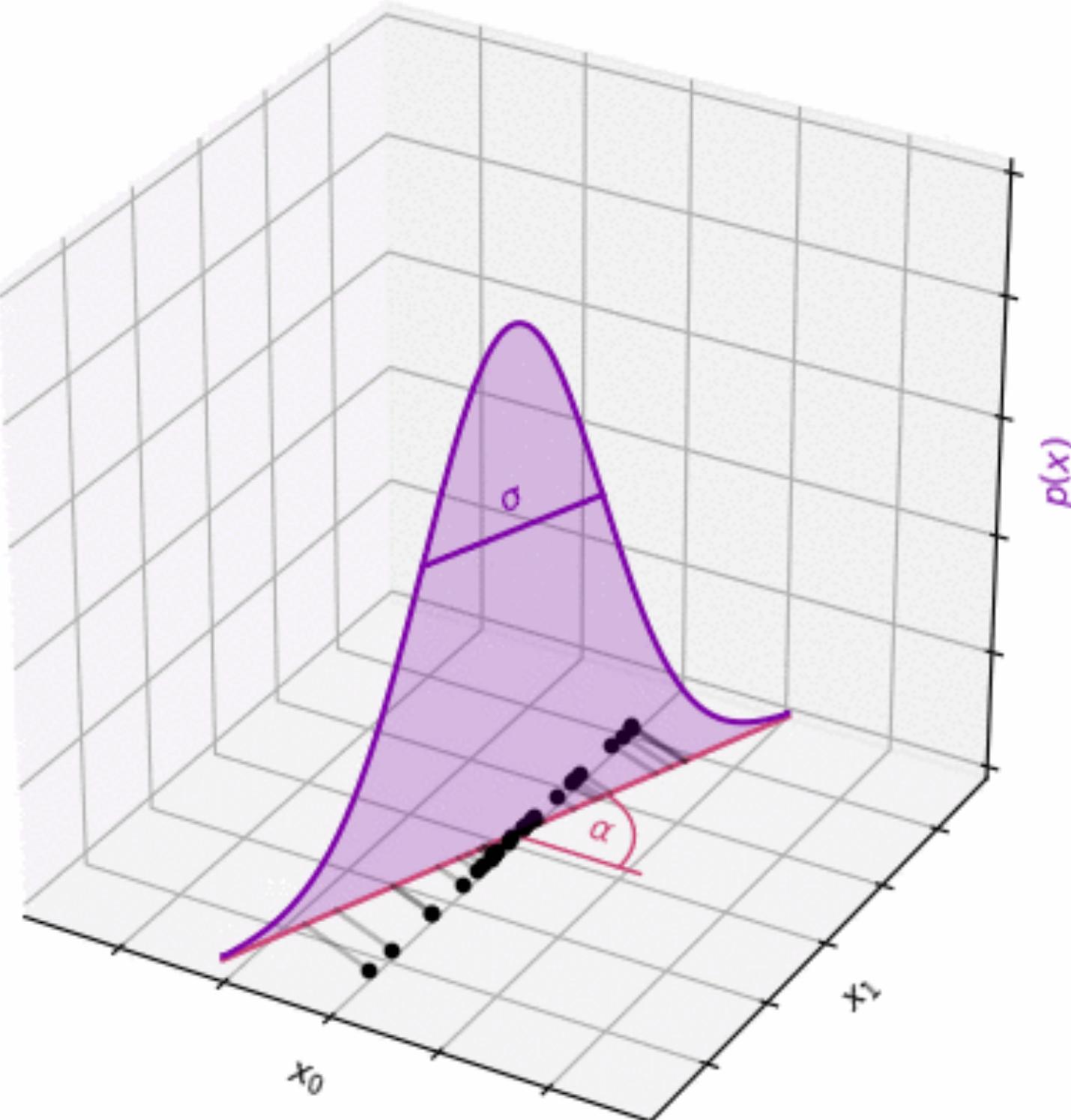
⇒ Learning  $\phi_f$  by maximum  
likelihood is unstable

# Maximum likelihood is not enough

Likelihood defined after projection to  $\mathcal{M}$ ,  
which is defined through NN weights  $\phi_f$

Family of likelihoods  $p_{\phi_f}(x|\phi_h)$   
rather than one likelihood  $p(x|\phi_f, \phi_h)$

⇒ Learning  $\phi_f$  by maximum  
likelihood is unstable

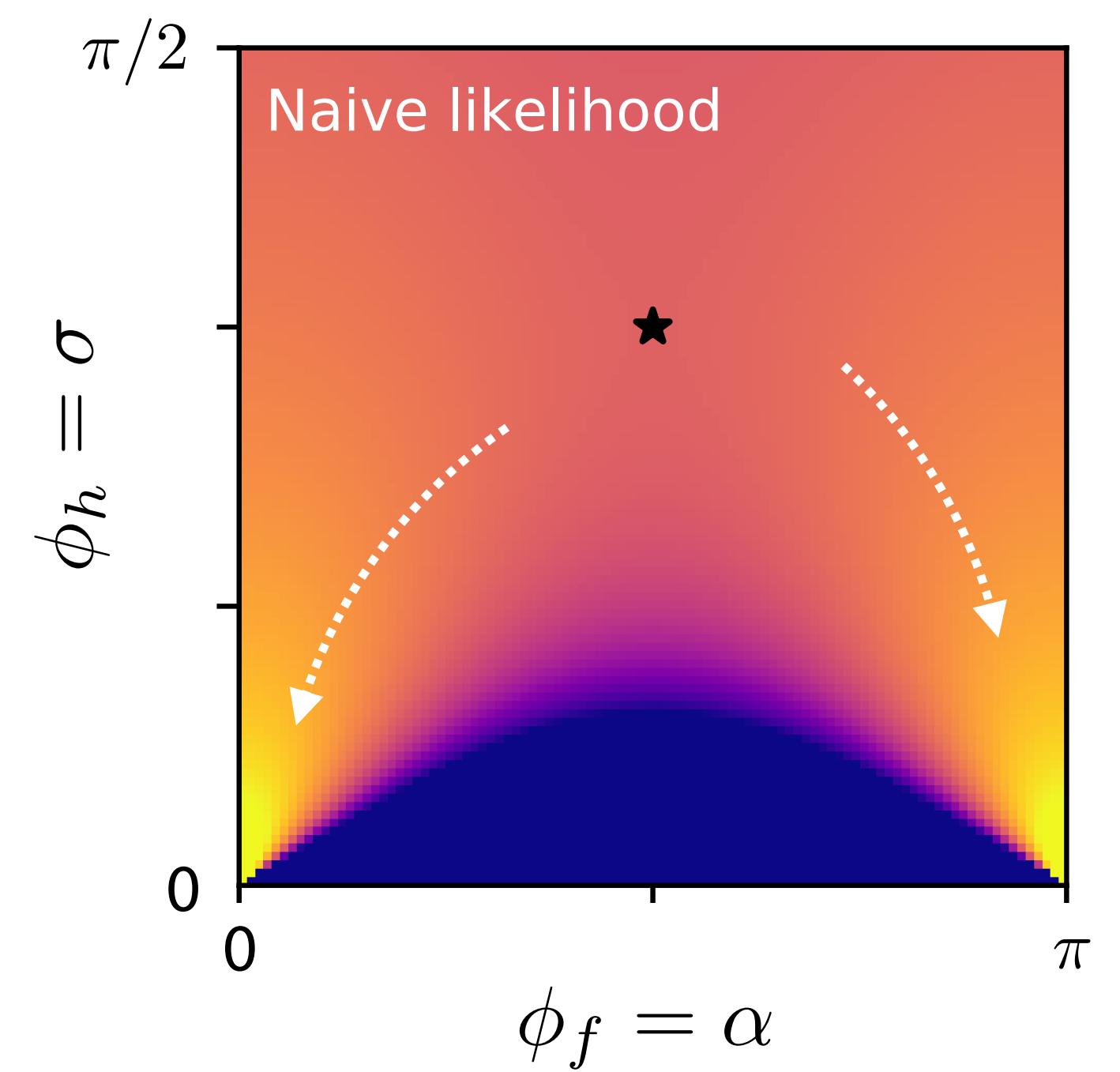
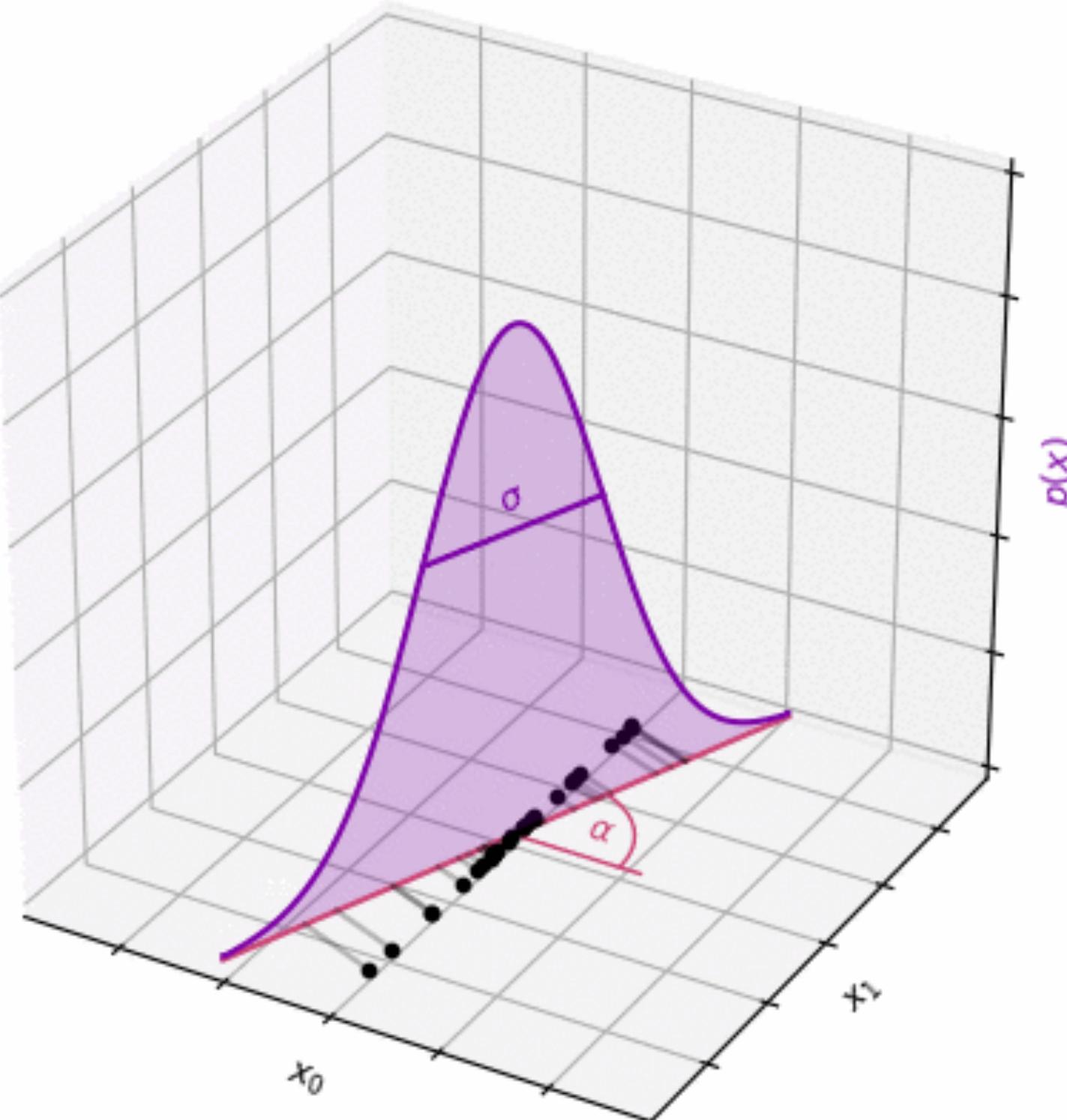


# Maximum likelihood is not enough

Likelihood defined after projection to  $\mathcal{M}$ ,  
which is defined through NN weights  $\phi_f$

Family of likelihoods  $p_{\phi_f}(x|\phi_h)$   
rather than one likelihood  $p(x|\phi_f, \phi_h)$

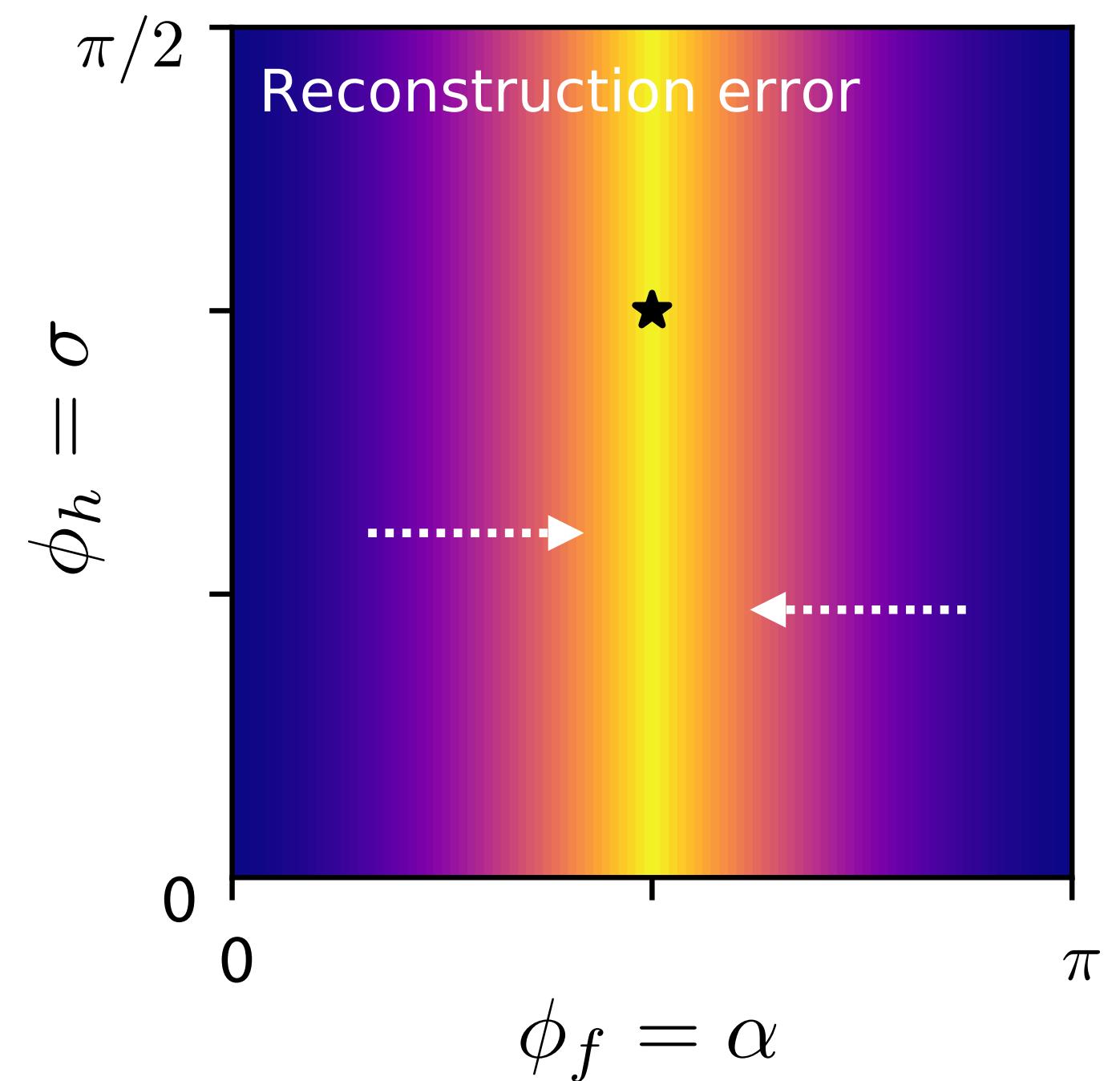
⇒ Learning  $\phi_f$  by maximum  
likelihood is unstable



# M/D training

Solution: separate training in two phases!

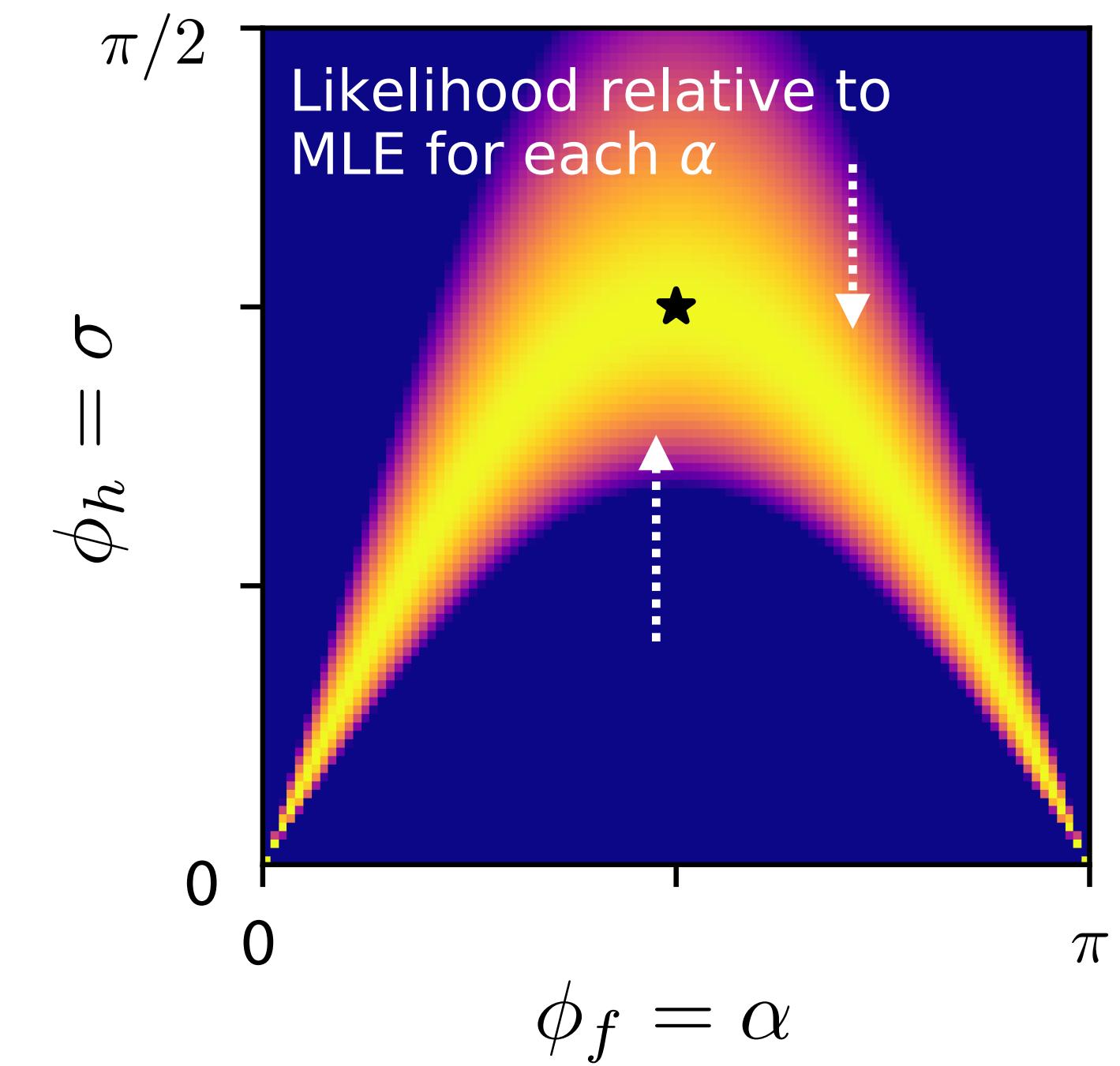
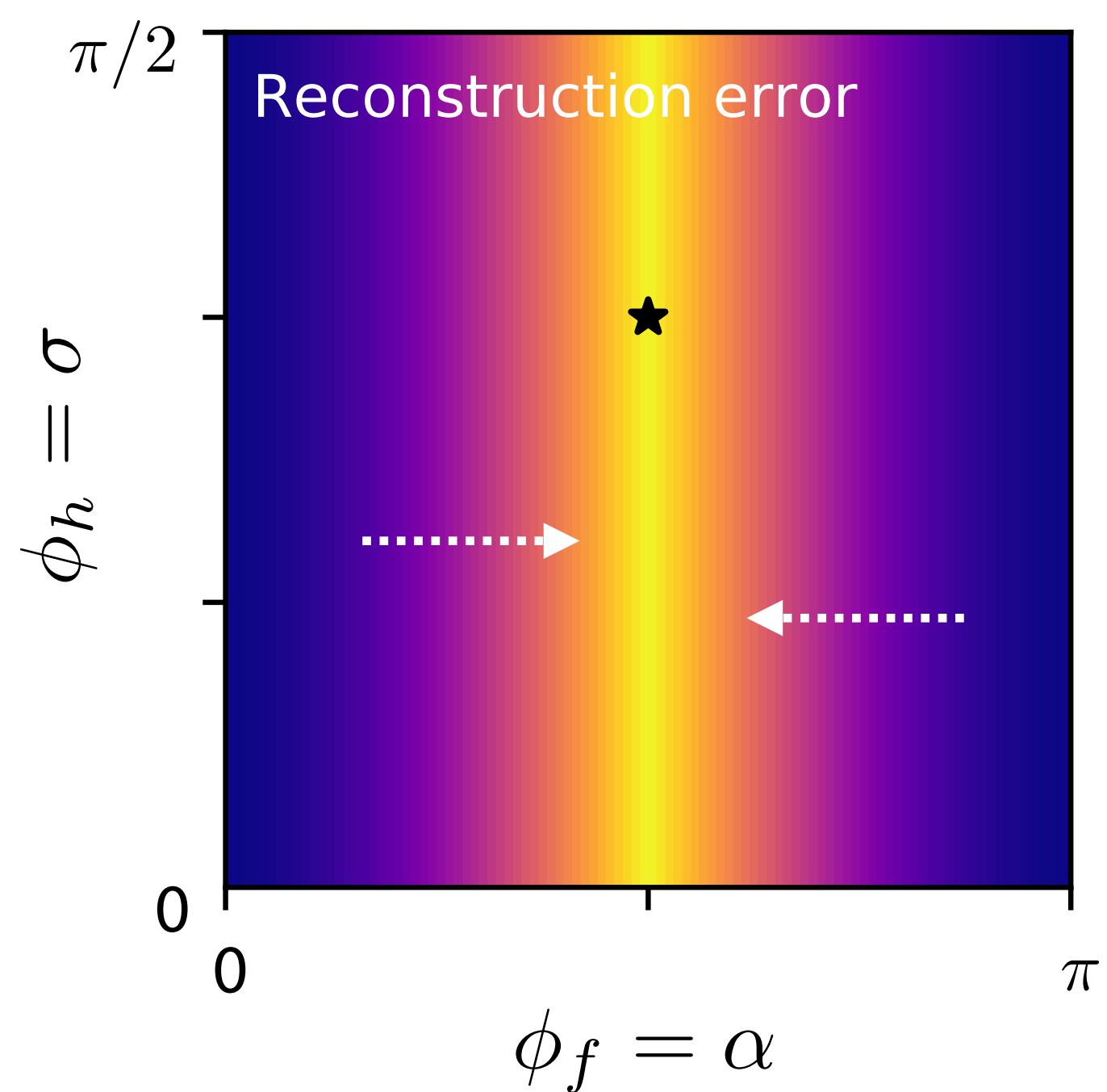
- **Manifold phase:**  
update  $\phi_f$  (and thus  $\mathcal{M}$ ) by minimizing  $\|x - x'\|$



# M/D training

Solution: separate training in two phases!

- **Manifold phase:**  
update  $\phi_f$  (and thus  $\mathcal{M}$ ) by minimizing  $\|x - x'\|$
- **Density phase:**  
update  $\phi_h$  (and thus  $p_{\mathcal{M}}(x)$ ) by maximum likelihood  
(keeping  $\mathcal{M}$  fixed)



# A second problem... and an accidental solution

The likelihood becomes expensive to evaluate for high-dimensional  $x$ :

$$\log p_{\mathcal{M}}(x) = \log p_{\tilde{u}}(h^{-1}(u)) - \log \det J_h(h^{-1}(u)) - \frac{1}{2} \log \det \left[ (\mathbf{1} \ 0) J_f^T(u) J_f(u) \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right]$$

Cannot separate determinant of  
product of non-square matrices

# A second problem... and an accidental solution

The likelihood becomes expensive to evaluate for high-dimensional  $x$ :

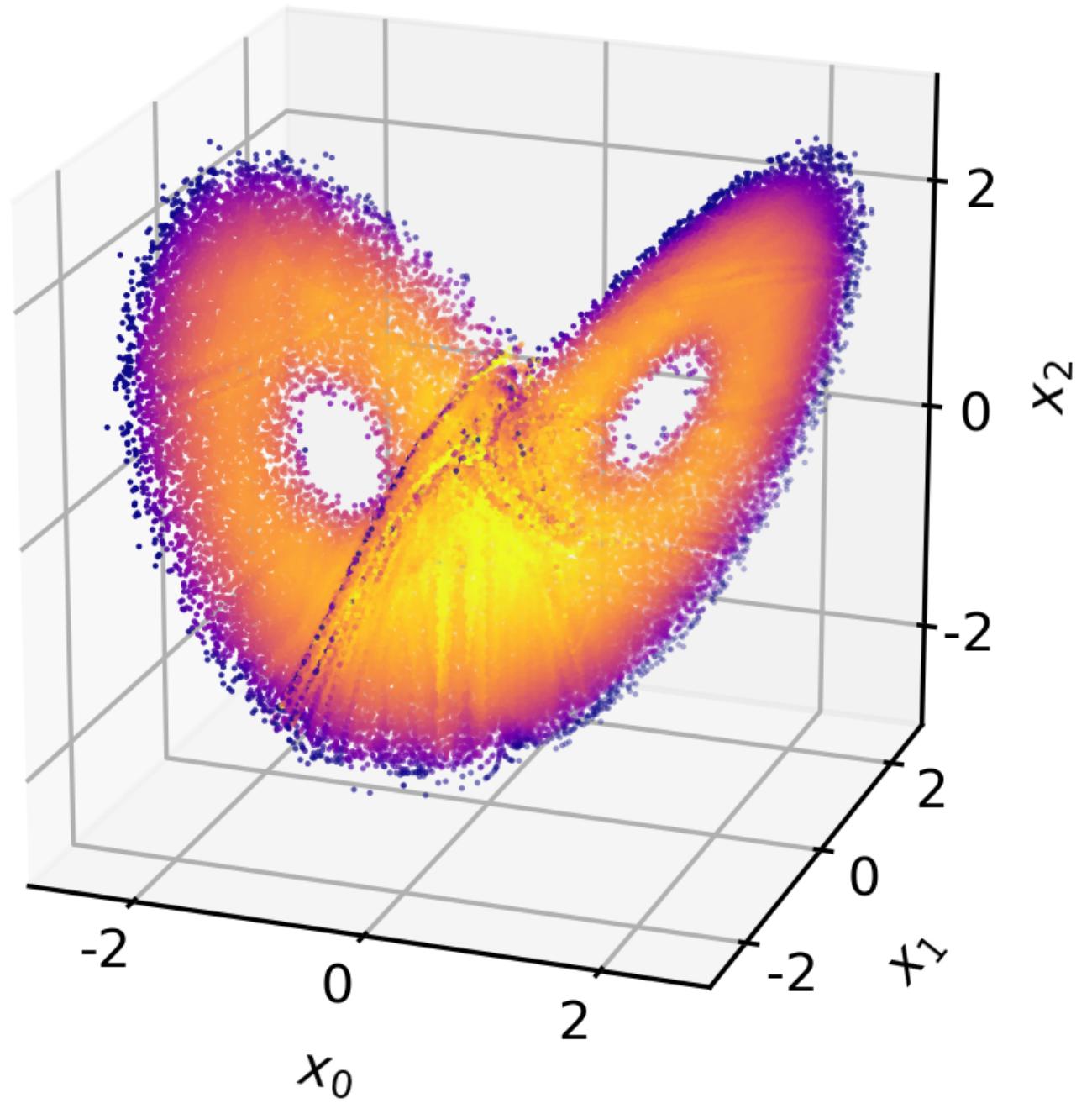
$$\log p_{\mathcal{M}}(x) = \log p_{\tilde{u}}(h^{-1}(u)) - \log \det J_h(h^{-1}(u)) - \frac{1}{2} \log \det \left[ (\mathbb{1} \ 0) J_f^T(u) J_f(u) \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right]$$

Cannot separate determinant of product of non-square matrices

M/D training sidesteps this problem: density phase only requires gradient

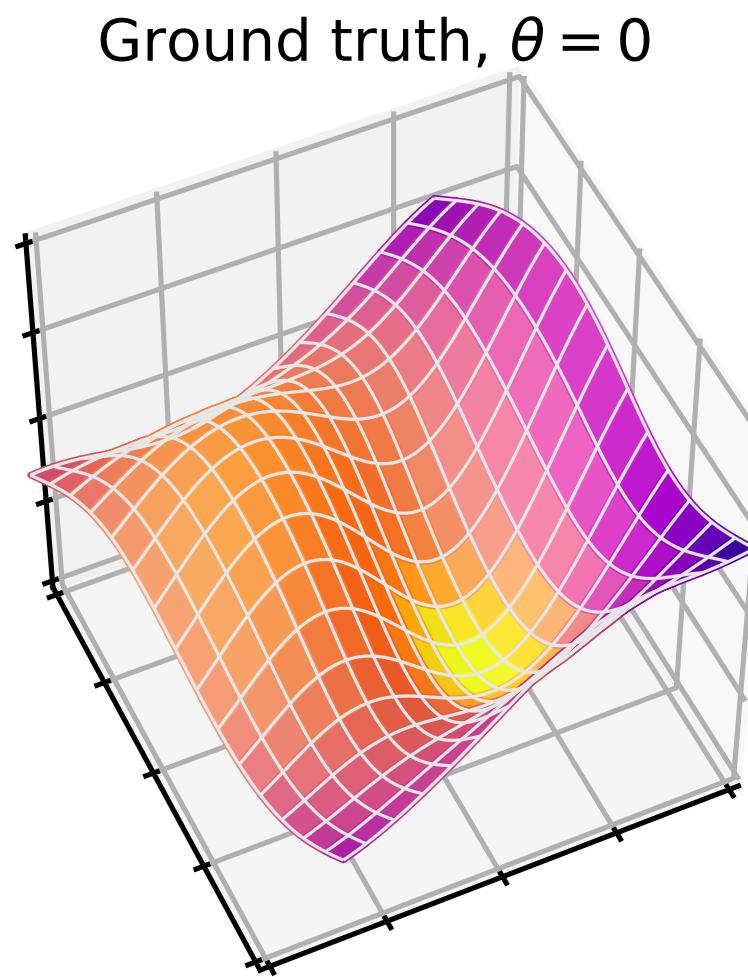
$$\nabla_{\phi_h} (\log p_{\mathcal{M}}(x)) = \nabla_{\phi_h} (\log p_{\tilde{u}}(h^{-1}(u))) - \nabla_{\phi_h} (\log \det J_h(h^{-1}(u))) - \underbrace{\nabla_{\phi_h} \frac{1}{2} \log \det \left[ (\mathbb{1} \ 0) J_f^T(u) J_f(u) \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right]}_{=0},$$

which can be computed efficiently!



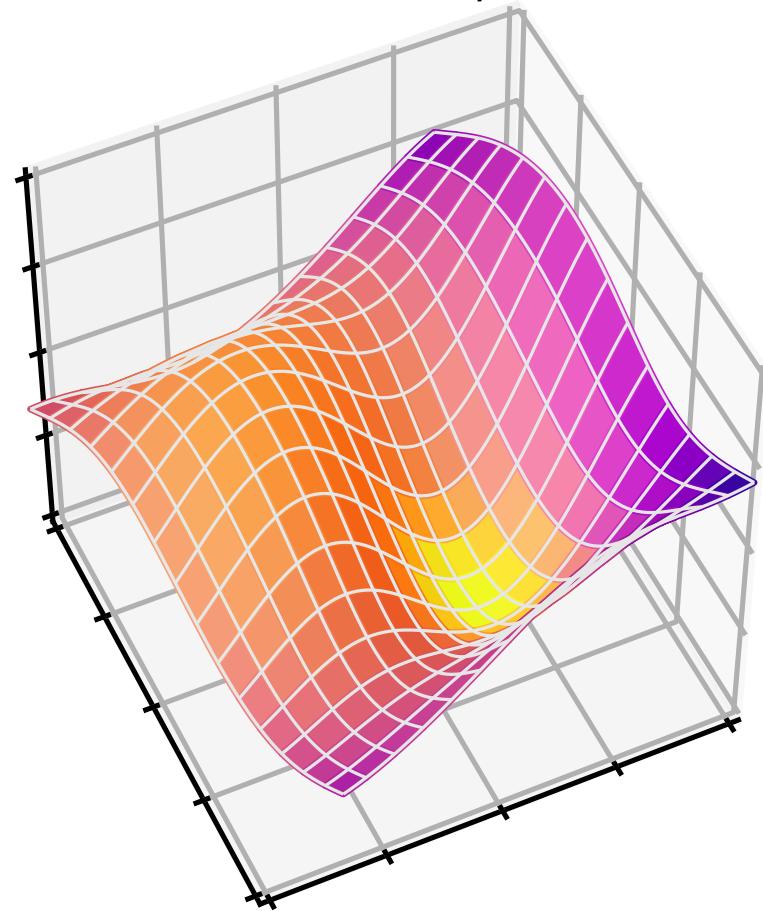
# Experiments

# Mixture model on a polynomial surface

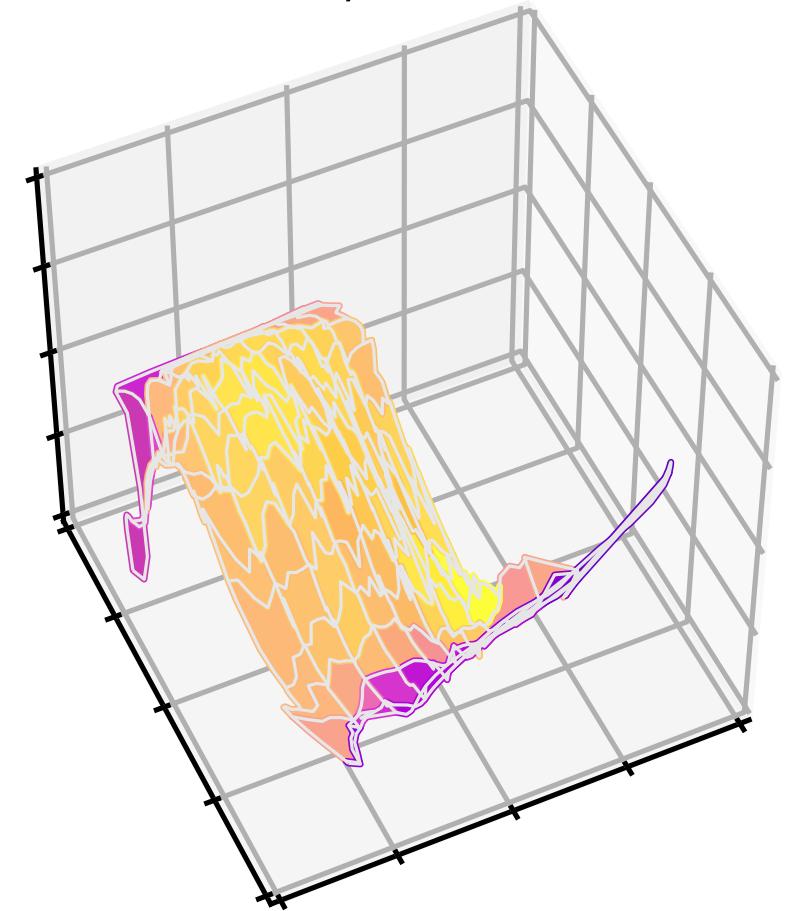


# Mixture model on a polynomial surface

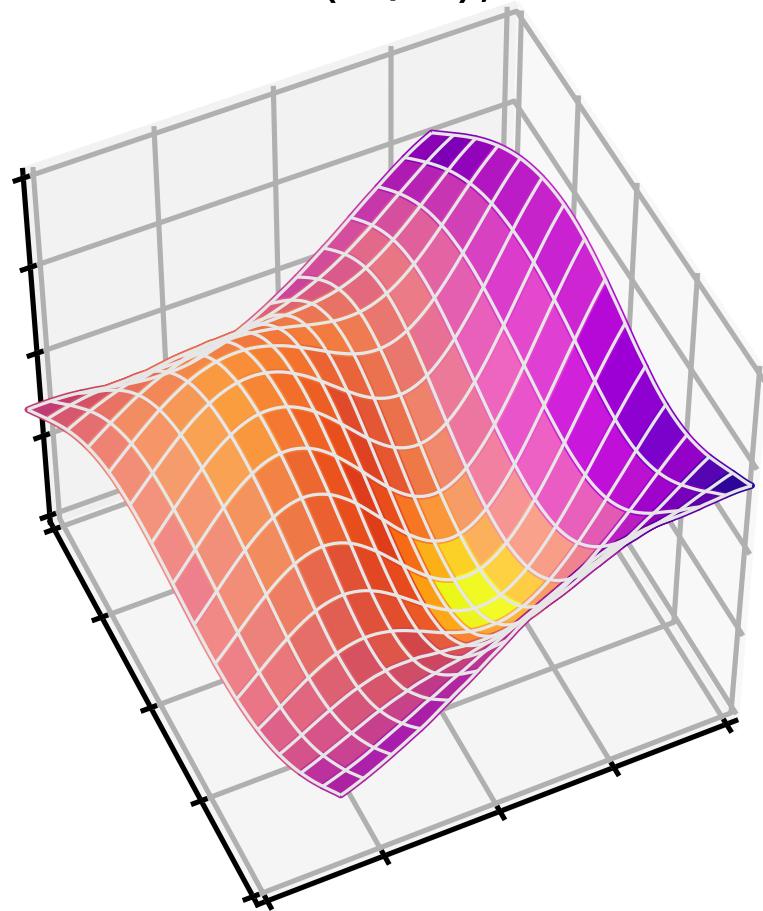
Ground truth,  $\theta = 0$



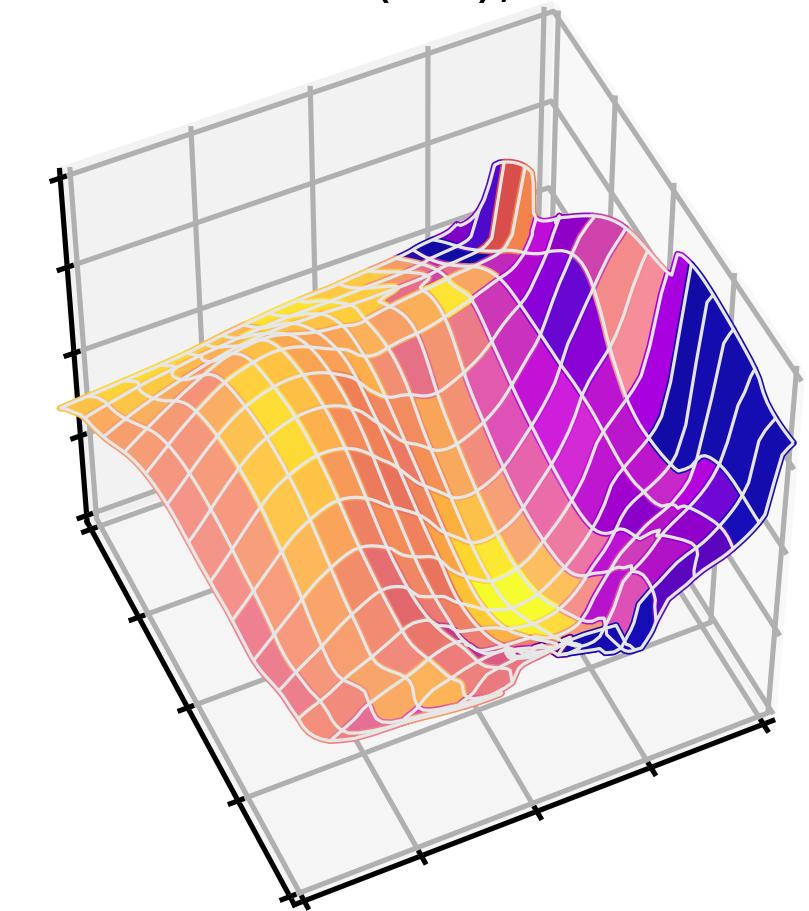
PIE,  $\theta = 0$



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

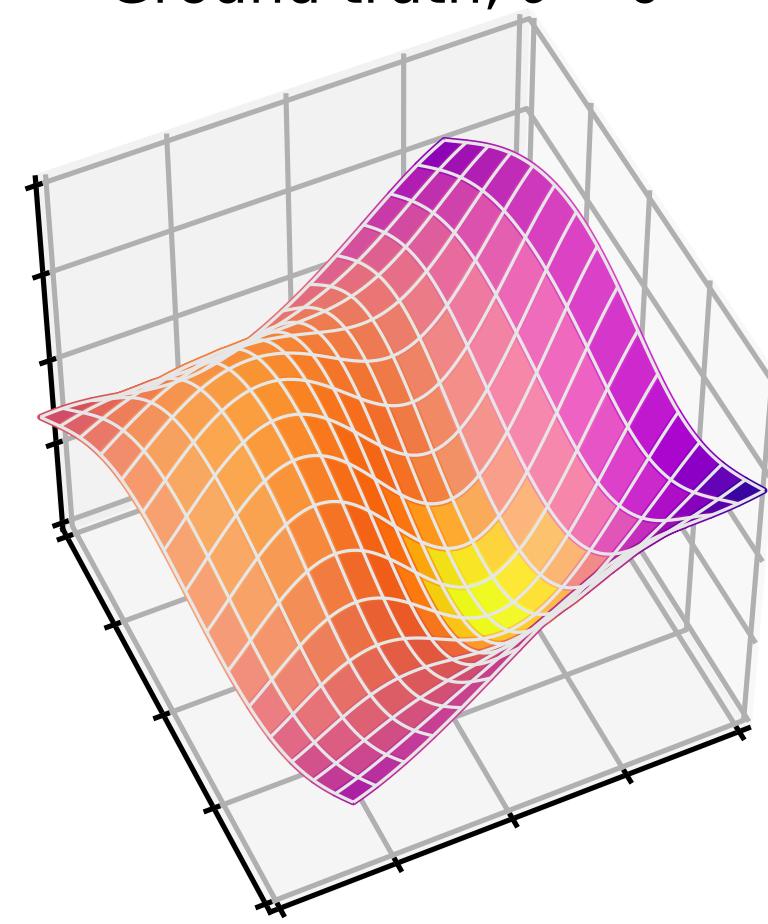


$\mathcal{M}$ -flow (OT),  $\theta = 0$

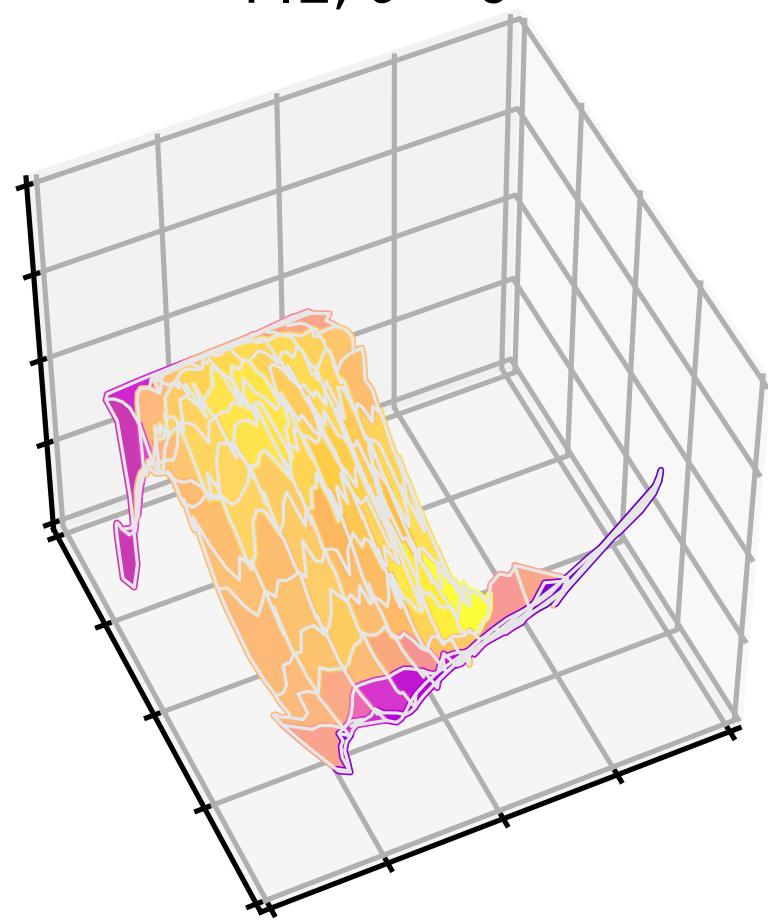


# Mixture model on a polynomial surface

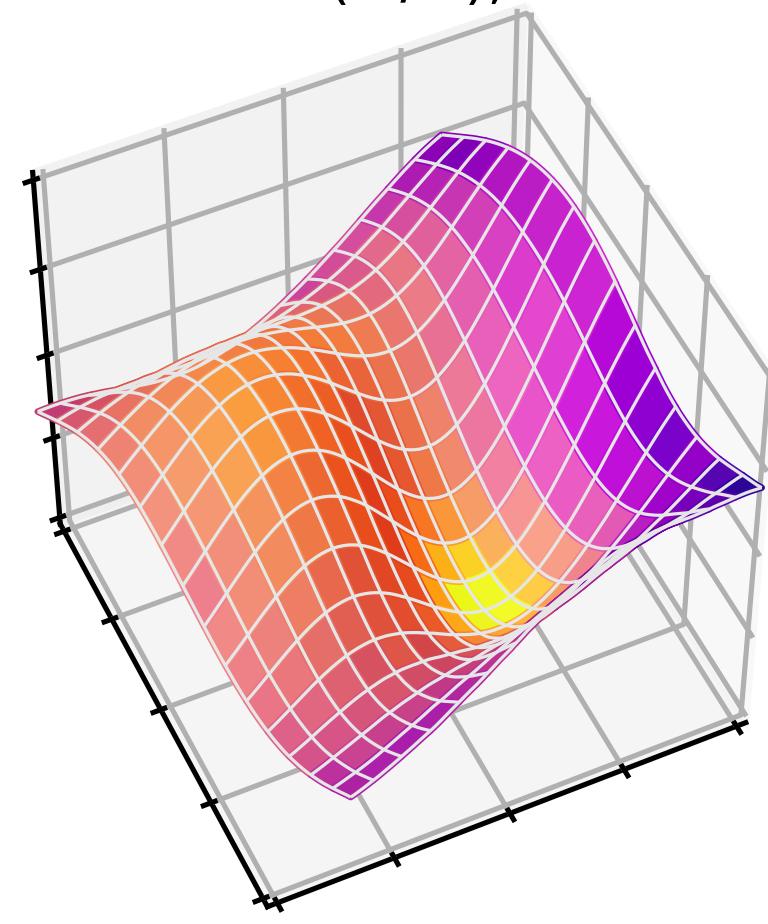
Ground truth,  $\theta = 0$



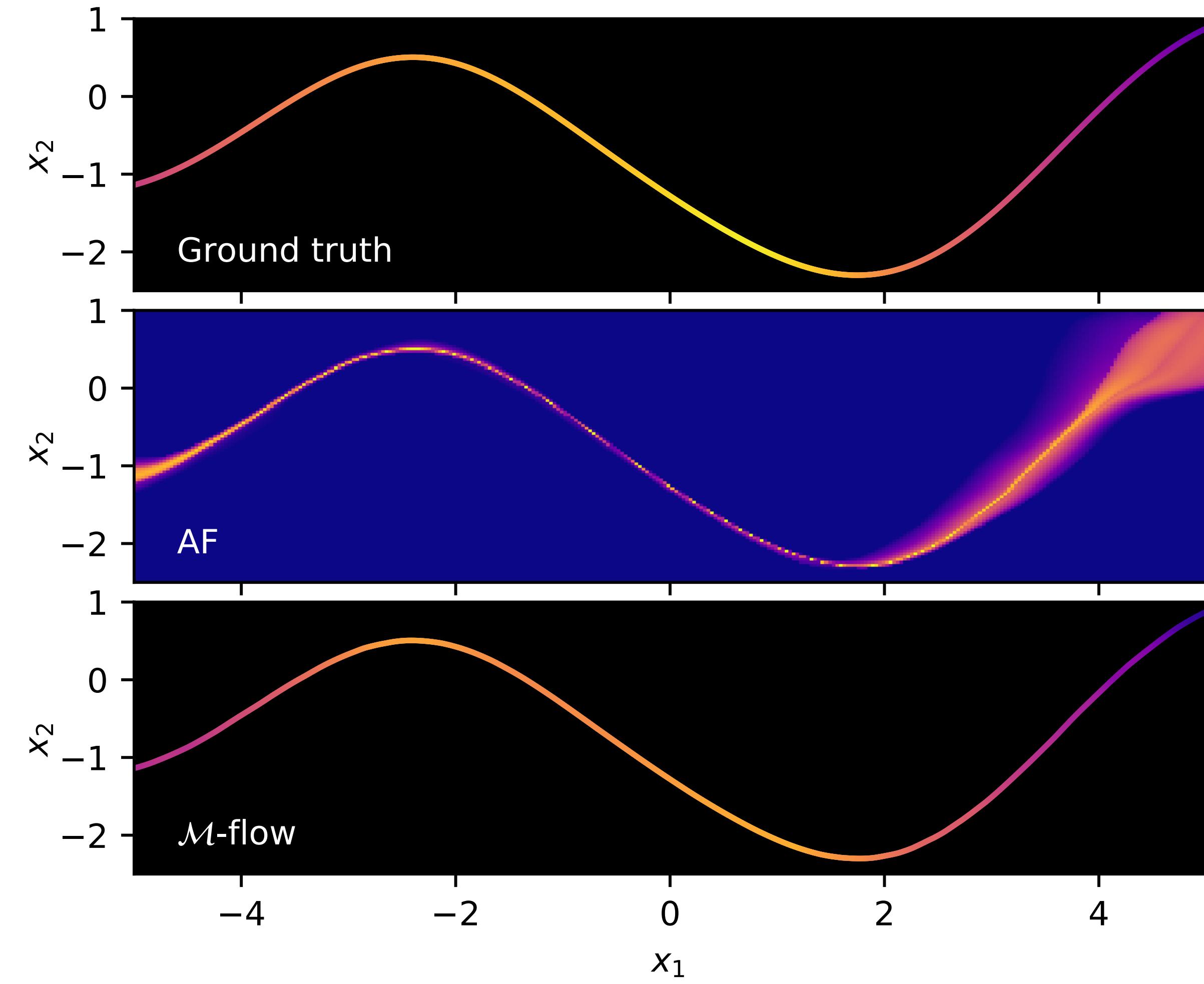
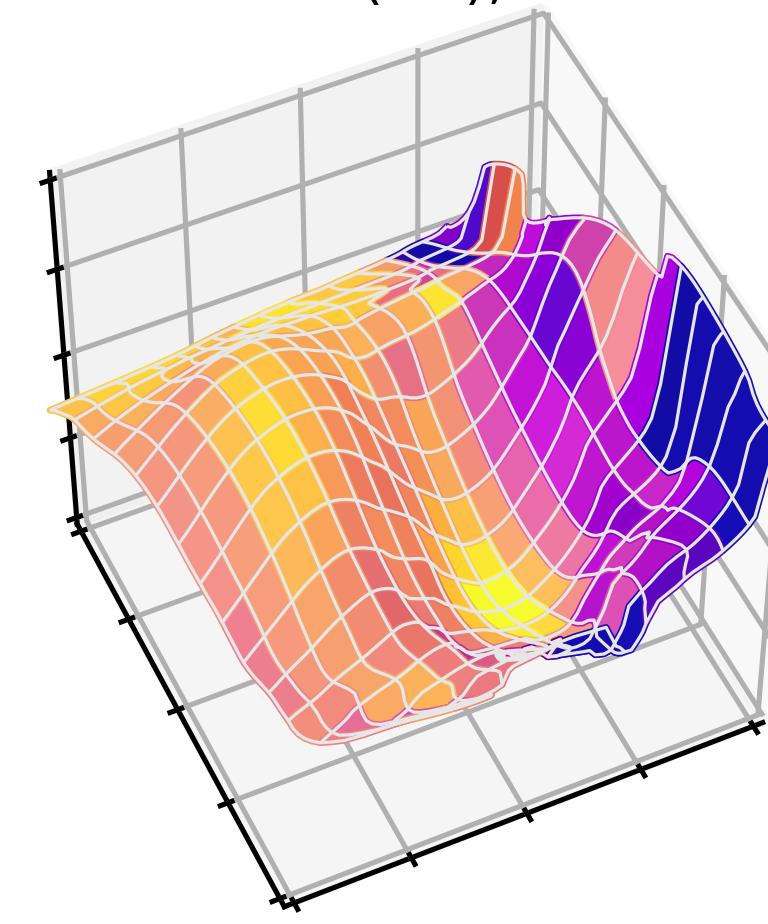
PIE,  $\theta = 0$



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

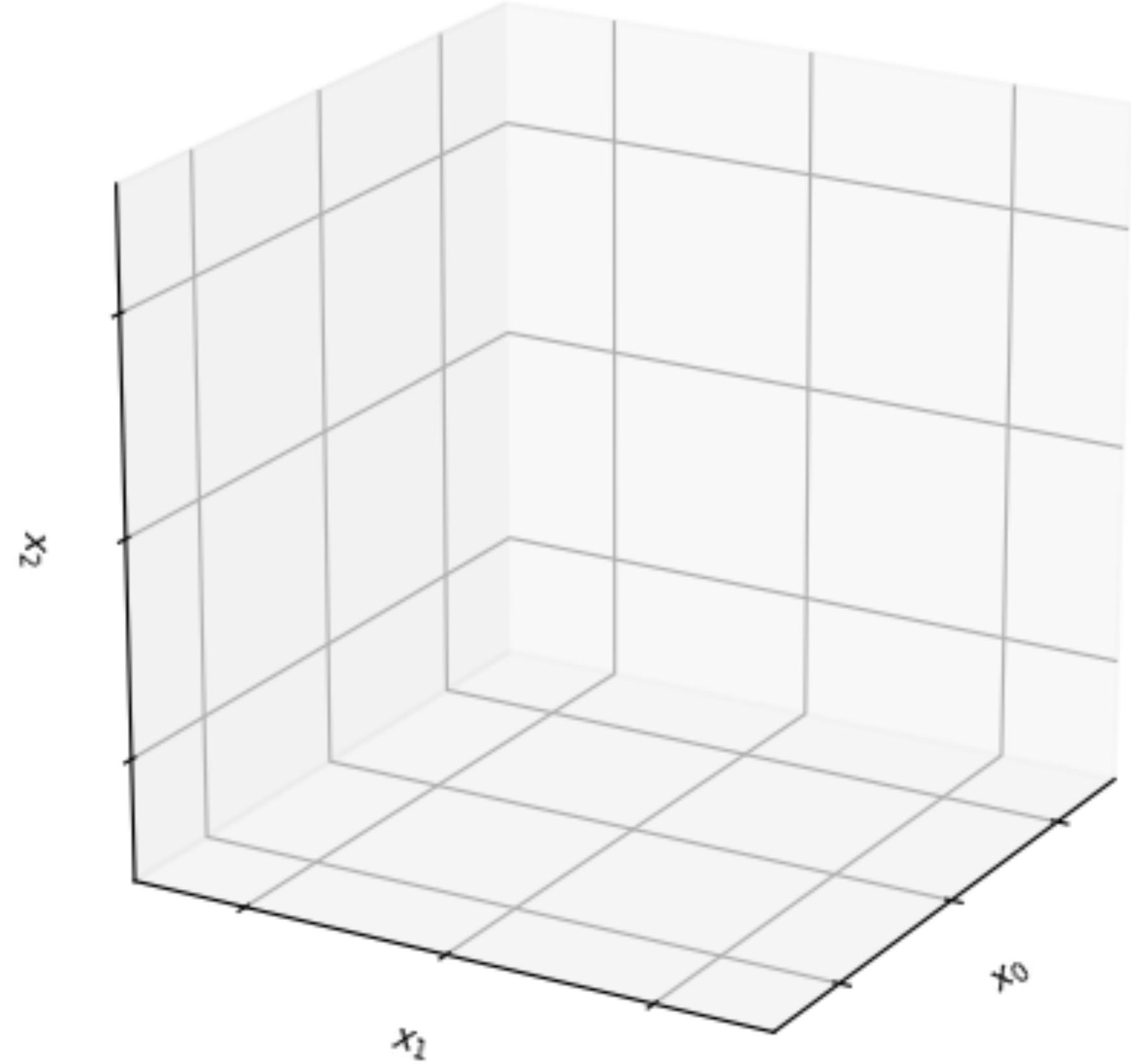


$\mathcal{M}$ -flow (OT),  $\theta = 0$



# Lorenz attractor

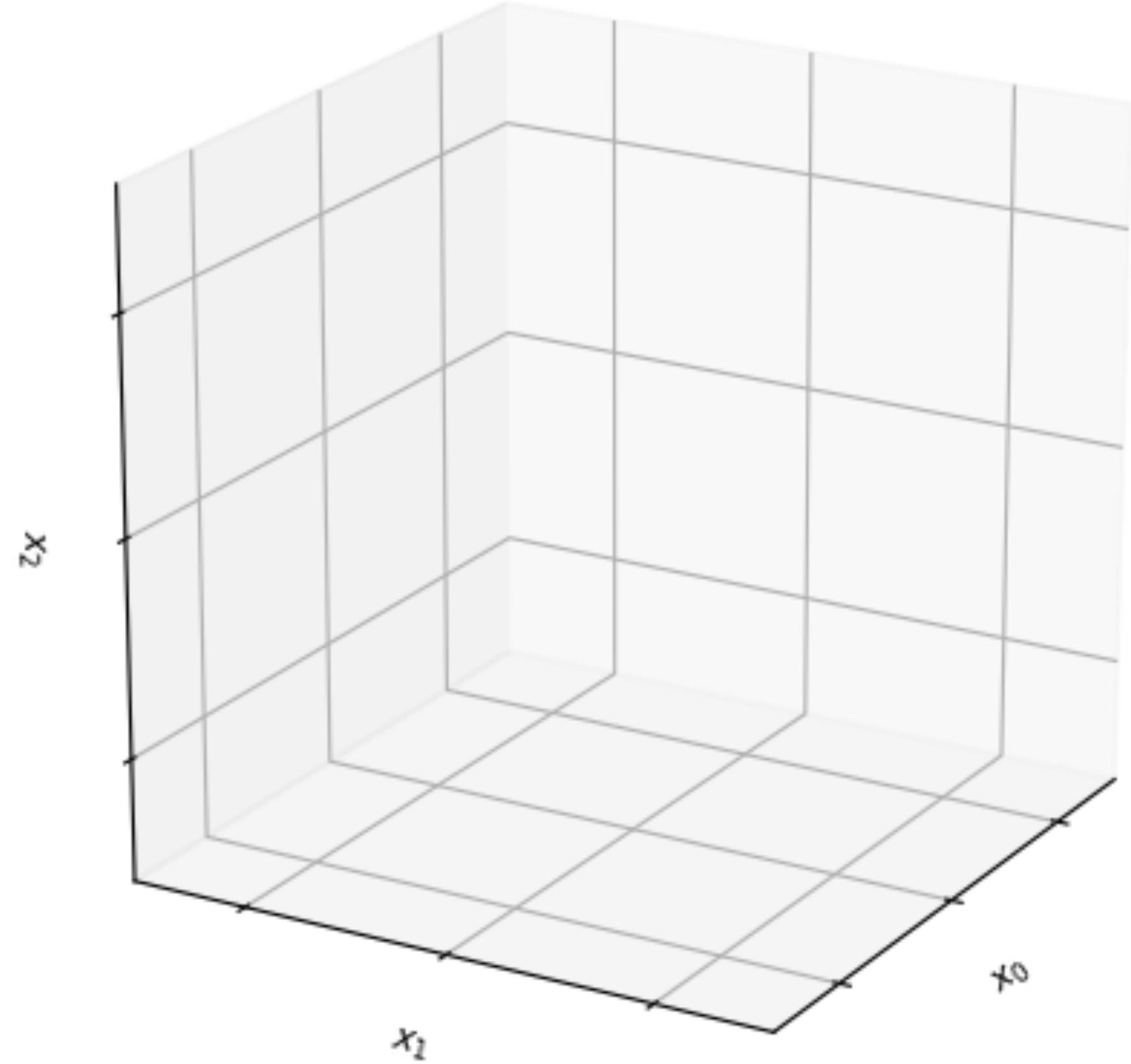
[E. Lorenz 1963]



$$\frac{dx_0}{dt} = \sigma(x_1 - x_0), \quad \frac{dx_1}{dt} = x_0(\rho - x_2) - x_1, \quad \frac{dx_2}{dt} = x_0x_1 - \beta x_2.$$

# Lorenz attractor

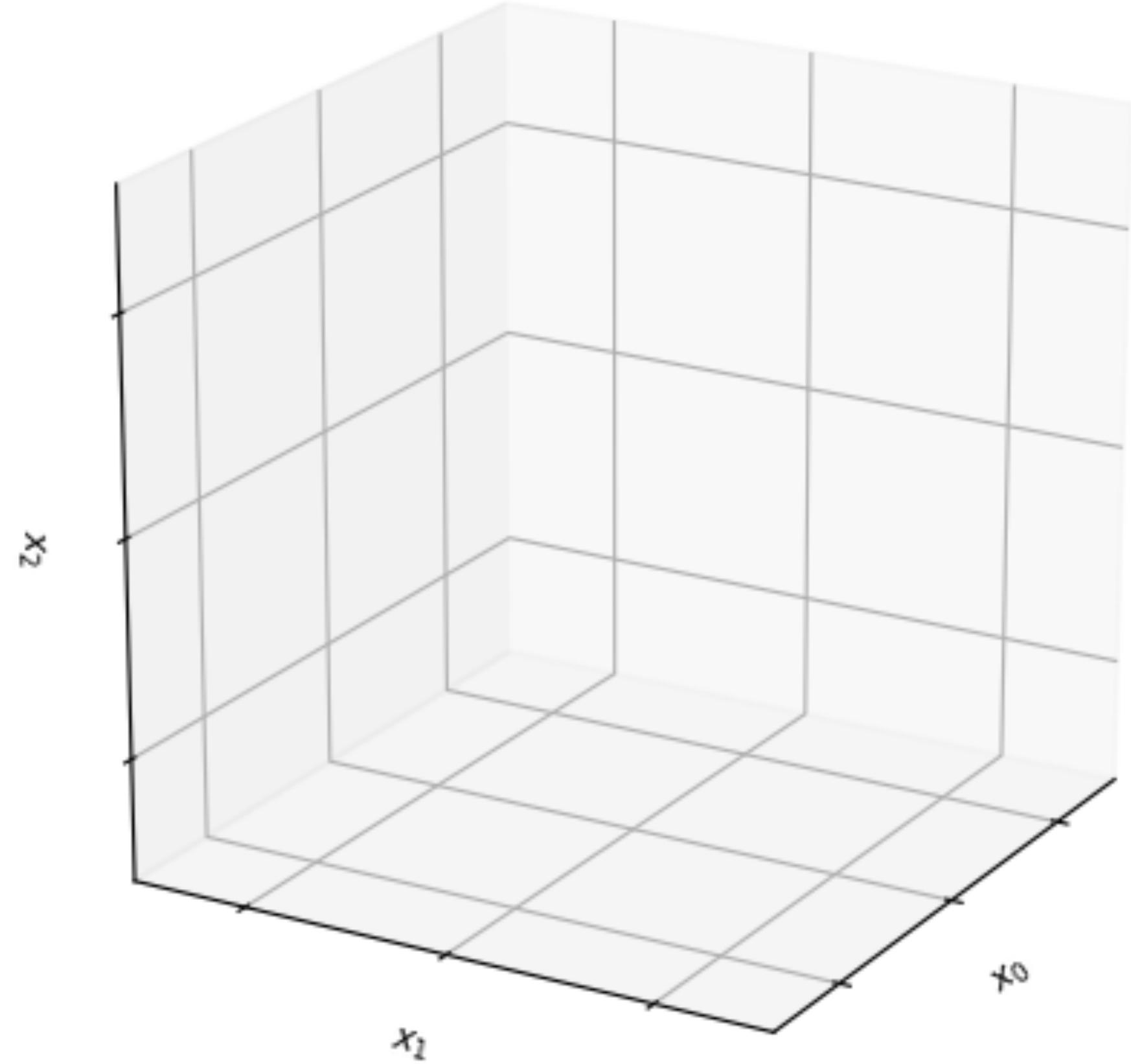
[E. Lorenz 1963]



$$\frac{dx_0}{dt} = \sigma(x_1 - x_0), \quad \frac{dx_1}{dt} = x_0(\rho - x_2) - x_1, \quad \frac{dx_2}{dt} = x_0x_1 - \beta x_2.$$

# Lorenz attractor

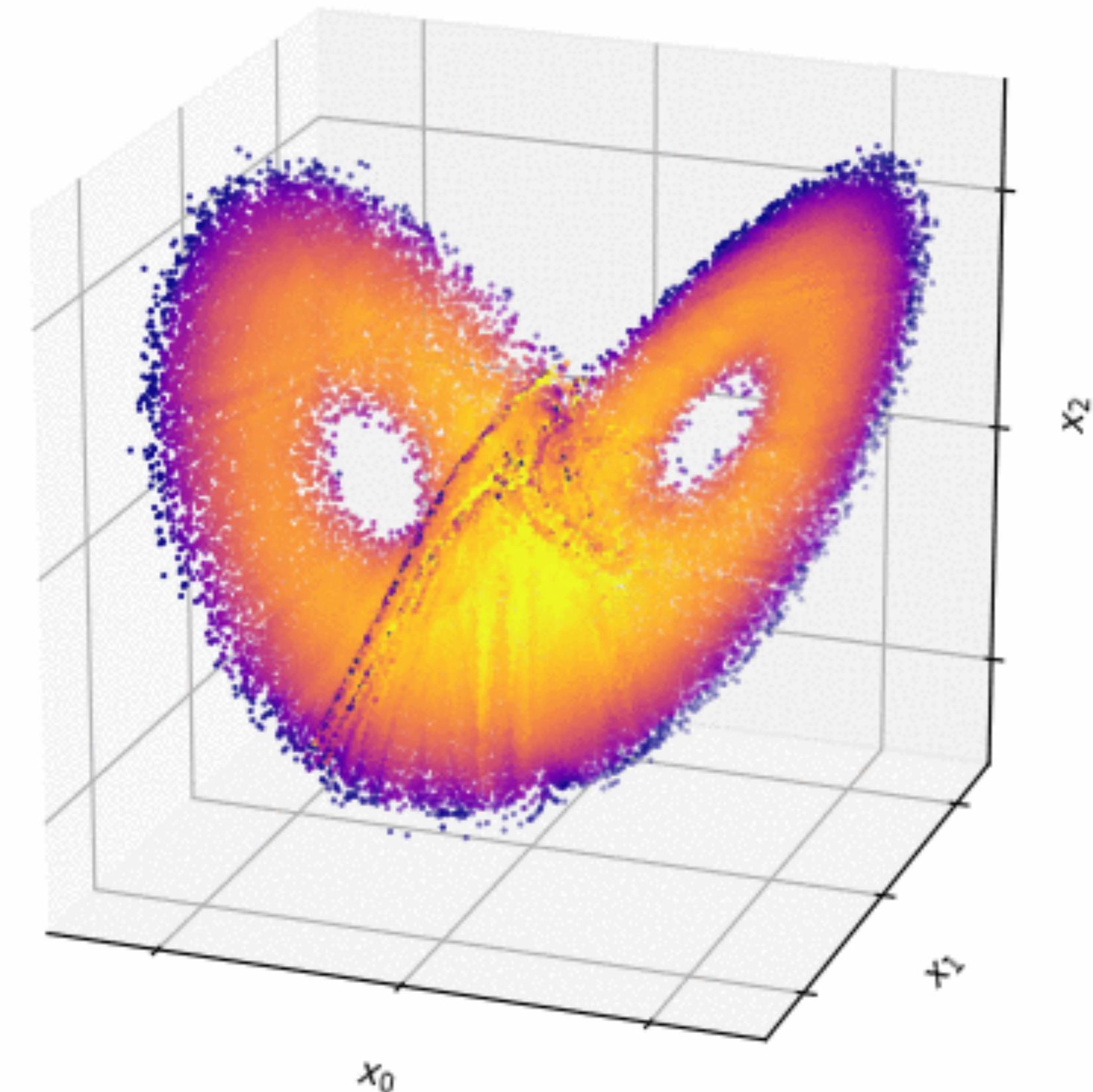
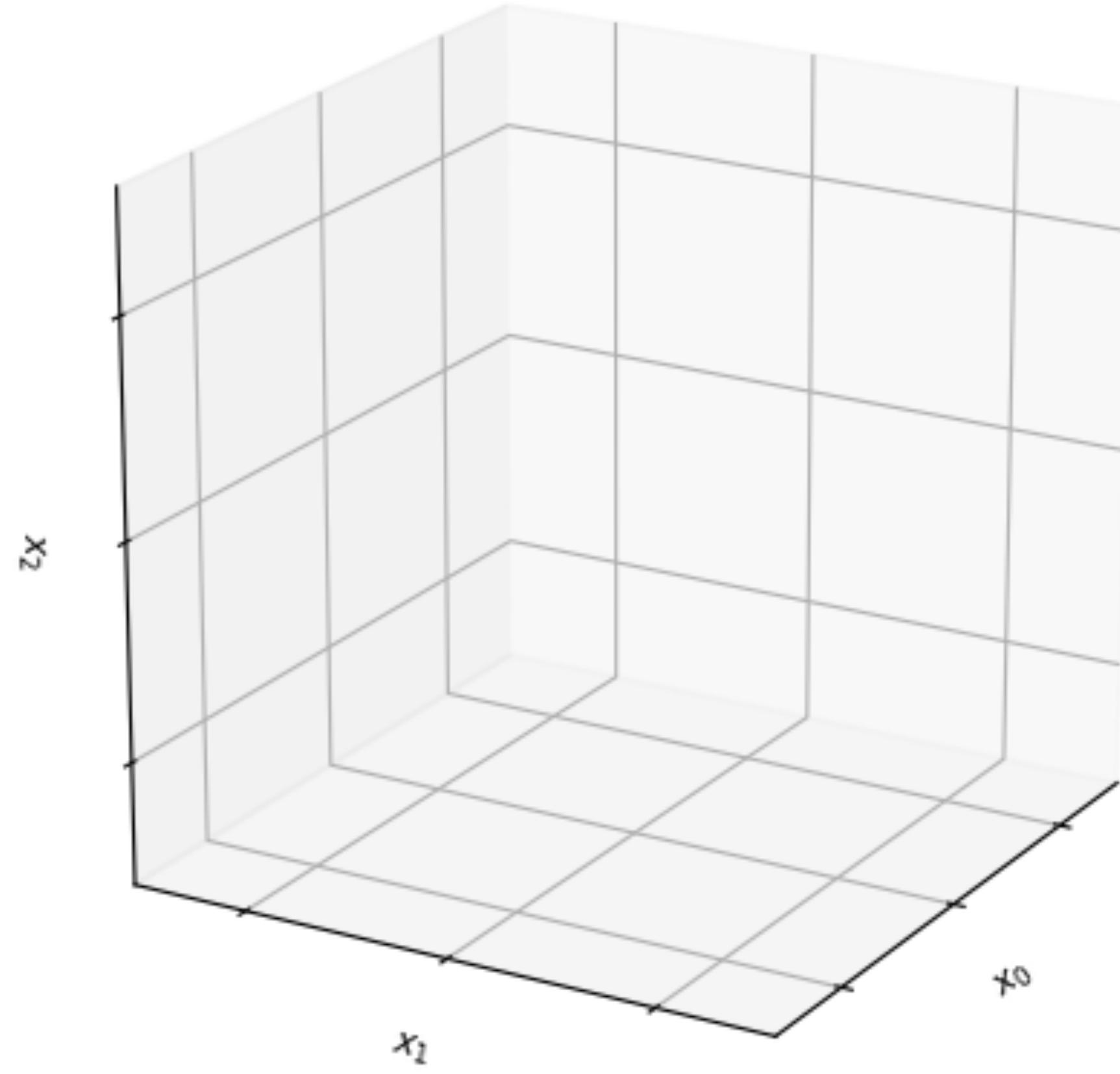
[E. Lorenz 1963]



$$\frac{dx_0}{dt} = \sigma(x_1 - x_0), \quad \frac{dx_1}{dt} = x_0(\rho - x_2) - x_1, \quad \frac{dx_2}{dt} = x_0x_1 - \beta x_2.$$

# Lorenz attractor

[E. Lorenz 1963]

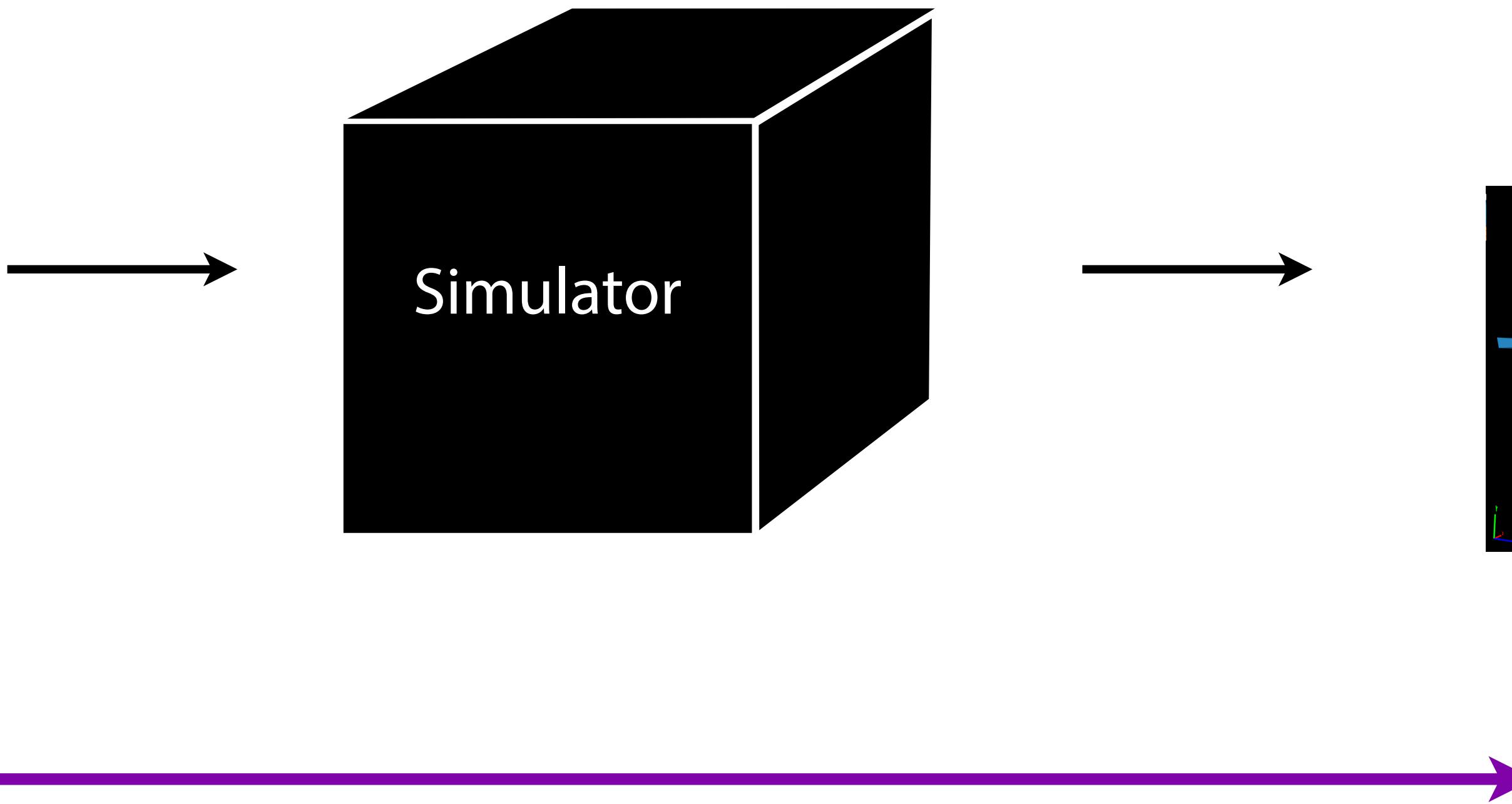


$$\frac{dx_0}{dt} = \sigma(x_1 - x_0), \quad \frac{dx_1}{dt} = x_0(\rho - x_2) - x_1, \quad \frac{dx_2}{dt} = x_0x_1 - \beta x_2.$$

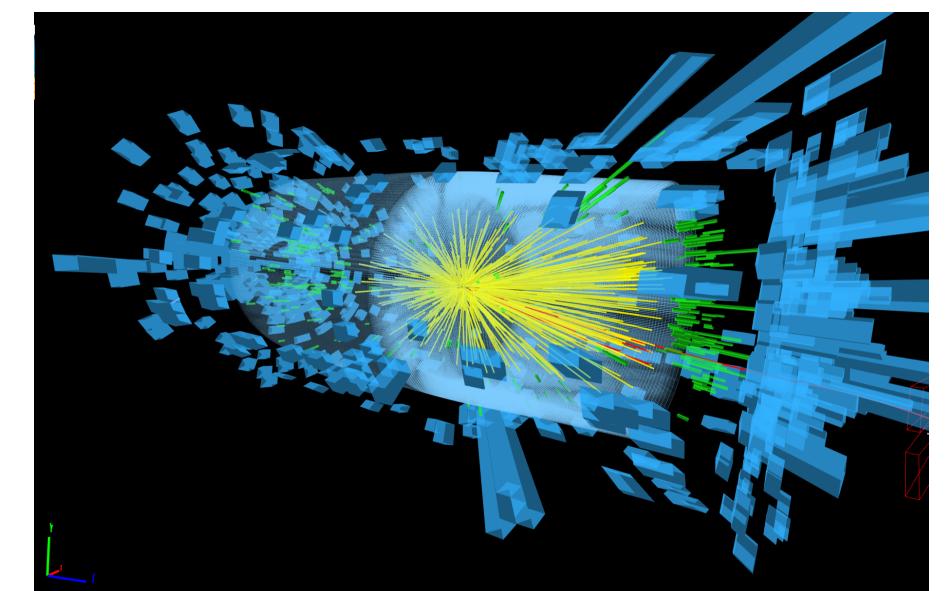
# Particle physics as a likelihood-free inference problem

$\mathcal{O}(10)$  parameters  $\theta$

$$S = \int d^4x \left[ \mathcal{L}_{\text{SM}} + \frac{f_{\phi,2}}{\Lambda^2} \frac{1}{2} \partial_\mu (\phi^\dagger \phi) \partial^\mu (\phi^\dagger \phi) + \frac{f_{\phi,3}}{\Lambda^2} \frac{1}{3} (\phi^\dagger \phi)^3 \right. \\ + \frac{f_{GG}}{\Lambda^2} (\phi^\dagger \phi) G_{\mu\nu}^a G^{\mu\nu a} - \frac{f_{BB}}{\Lambda^2} \frac{g'^2}{4} (\phi^\dagger \phi) B_{\mu\nu} B^{\mu\nu} - \frac{f_{WW}}{\Lambda^2} \frac{g^2}{4} (\phi^\dagger \phi) W_{\mu\nu}^a W^{\mu\nu a} \\ + \frac{f_B}{\Lambda^2} \frac{ig'}{2} (D^\mu \phi)^\dagger D^\nu \phi B_{\mu\nu} + \frac{f_W}{\Lambda^2} \frac{ig}{2} (D^\mu \phi)^\dagger \sigma^a D^\nu \phi W_{\mu\nu}^a \\ + \frac{f_\ell}{\Lambda^2} (\phi^\dagger \phi) \bar{L}_L \phi \ell_R + \frac{f_u}{\Lambda^2} (\phi^\dagger \phi) \bar{Q}_L \tilde{\phi} u_R + \frac{f_d}{\Lambda^2} (\phi^\dagger \phi) \bar{Q}_L \phi d_R \\ \left. + \frac{f_{G\widetilde{G}}}{\Lambda^2} (\phi^\dagger \phi) G_{\mu\nu}^a \widetilde{G}^{\mu\nu a} - \frac{f_{B\widetilde{B}}}{\Lambda^2} \frac{g'^2}{4} (\phi^\dagger \phi) B_{\mu\nu} \widetilde{B}^{\mu\nu} - \frac{f_{W\widetilde{W}}}{\Lambda^2} \frac{g^2}{4} (\phi^\dagger \phi) W_{\mu\nu}^a \widetilde{W}^{\mu\nu a} \right]$$



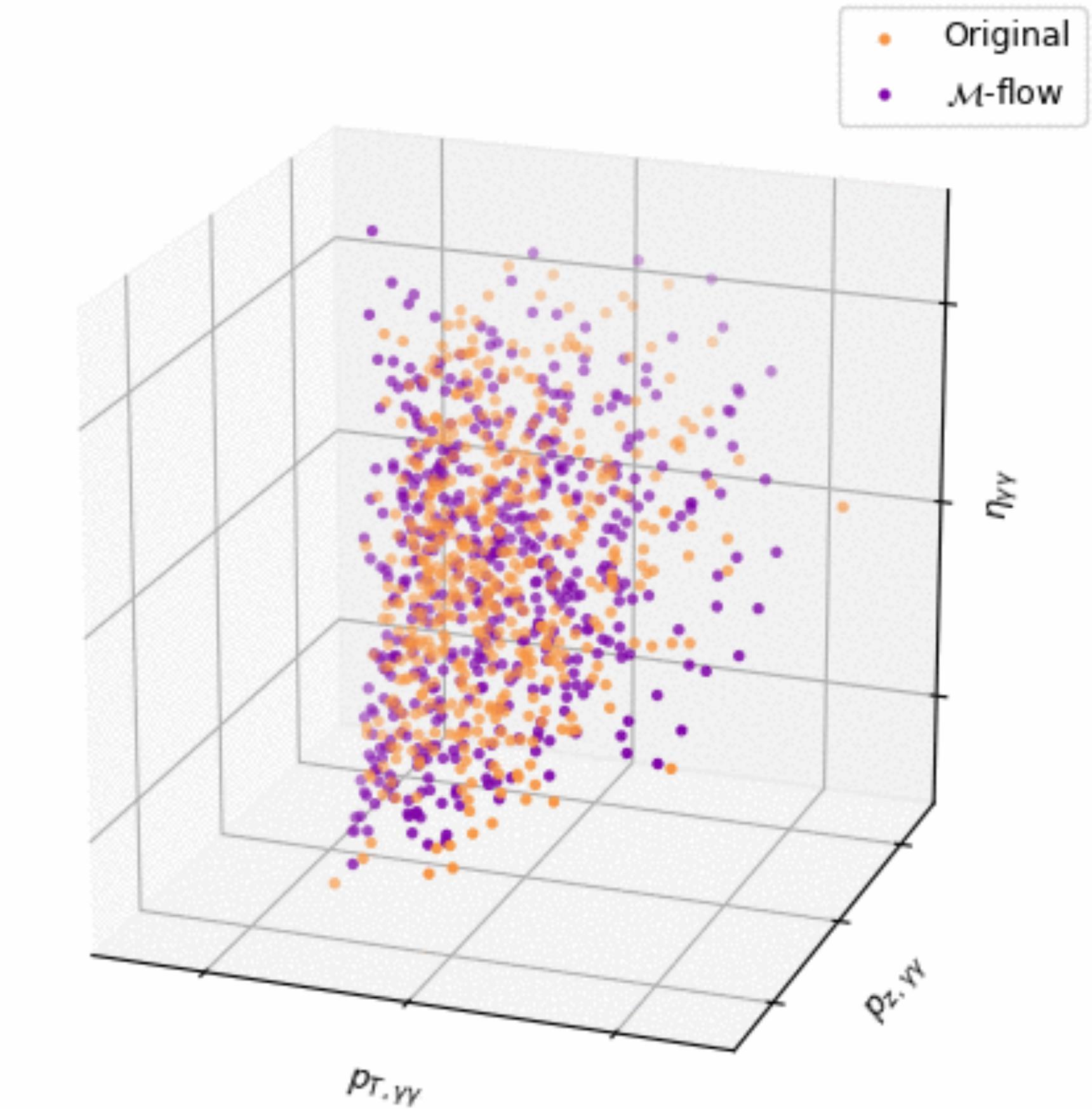
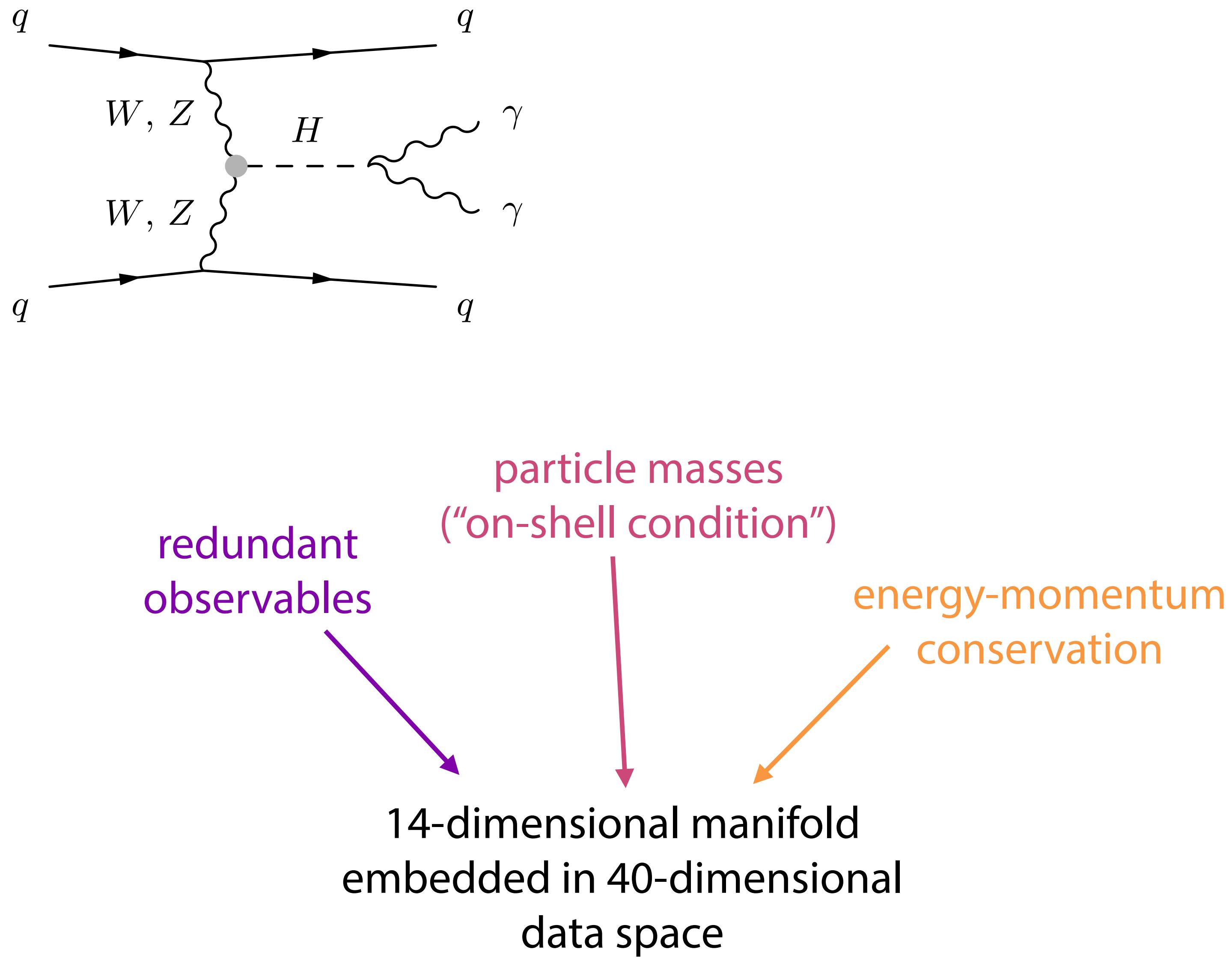
$\mathcal{O}(10 \dots 1000)$  observables  $x$



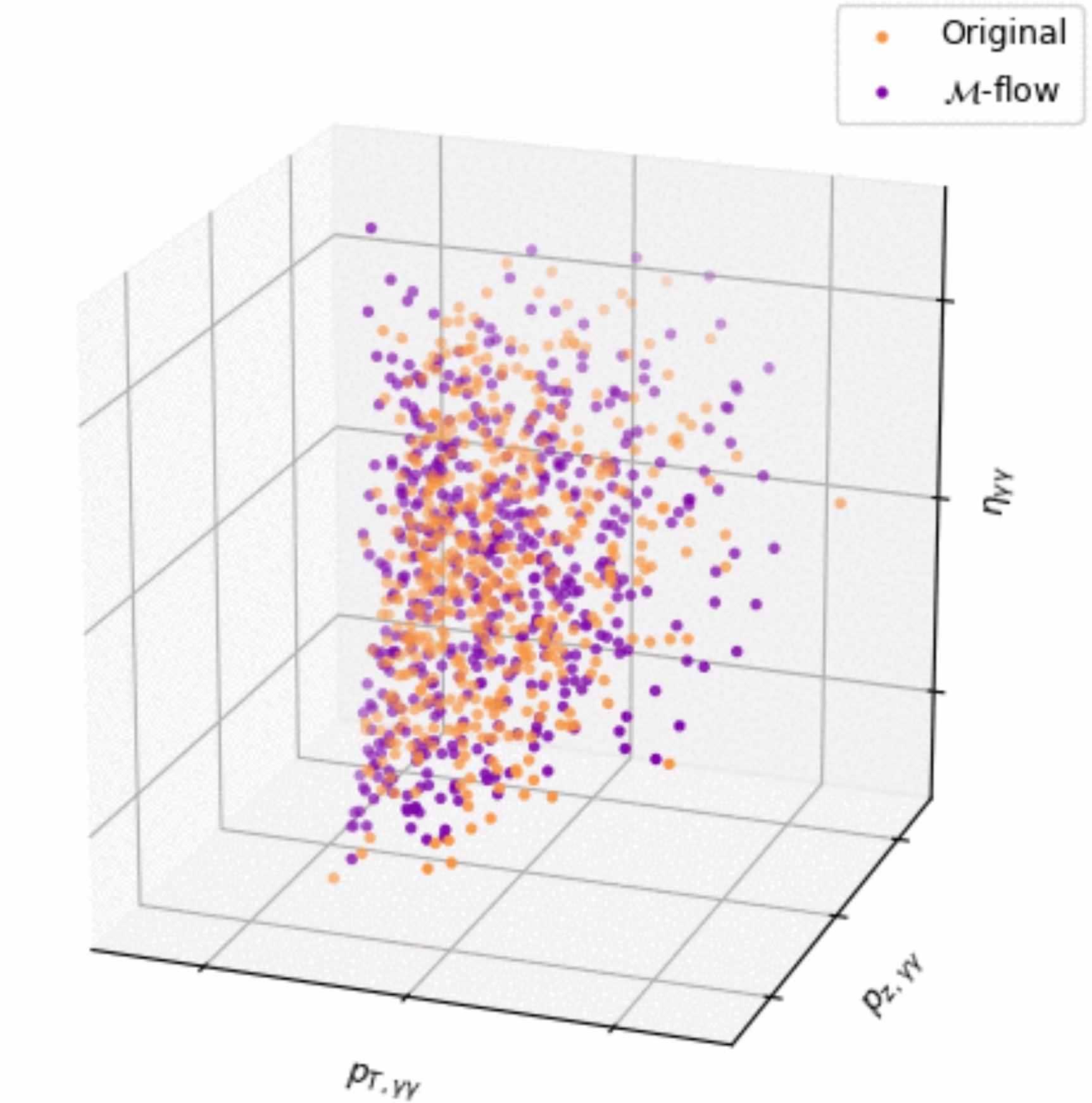
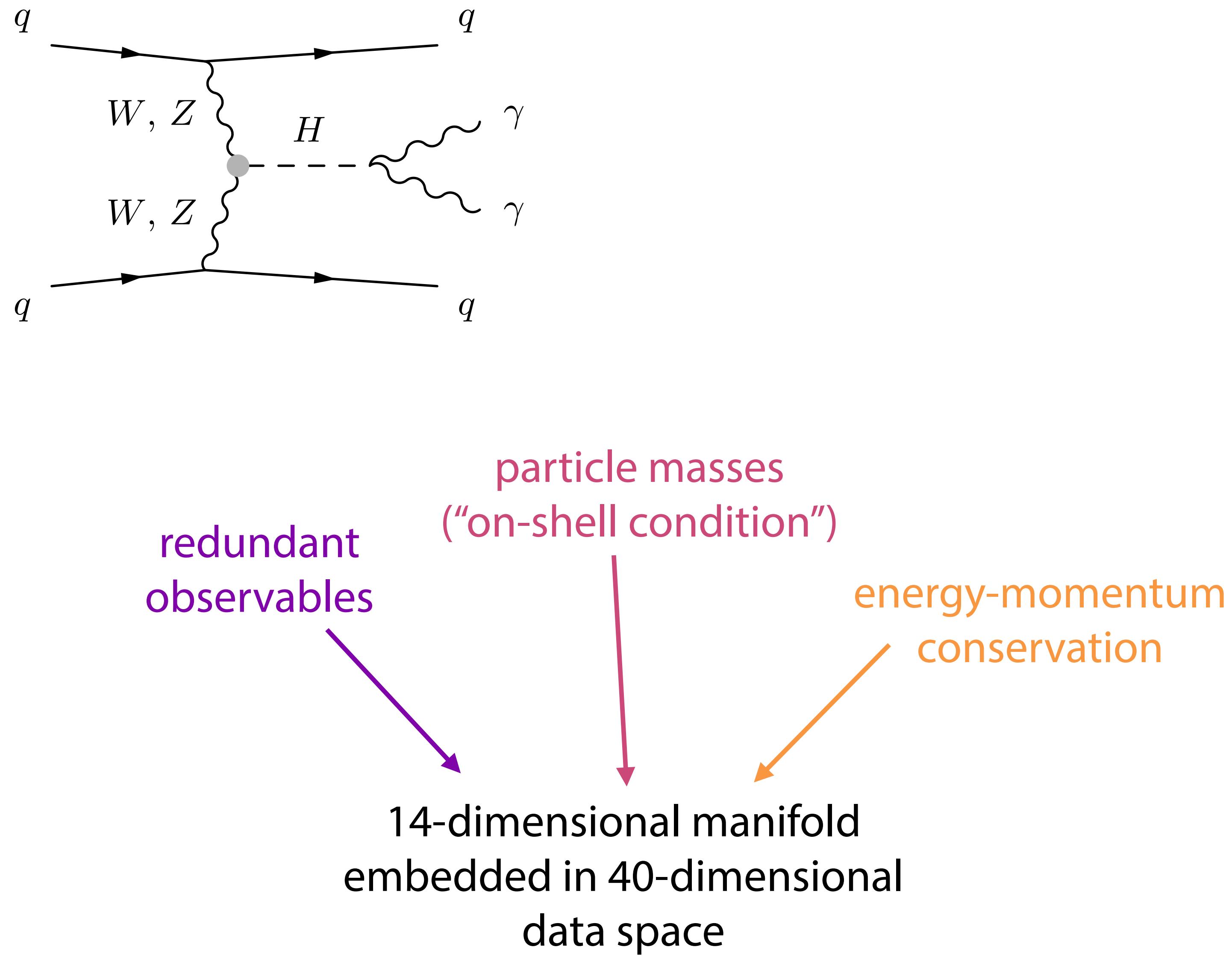
Prediction: Simulator can sample  $x \sim p(x|\theta)$

Inference: Simulator likelihood  $p(x|\theta)$  is intractable,  
but we can train ambient flows or  $\mathcal{M}$ -flows as surrogate

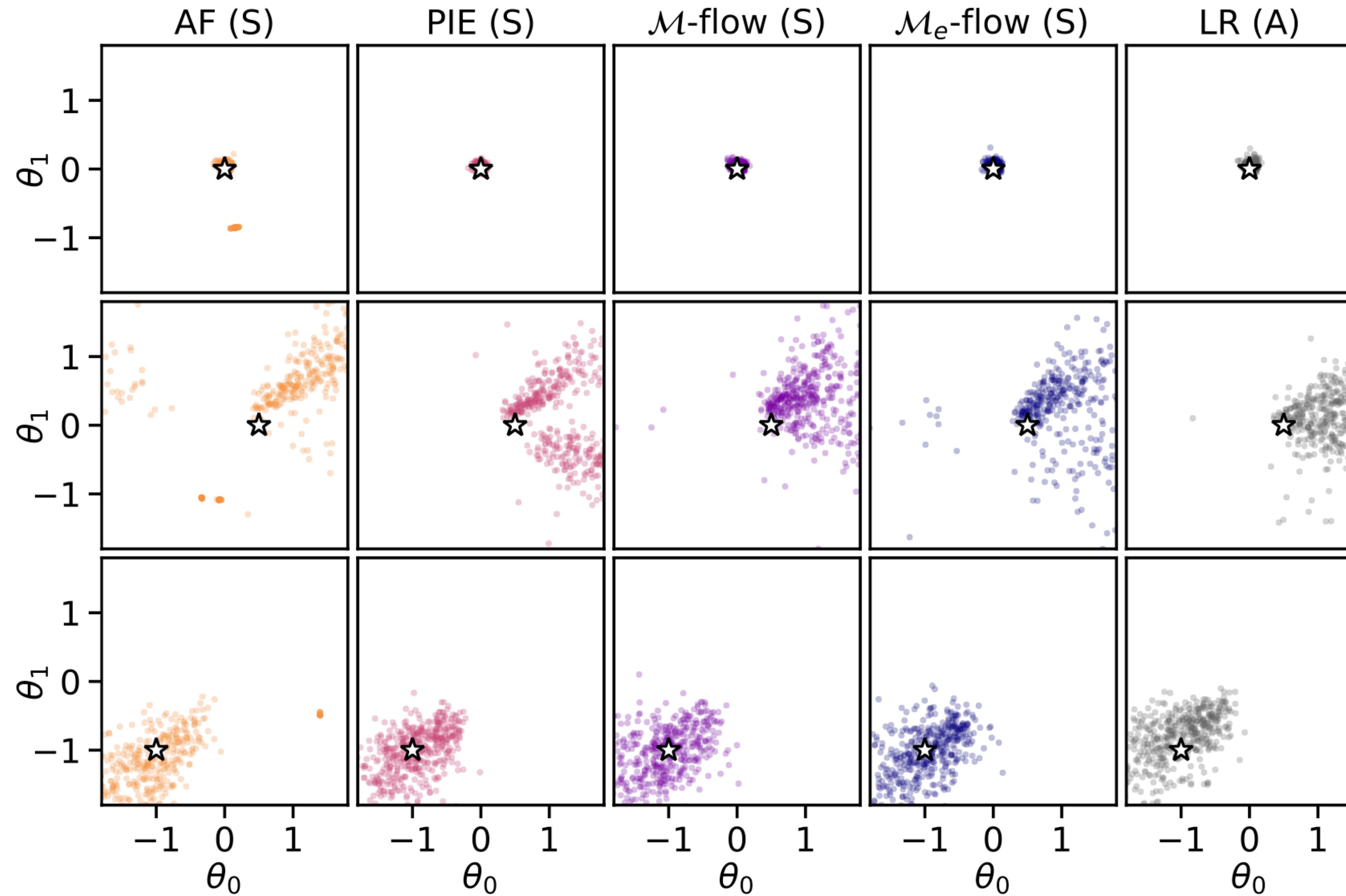
# Particle physics: structure



# Particle physics: structure



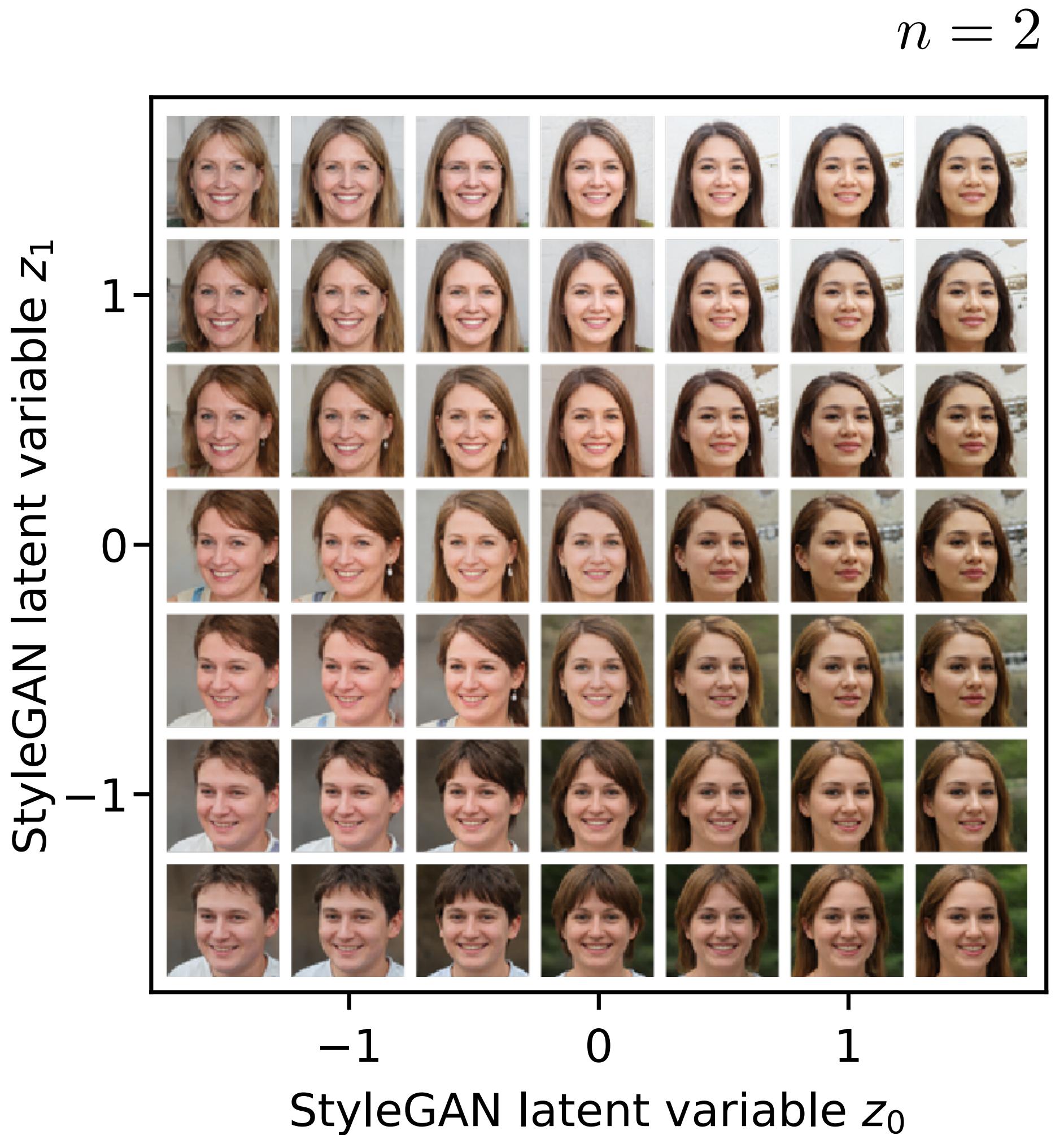
# Particle physics: results



# Image manifolds

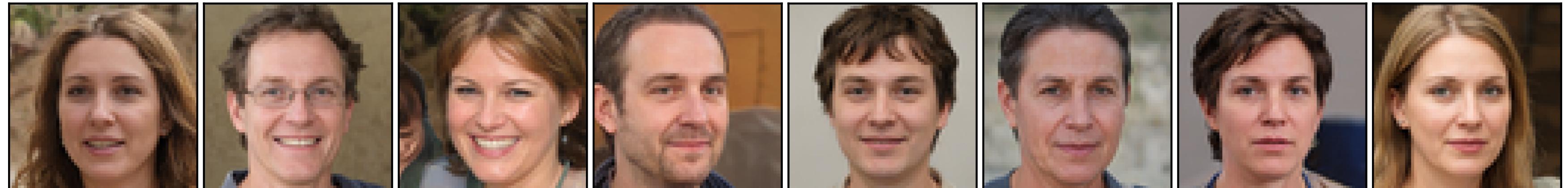
Q: How to make image datasets where we **know** that data lives on an  $n$ -dimensional manifold?

A: take a pretrained GAN model, sample  $n$  of its latent variables, and keep all others fixed



# Samples

Test data



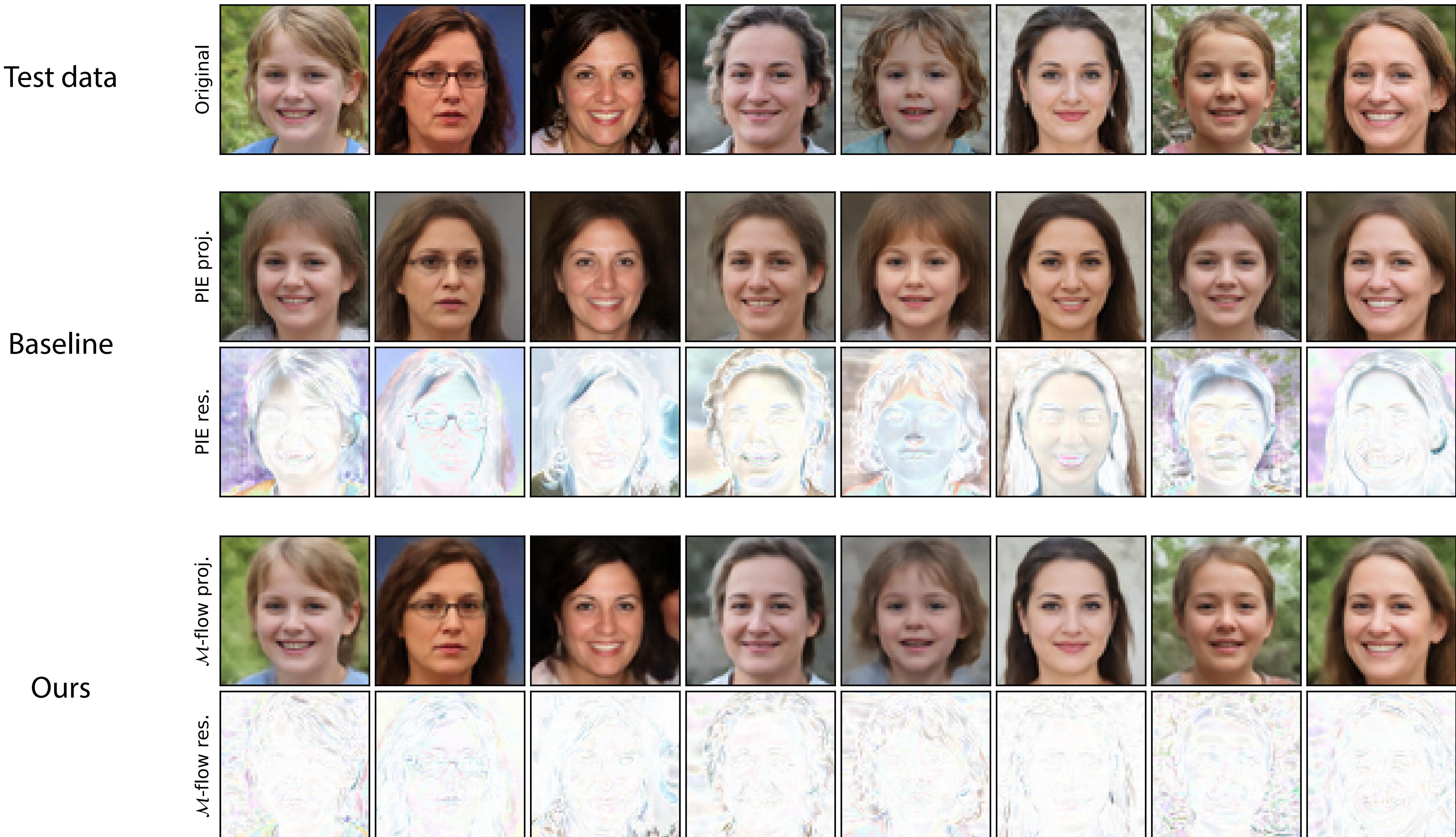
Baselines



Ours



# Projections to learned manifolds



# Open questions

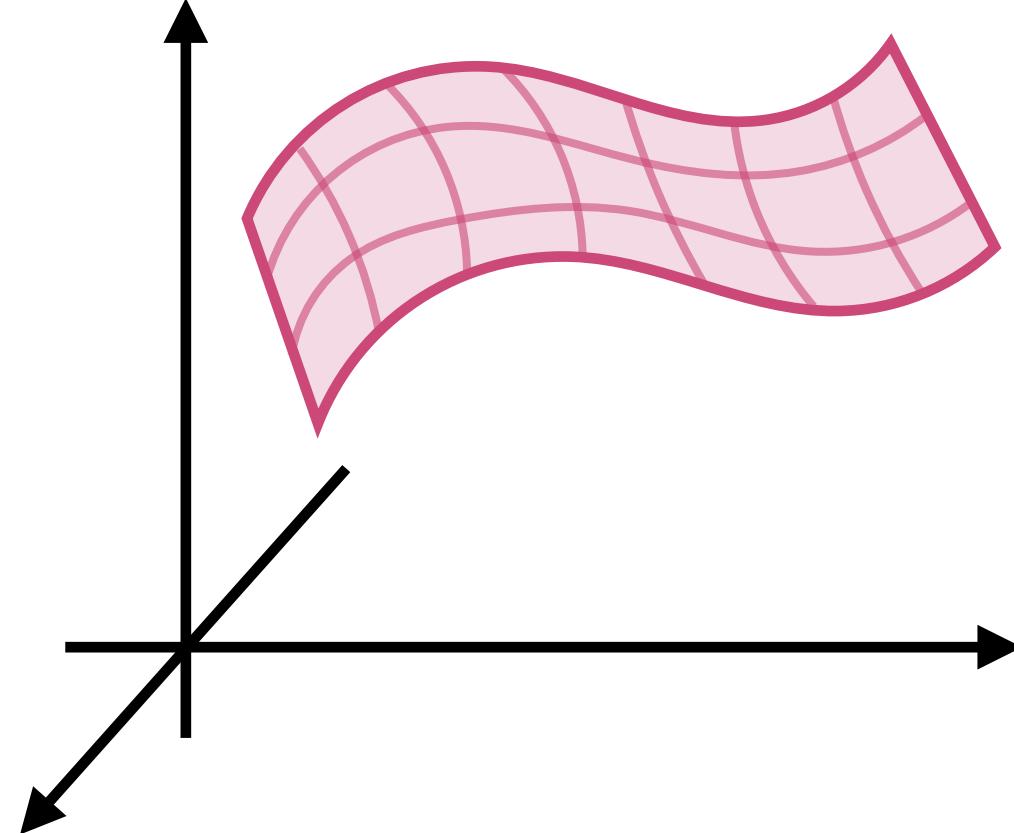
How do you learn the manifold dimensionality?



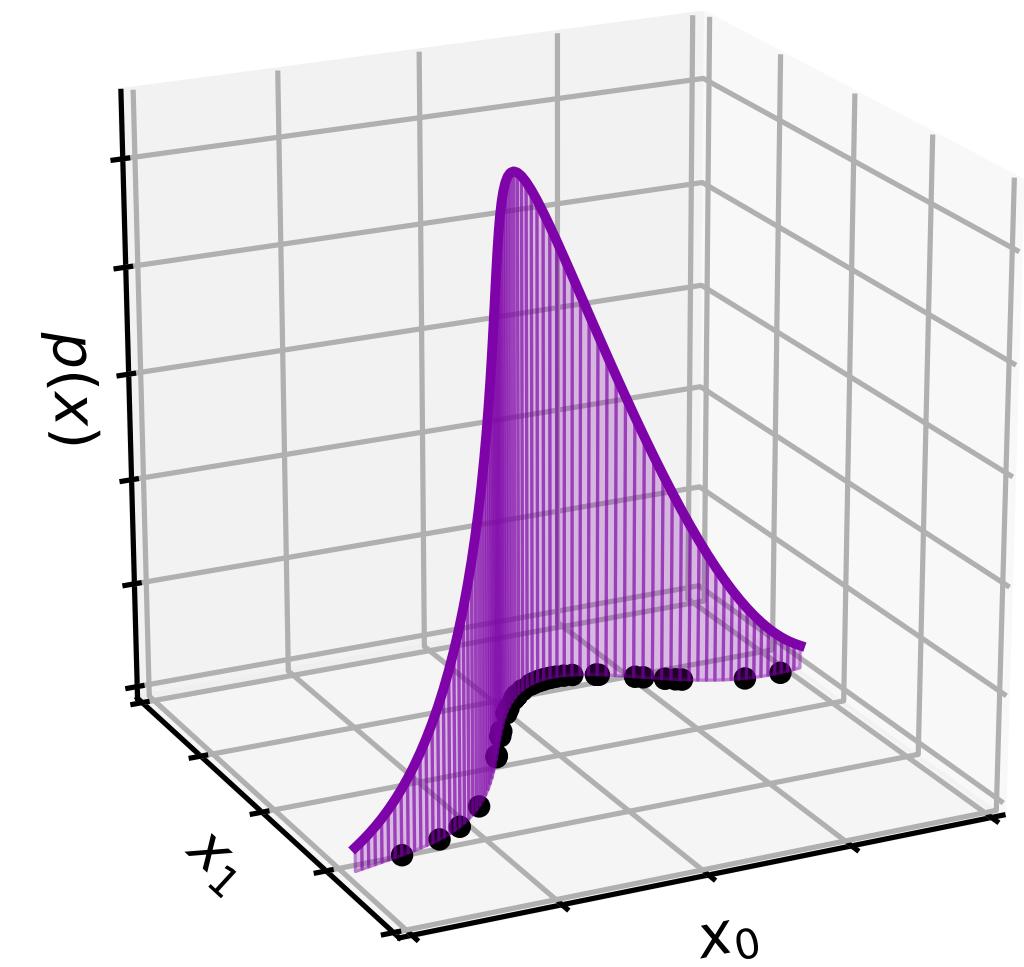
What about non-trivial topologies?

What are good architectures for image data?

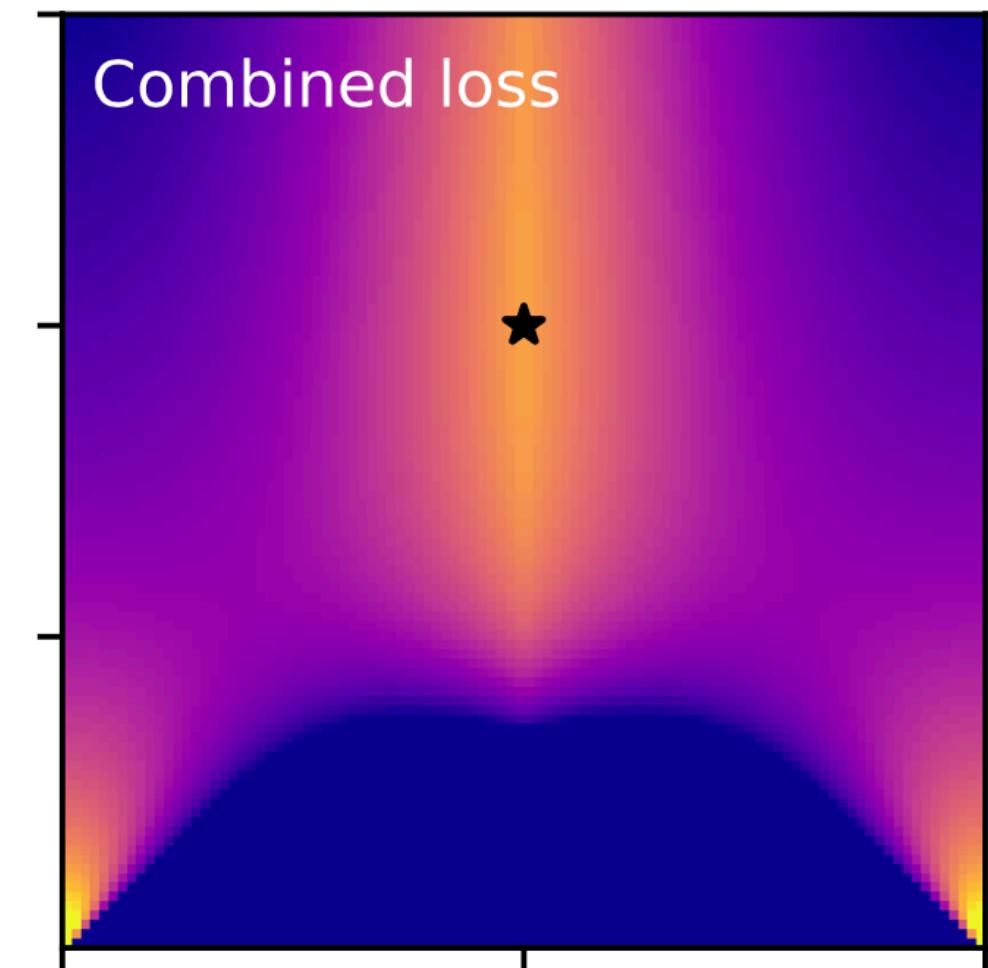
Are there transformations for which the likelihood can be computed faster?



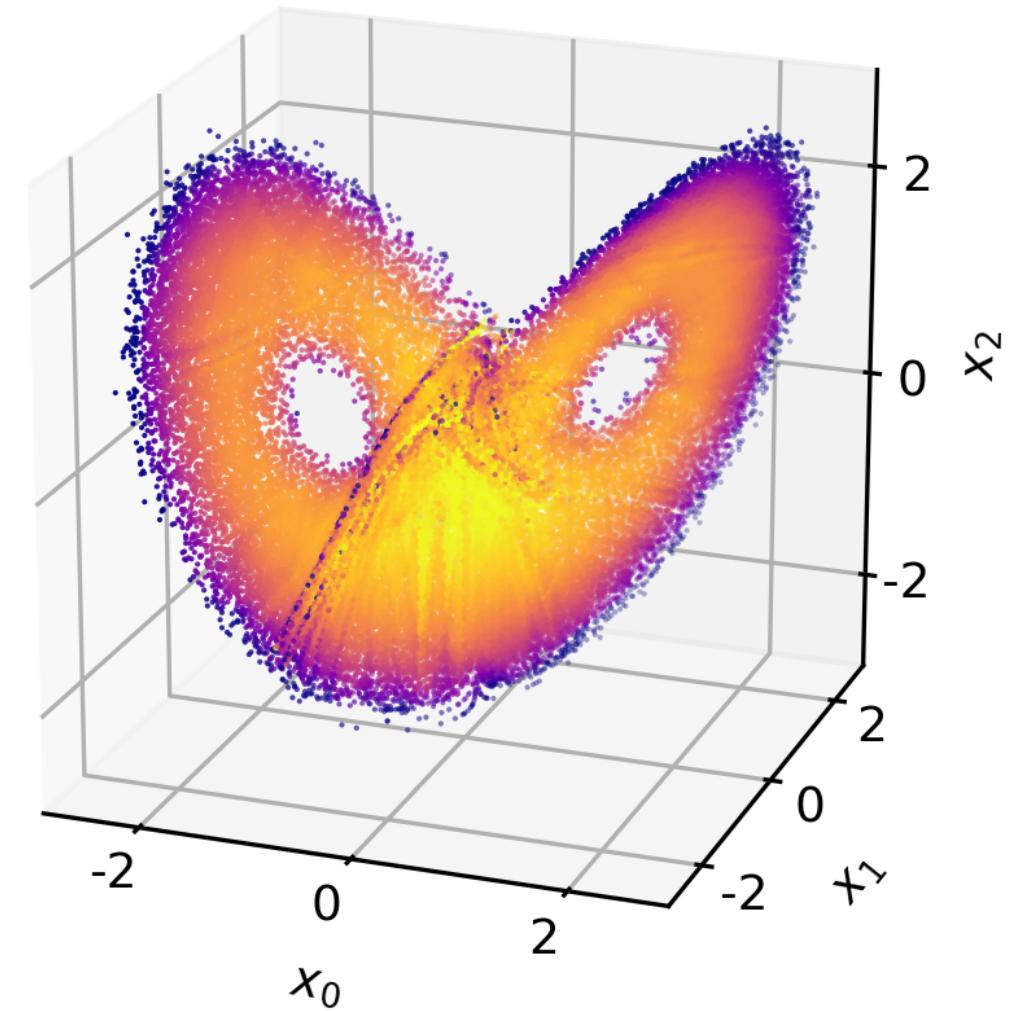
Standard ambient flows  
cannot represent lower-dim. data manifolds



$\mathcal{M}$ -flows learn data  
manifold and a tractable  
density on it



Maximum likelihood is not  
enough, but  $\mathcal{M}$ -flows can  
be trained with M/D  
algorithm



First experiments:  $\mathcal{M}$ -flows  
learn data manifolds well,  
good performance on  
inference tasks

More at: JB, Kyle Cranmer  
arXiv:2003.13913  
NeurIPS 2019

**Flows for simultaneous manifold learning and density estimation**

Johann Brehmer<sup>a,b,1</sup> and Kyle Cranmer<sup>a,b</sup>

<sup>a</sup>Center for Data Science, New York University, USA; <sup>b</sup>Center for Cosmology and Particle Physics, New York University, USA

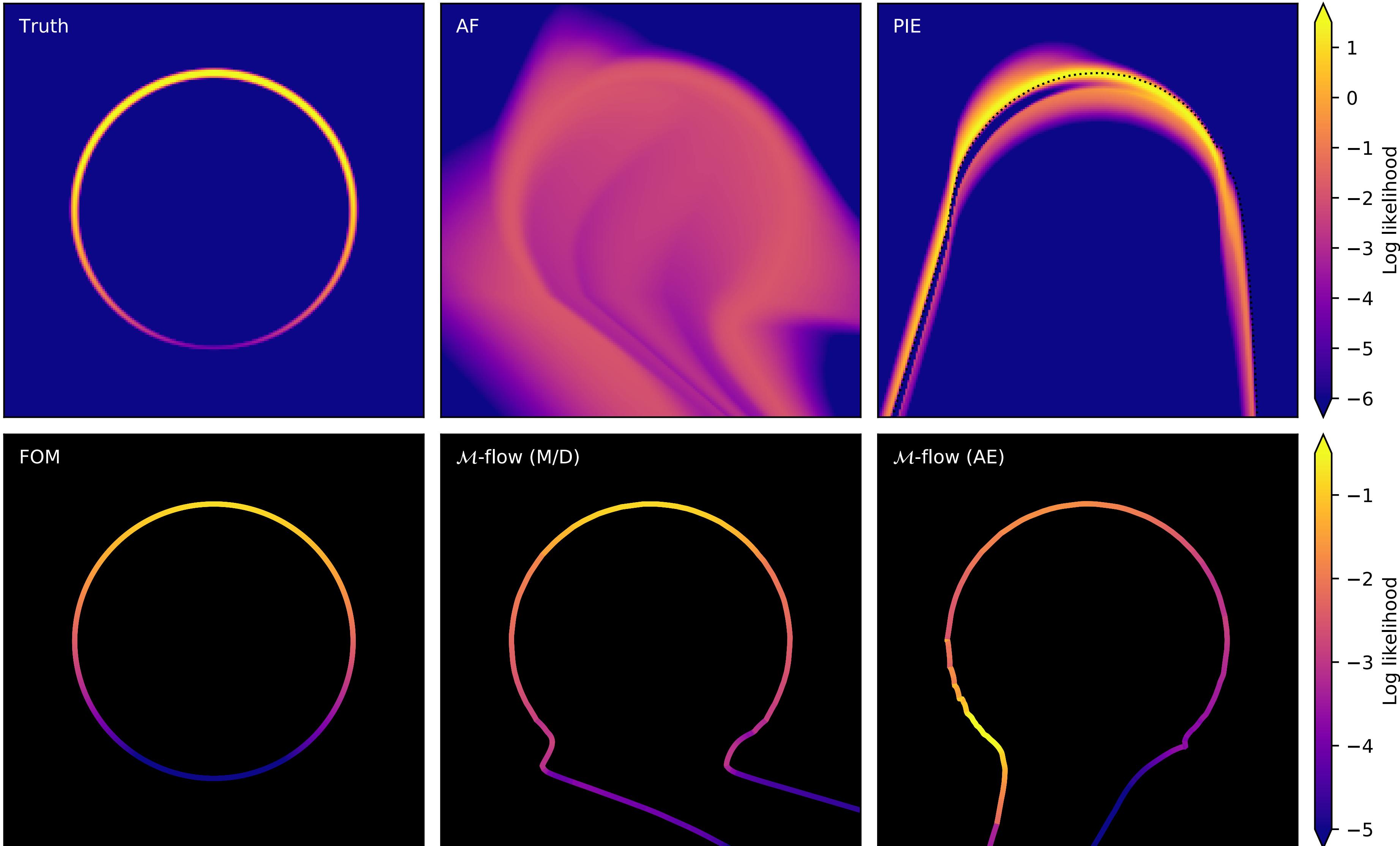
June 16, 2020

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

# B-sides



# Gaussian on a circle



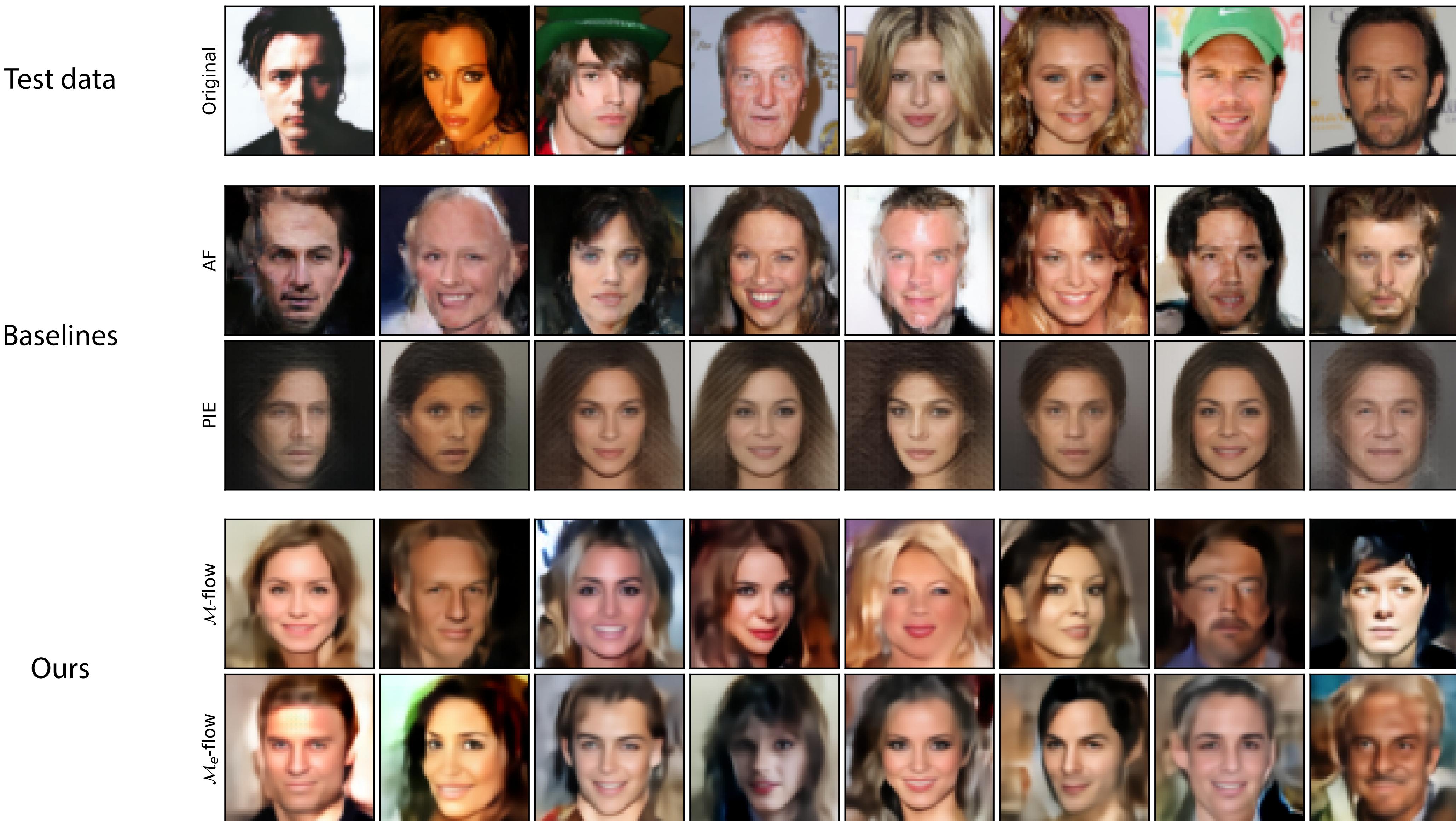
# Latent tour



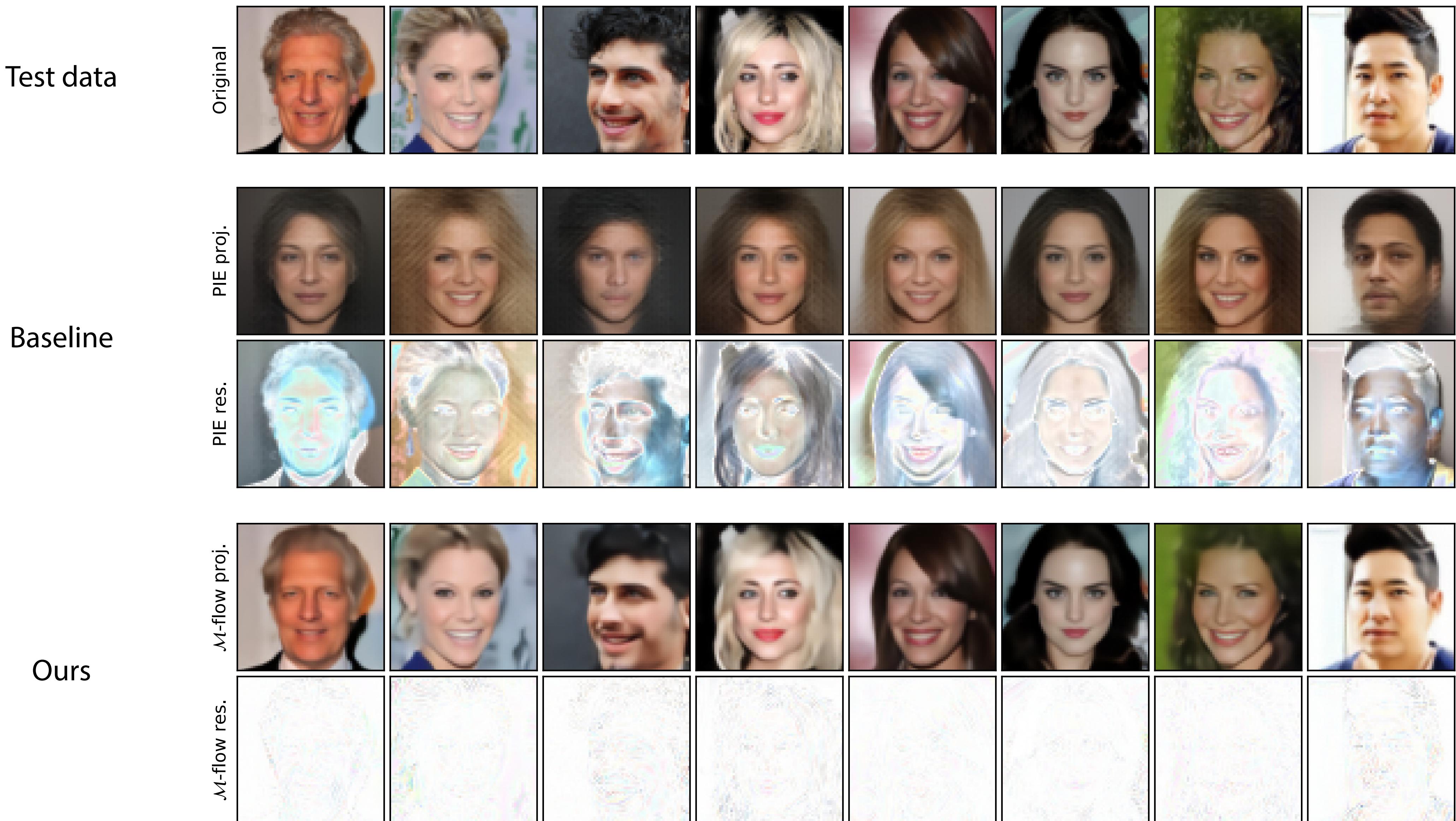
# Latent tour



# Real-world images: CelebA samples



# CelebA projections



# Metrics

Model	Polynomial surface			Particle physics			Images			
	Distance	RE	MMD	Closure	RE	$\log p(\theta^*)$	$n = 2$ FID	$n = 64$ FID	$n = 64$ $\log p(\theta^*)$	CelebA FID
AF	0.005	–	0.071	<b>0.0019</b>	–	−3.94	58.3	24.0	0.17	<b>33.6</b>
PIE	0.035	1.278	0.131	0.0023	2.054	−4.68	139.5	32.2	−6.40	75.7
$\mathcal{M}$ -flow	<b>0.002</b>	<b>0.003</b>	0.020	0.0045	<b>0.012</b>	−1.71	43.9	<b>20.8</b>	<b>2.67</b>	37.4
$\mathcal{M}_e$ -flow	<b>0.002</b>	<b>0.002</b>	<b>0.007</b>	0.0046	0.029	<b>−1.44</b>	<b>43.5</b>	23.7	1.81	35.8