NOTAGAN

Normalizing flows for simul**t**aneous m**a**nifold learnin**g** **a**nd density estimatio**n**

Johann Brehmer
New York University

Physics x ML meeting, NYU
June 26, 2020

NOTAGAN

Normalizing flows for simul**t**aneous m**a**nifold learnin**g** **a**nd density estimatio**n**

Johann Brehmer

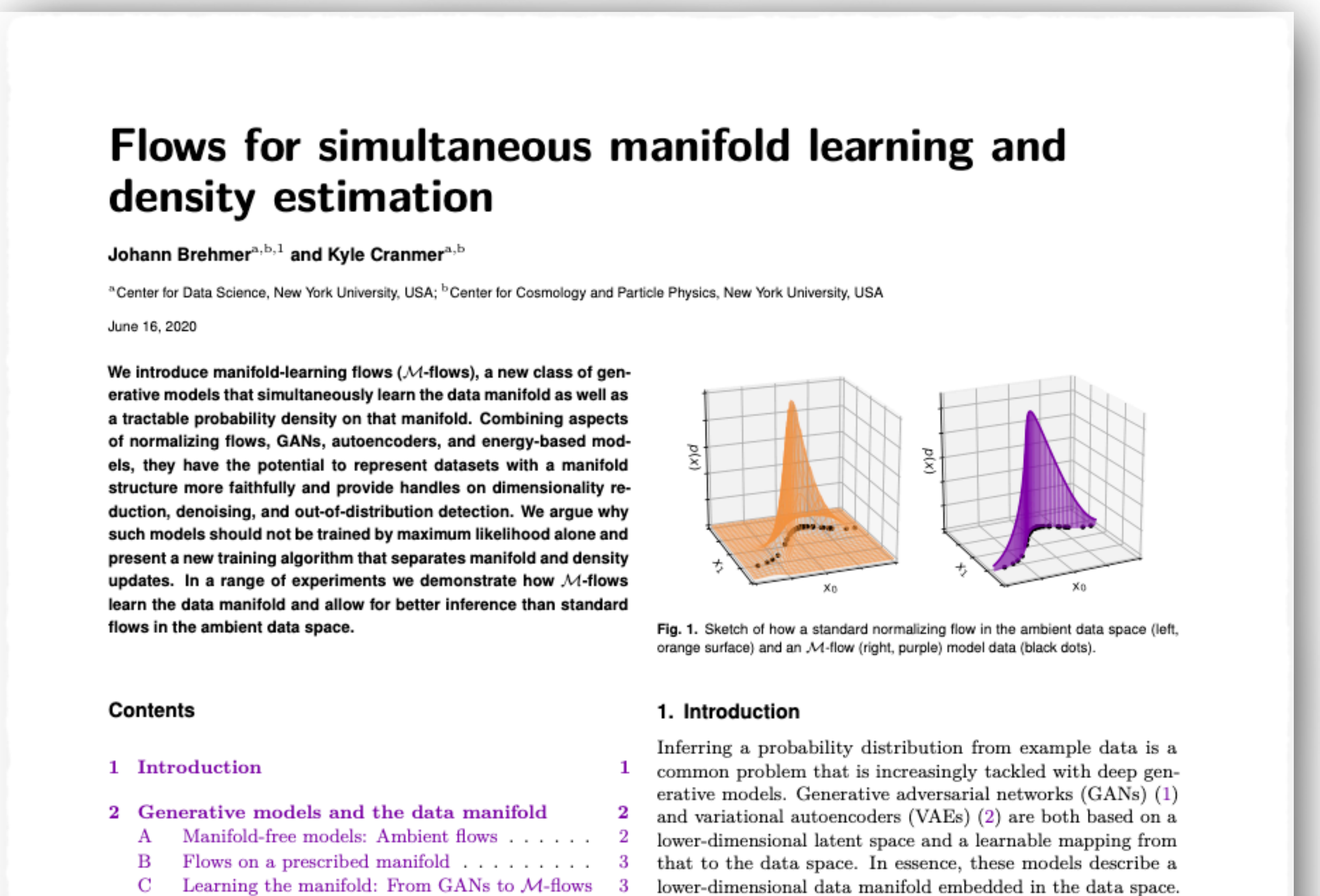
New York University

NYU MSDSE Moore - Sloan Data Science Environments dianahep iris hep Institute for Research and Innovation in Software for High Energy Physics The SCAILFIN Project scailfin.github.io BROOKHAVEN NATIONAL LABORATORY Scientific Data and Computing Center

$\mathcal{M}$-flow: [JB, Kyle Cranmer 2003.13913]

A new generative model that

- describes data with a probability density on a lower-dimensional manifold
- learns manifold and density from data
- has a tractable density and performs well on inference tasks
- provides handles for dimensionality reduction, denoising, OOD detection



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 1

2 Generative models and the data manifold 2
  A Manifold-free models: Ambient flows 2
  B Flows on a prescribed manifold 3
  C Learning the manifold: From GANs to $\mathcal{M}$-flows 3

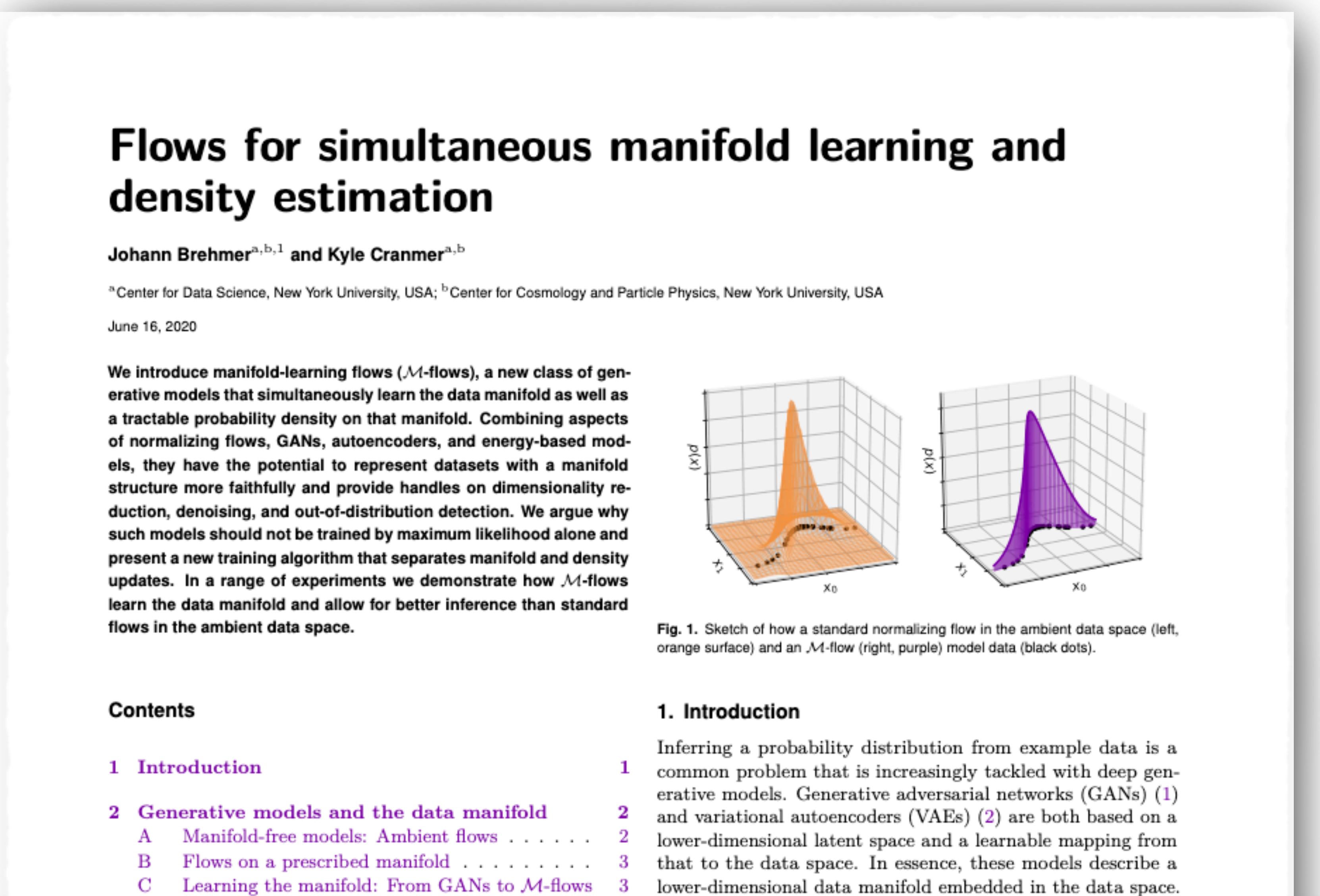
1. Introduction

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 that to the data space. In essence, these models describe a lower-dimensional data manifold embedded in the data space.

$\mathcal{M}$-flow: [JB, Kyle Cranmer 2003.13913]

A new generative model that

- describes data with a probability density on a lower-dimensional manifold
- learns manifold and density from data
- has a tractable density and performs well on inference tasks
- provides handles for dimensionality reduction, denoising, OOD detection



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 1
2 Generative models and the data manifold 2
A Manifold-free models: Ambient flows 2
B Flows on a prescribed manifold 3
C Learning the manifold: From GANs to $\mathcal{M}$-flows 3

1. Introduction

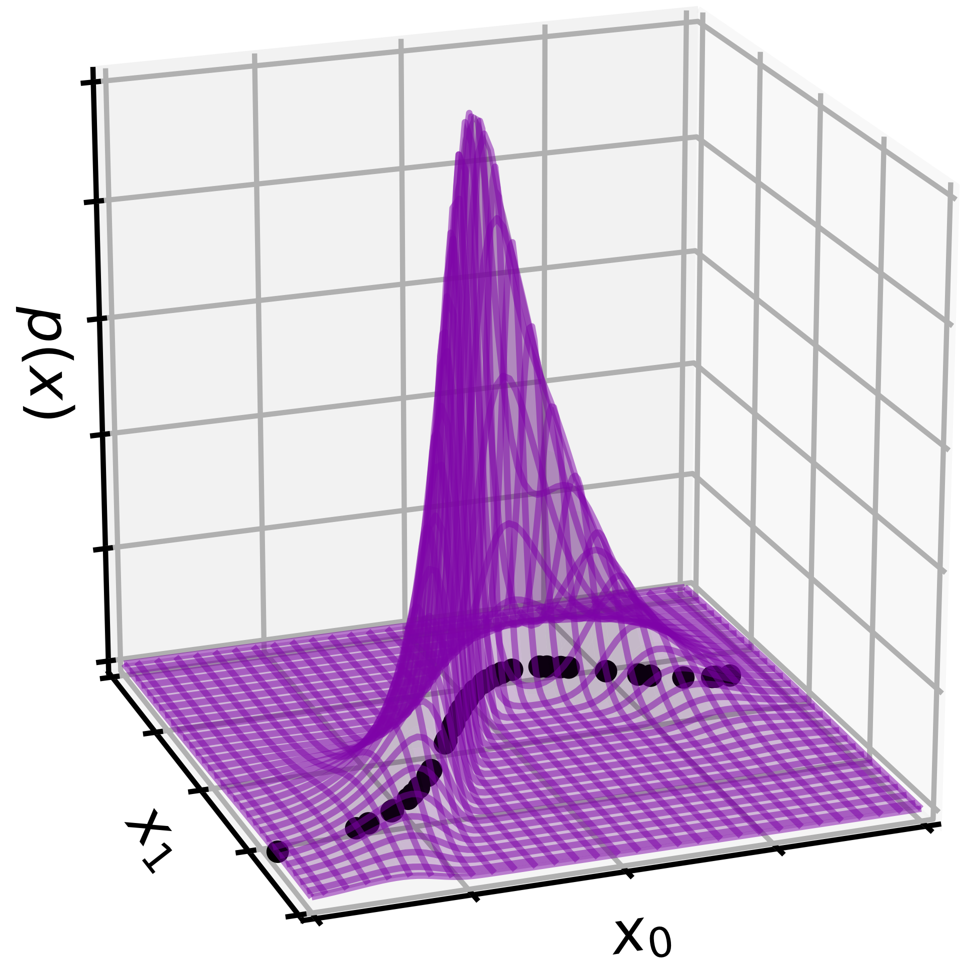
Inferring a probability distribution from example data is a common problem that is increasingly tackled with deep generative models. Generative adversarial networks (GANs) (1) and variational autoencoders (VAEs) (2) are both based on a lower-dimensional latent space and a learnable mapping from that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

m-flo:

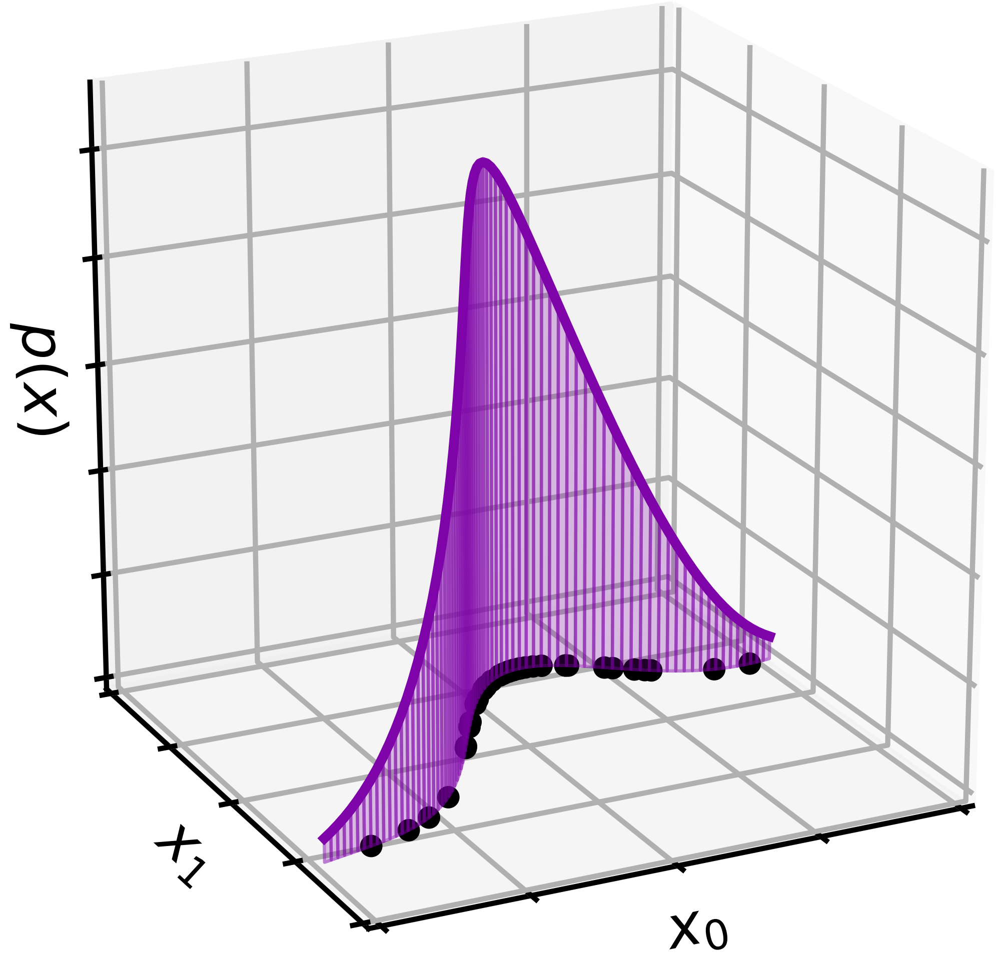
Japanese hip hop group known for

- Love me after 12 am
- The love bug
- Love song
- Love don't cry
- Stuck in your love
- The other side of love
- Lotta Love
- and many more

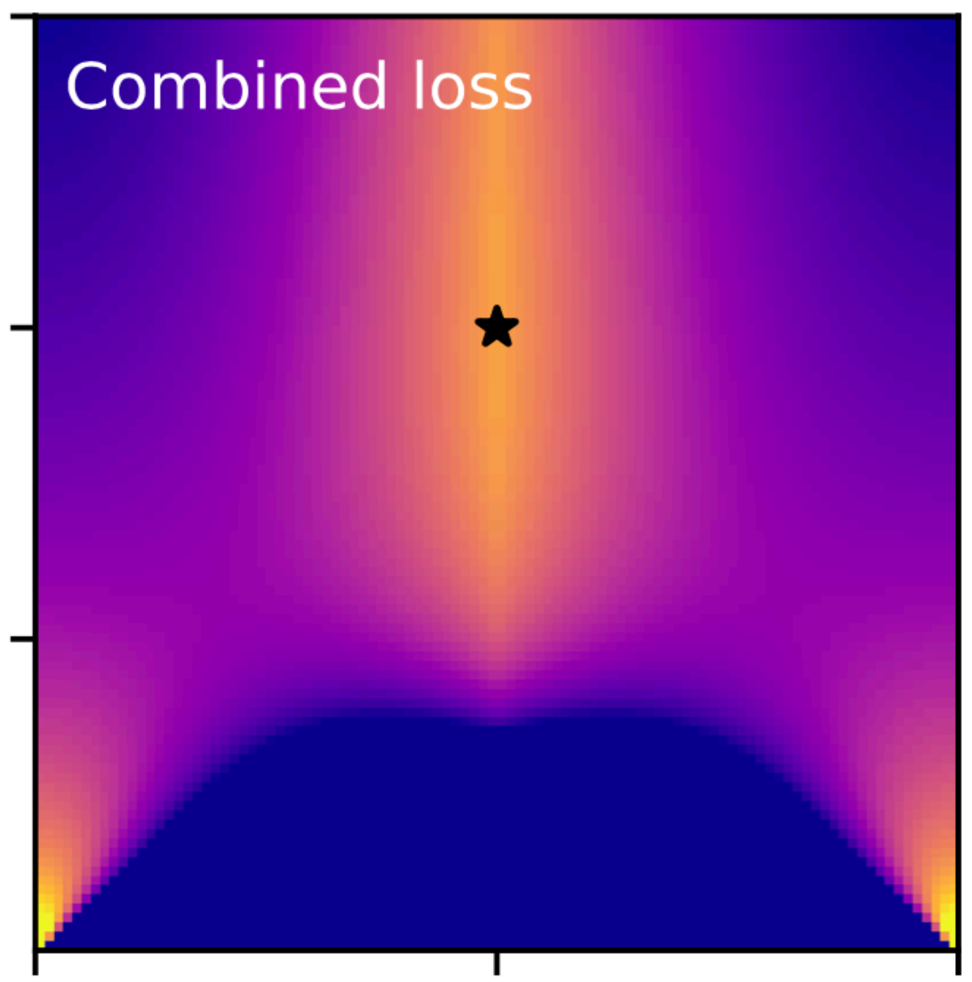
[m-flo / Rhythm Zone]



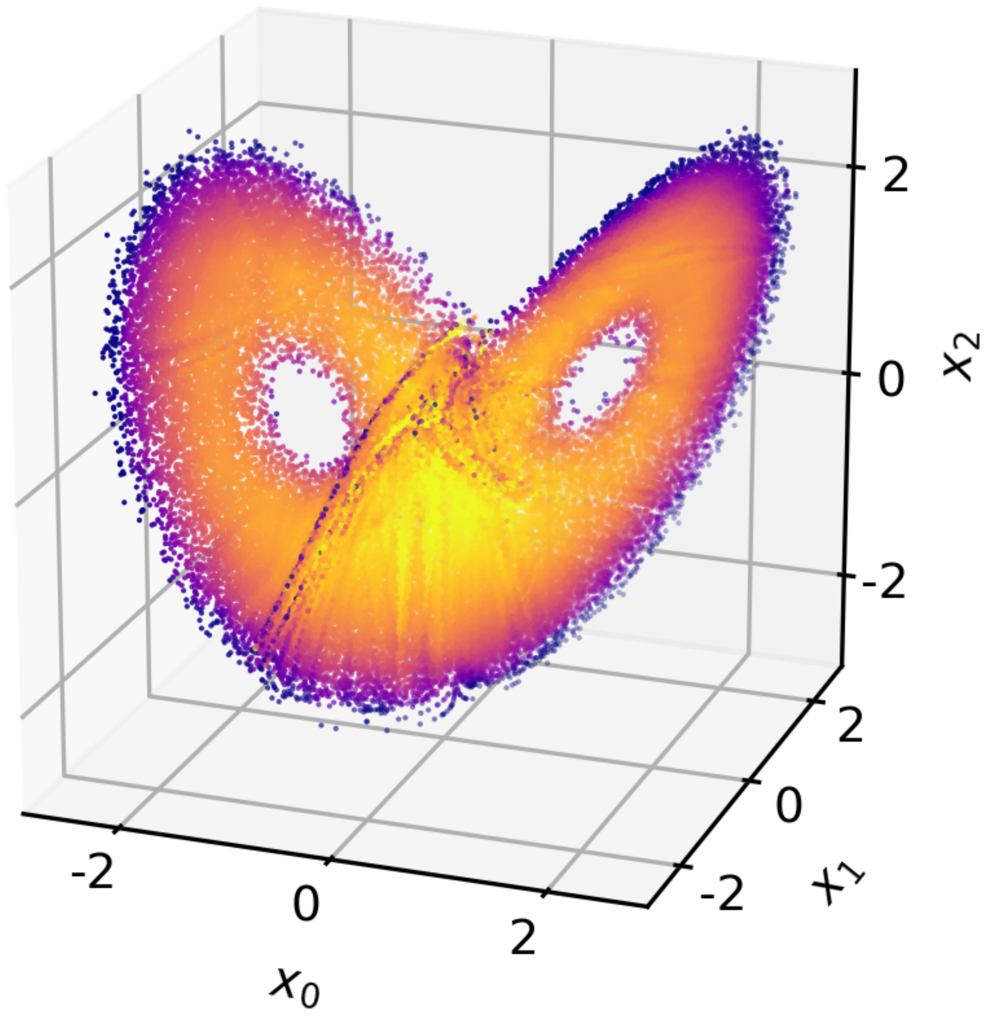
1. Generative models and the data manifold



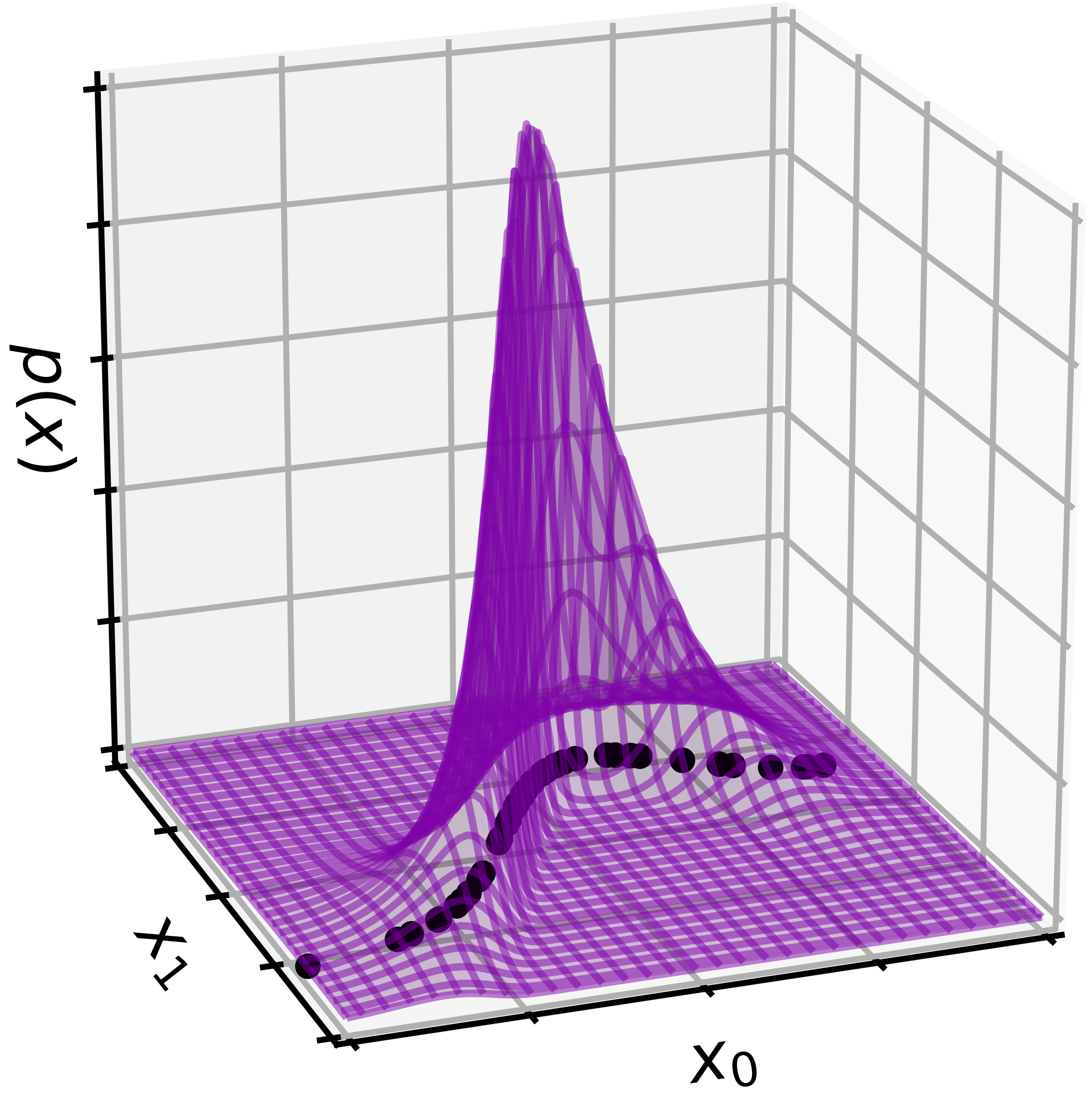
2. $\mathcal{M}$-flows



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



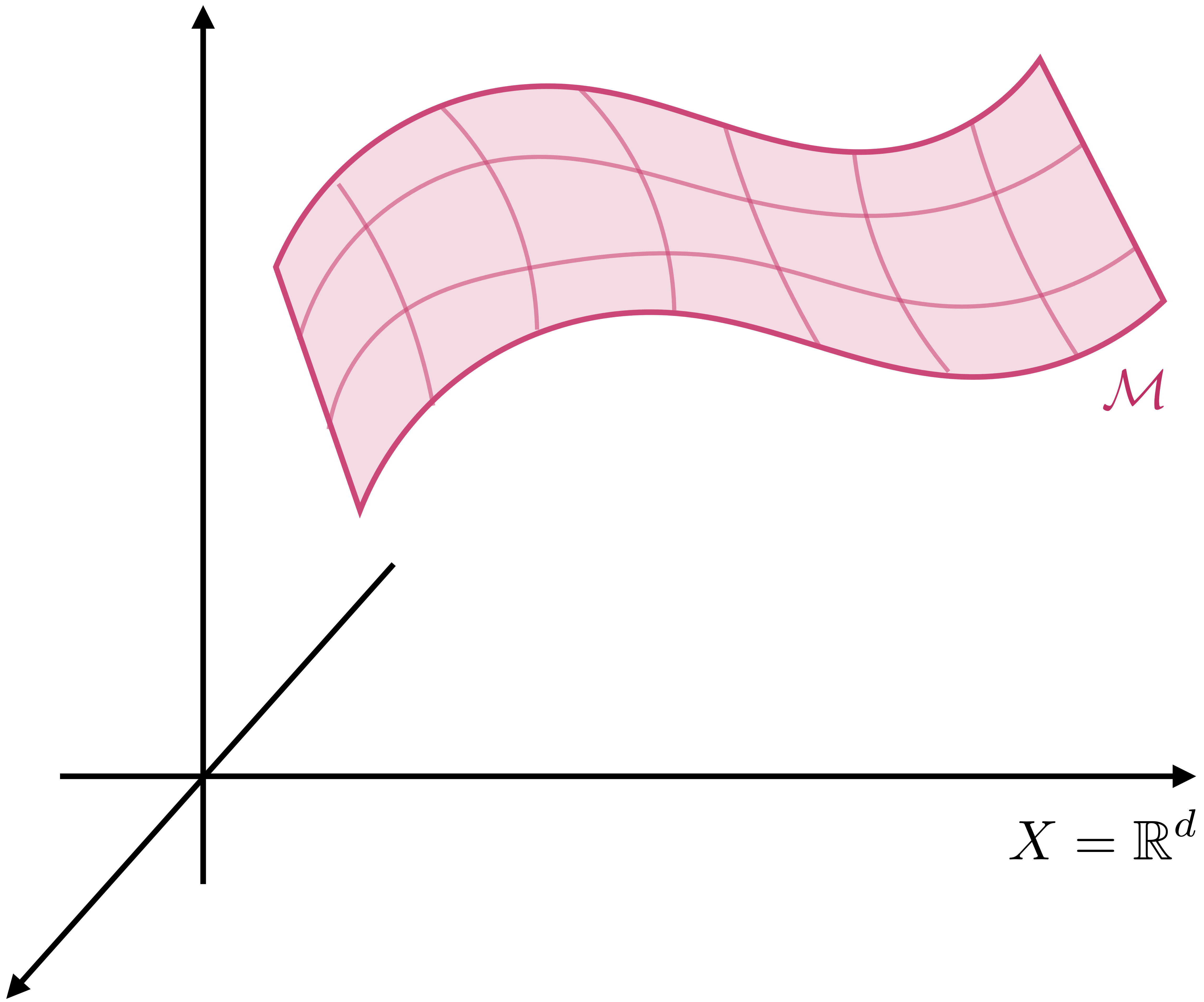
4. Experiments



Generative models and the data manifold

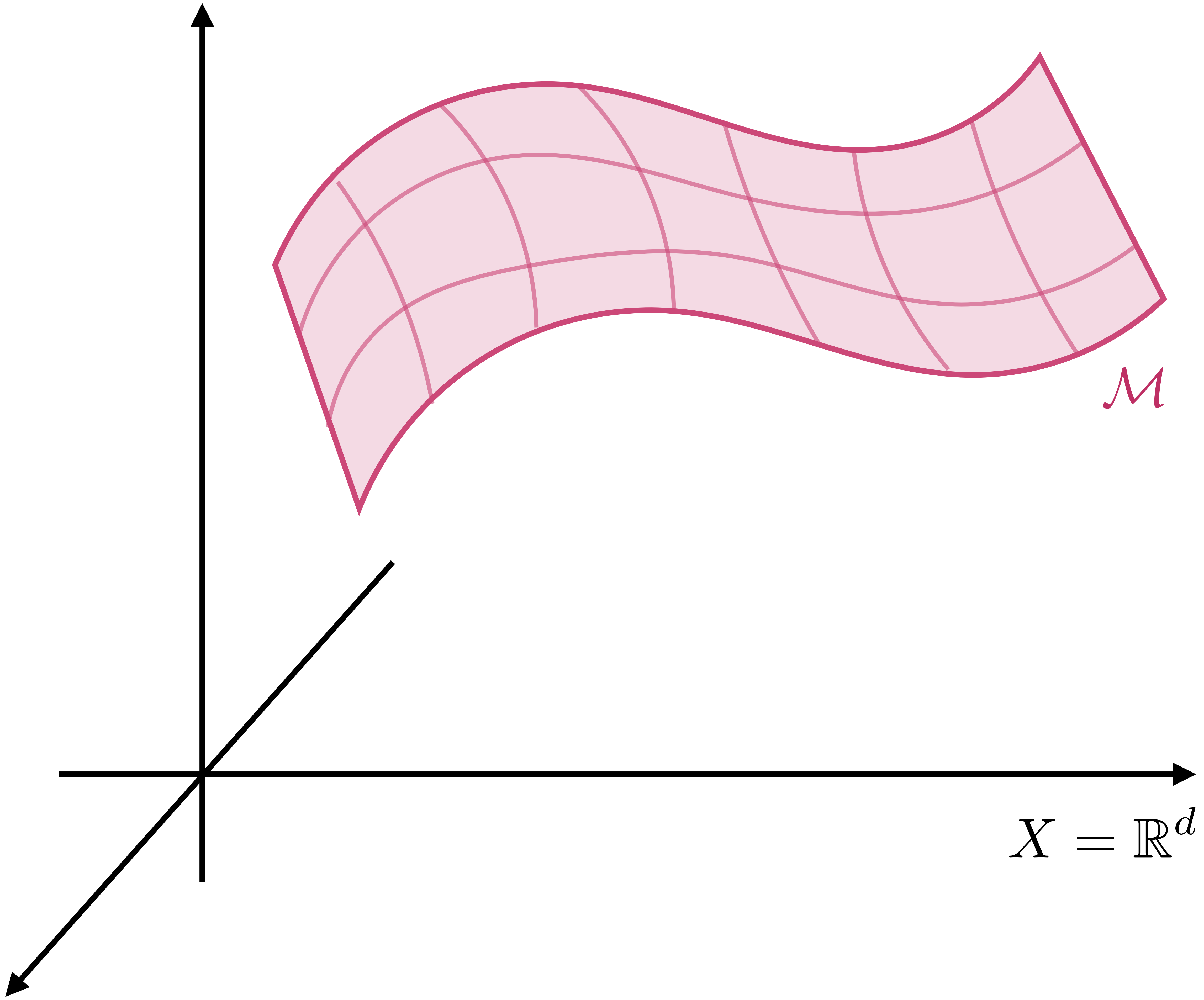
The data manifold

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



The data manifold

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

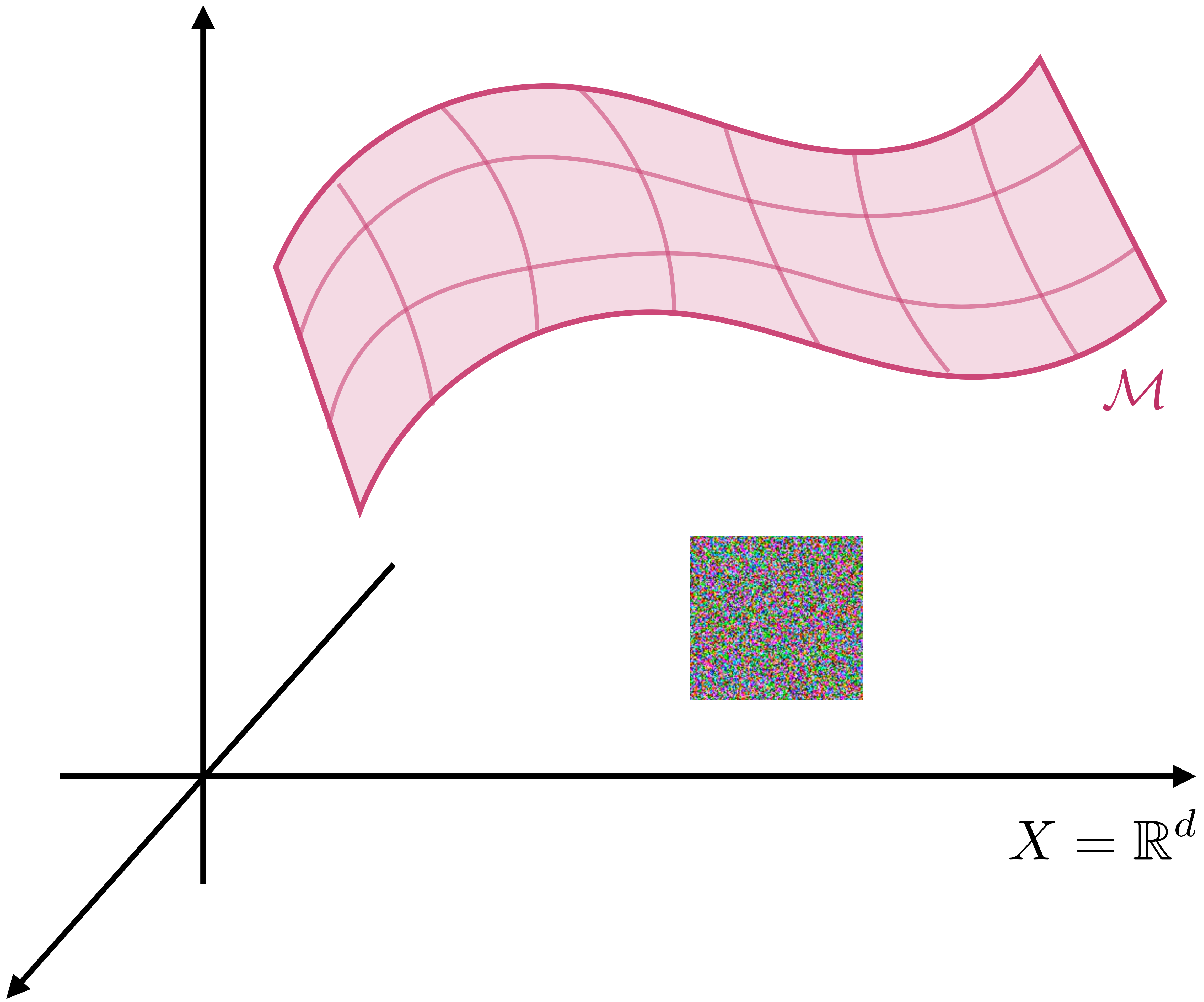


- Robot arms: limited degrees of freedom
- Particle physics: energy-momentum conservation, on-shell conditions, redundant observables
- Many other high-dimensional datasets (e.g. images): empirical evidence for (approximate) data manifold
  [L. Cayton 2005; …]

The data manifold

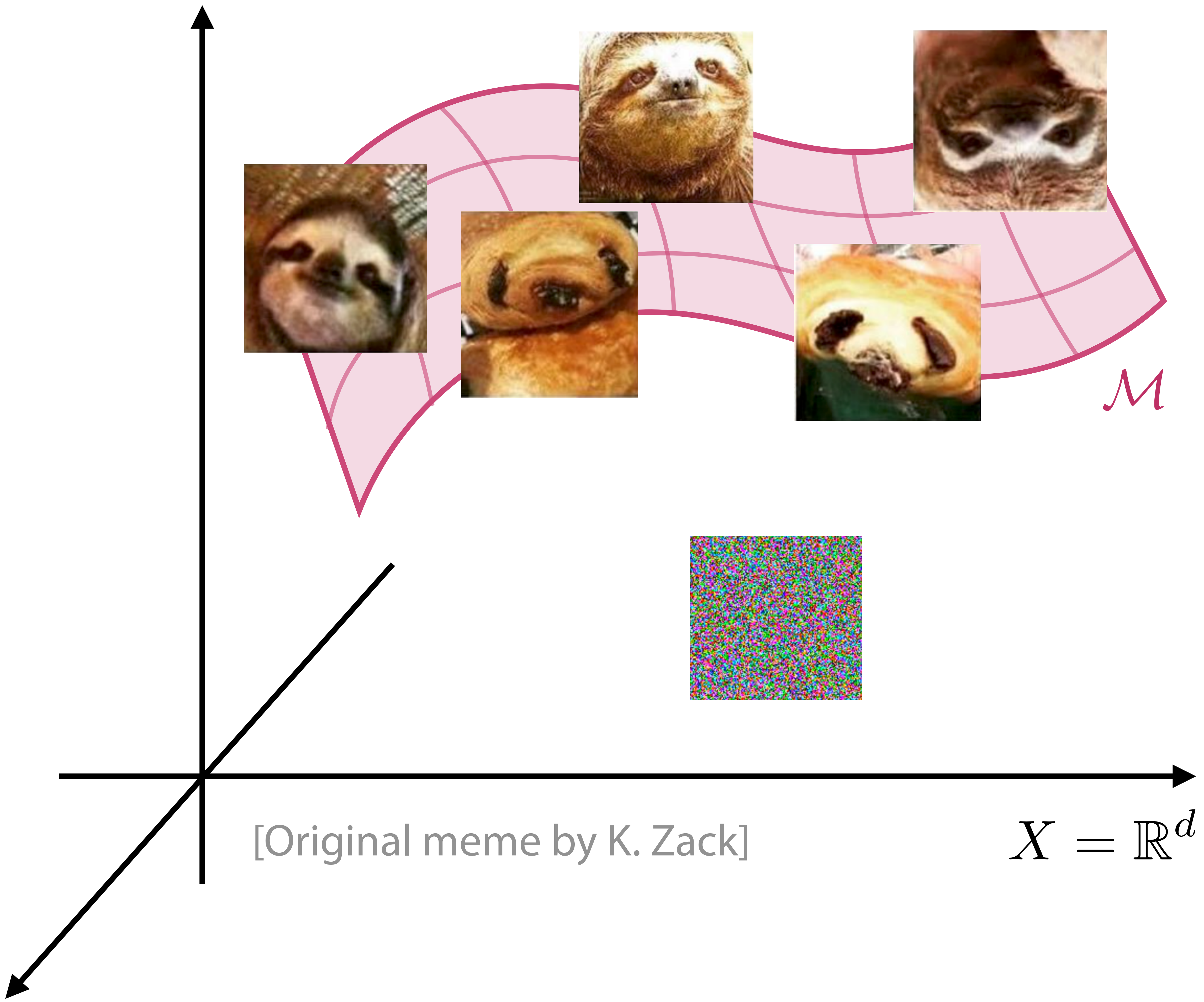
Data often live on a n-dimensional manifold embedded in the d-dimensional ambient space

- Robot arms: 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 data manifold

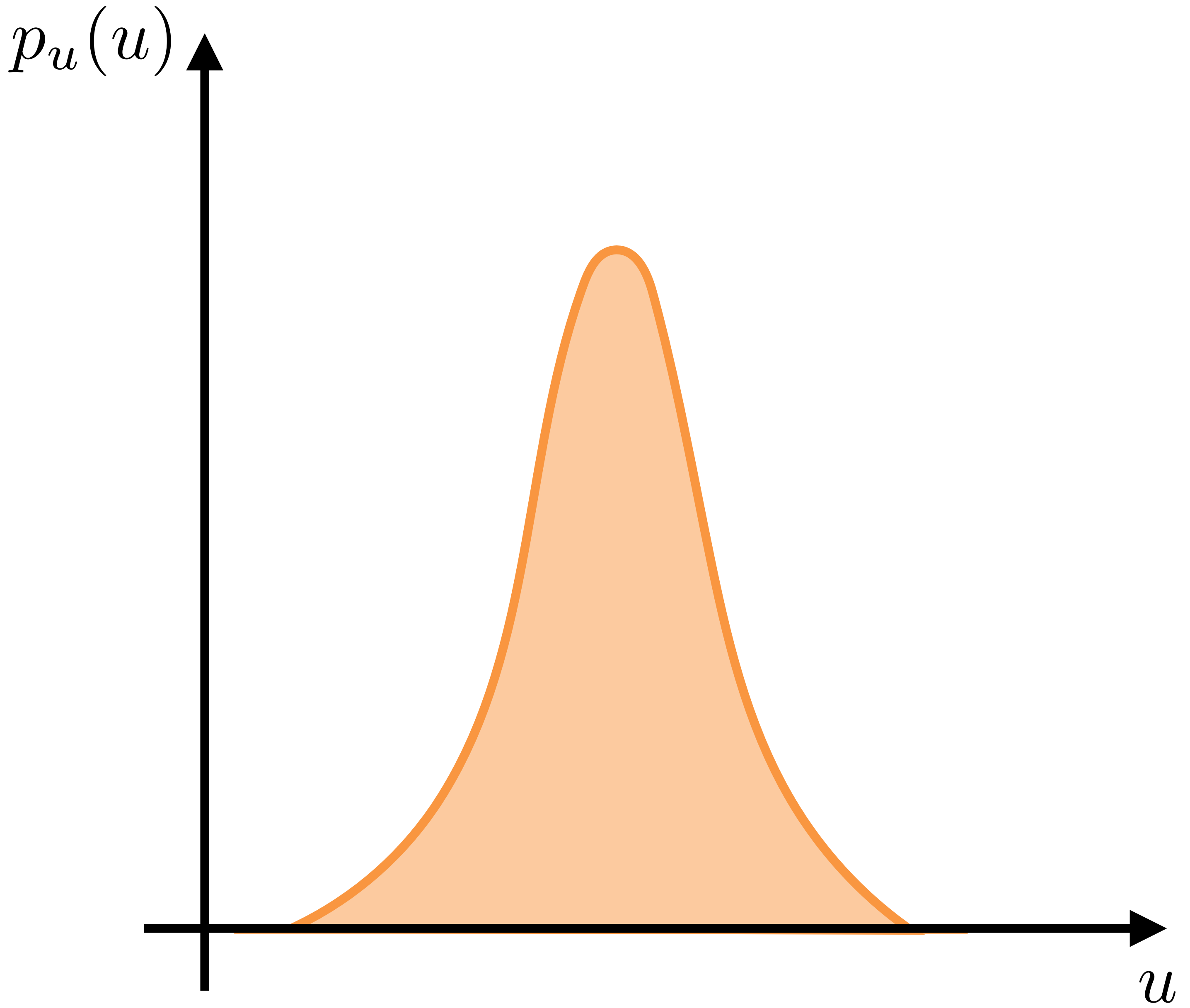
Data often live on a n-dimensional manifold embedded in the d-dimensional ambient space



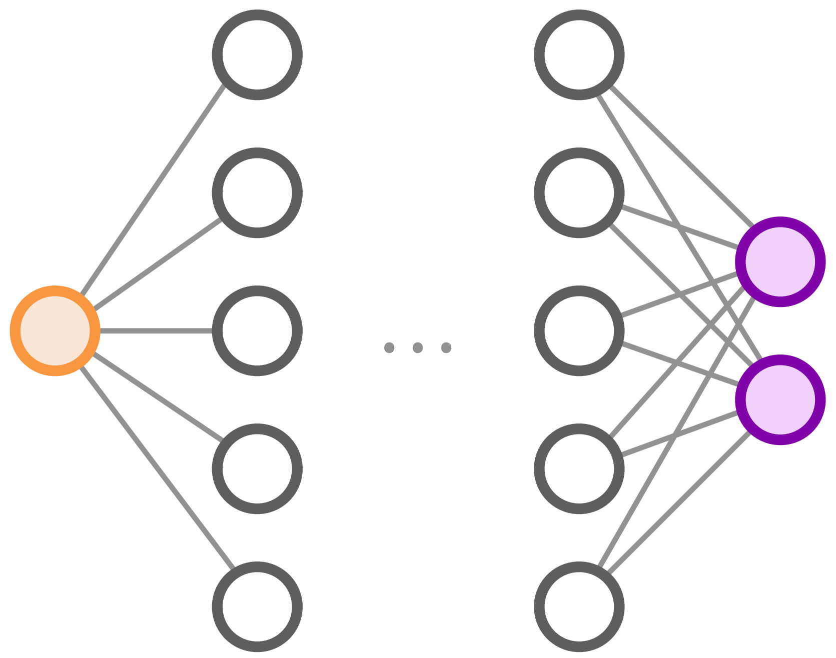
- Robot arms: limited degrees of freedom
- Particle physics: energy-momentum conservation, on-shell conditions, redundant observables
- Many other high-dimensional datasets (e.g. images): empirical evidence for (approximate) data manifold
  [L. Cayton 2005; …]

Generative adversarial networks (GANs)

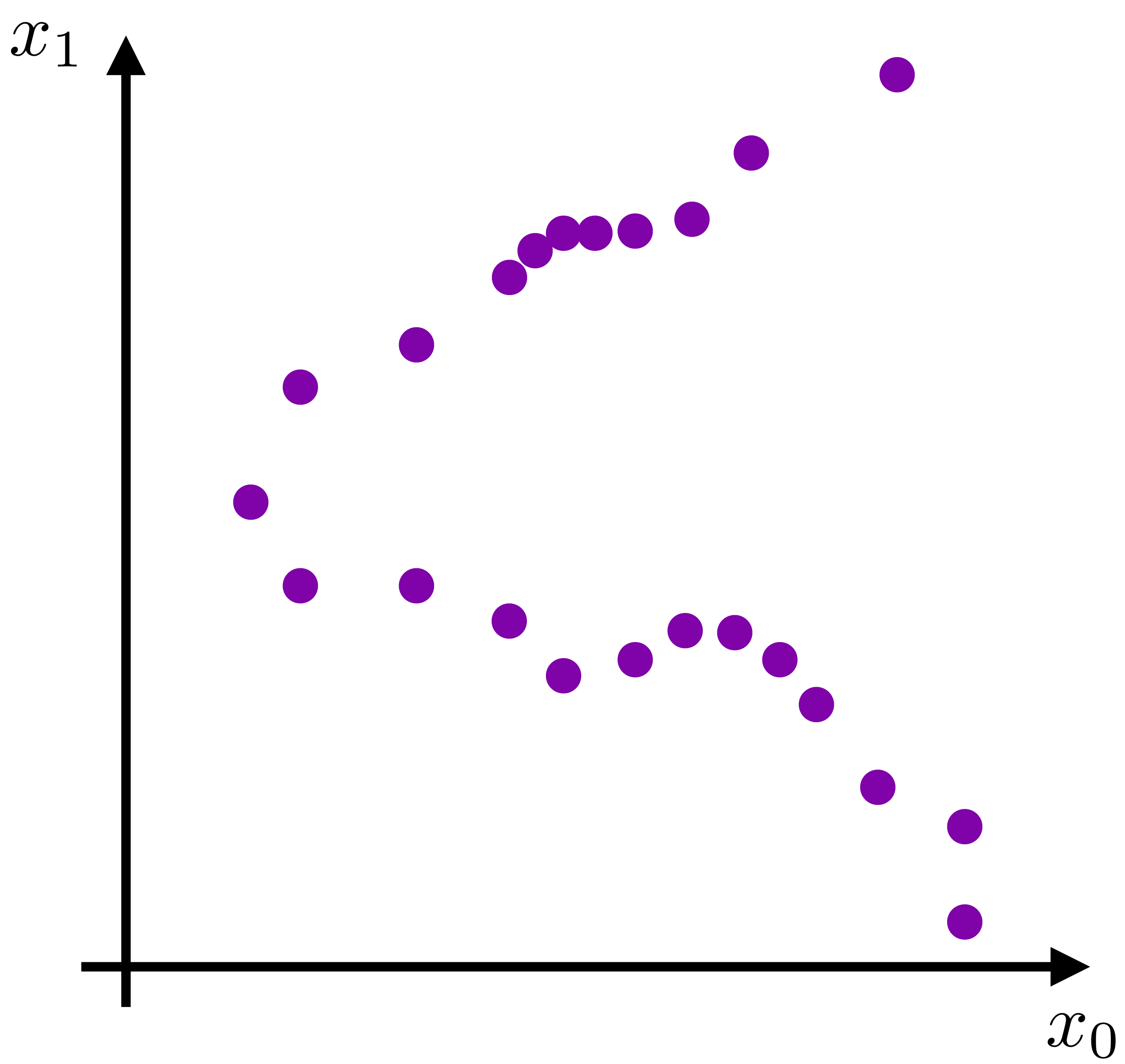
[I. Goodfellow et al 1406.2661]



$u \sim p_u(u)$



$\xrightarrow{g}$



x

n-dim. latent variables

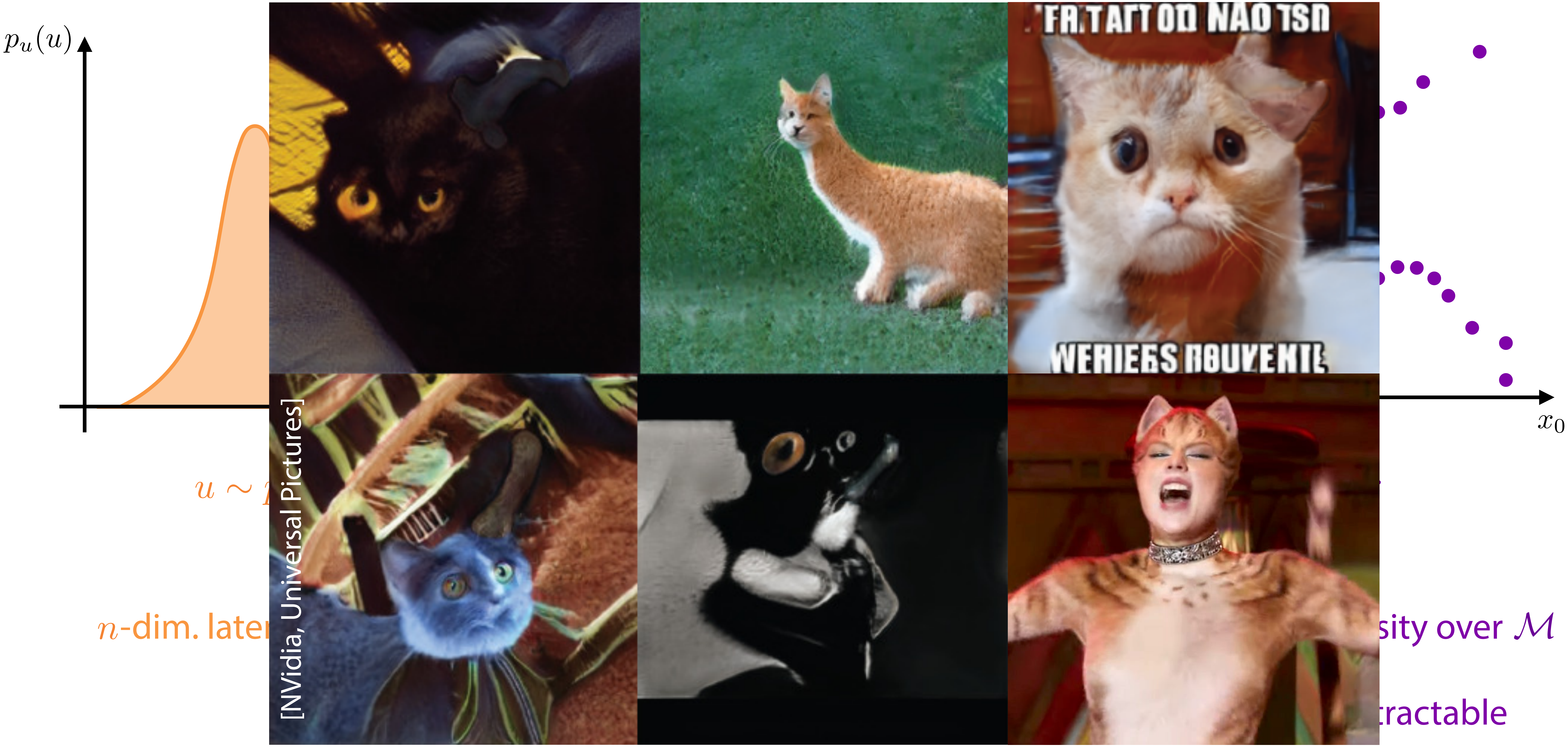
unconstrained NN

implicit density over $\mathcal{M}$

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

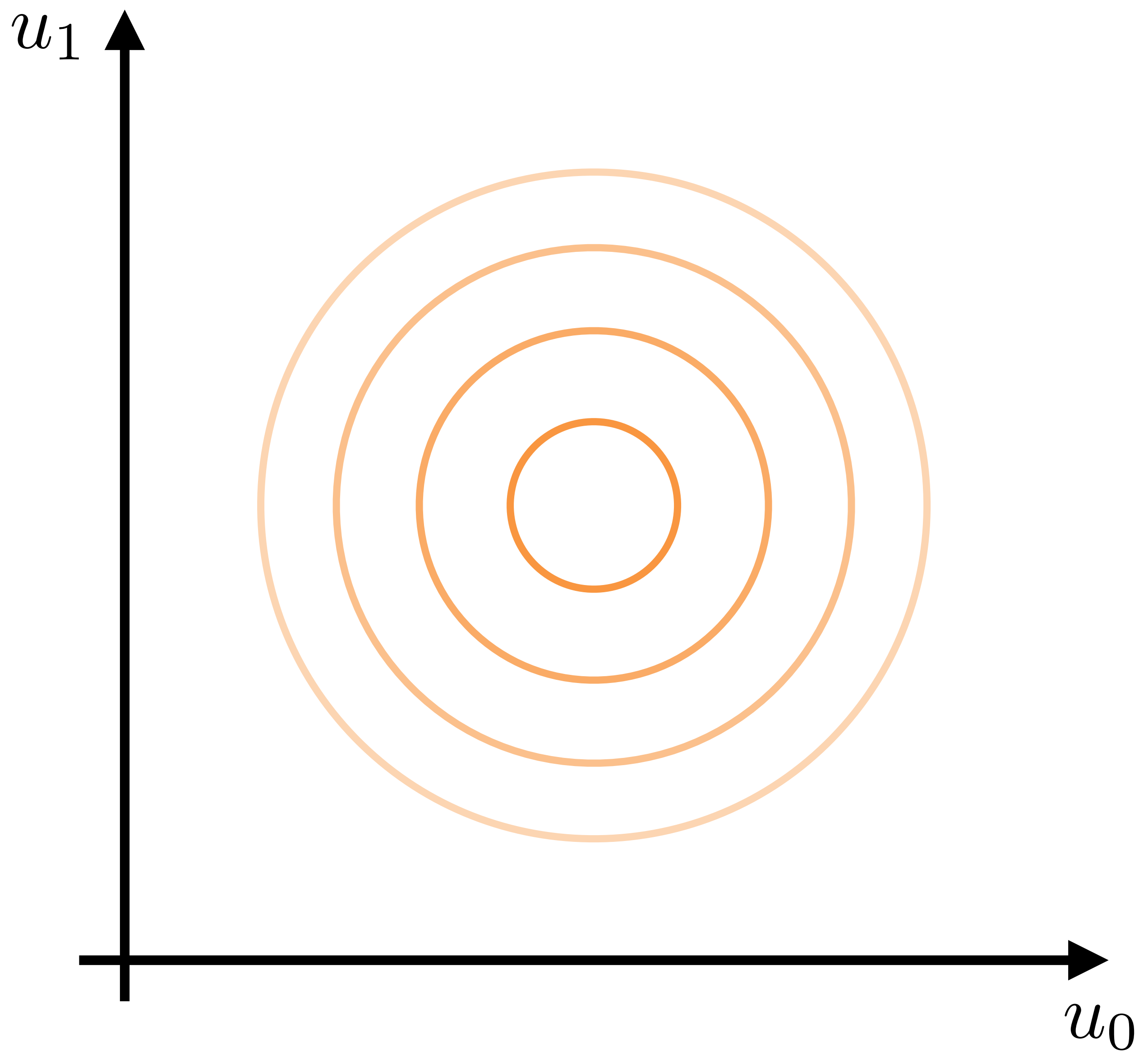
Generative adversarial networks (GANs)

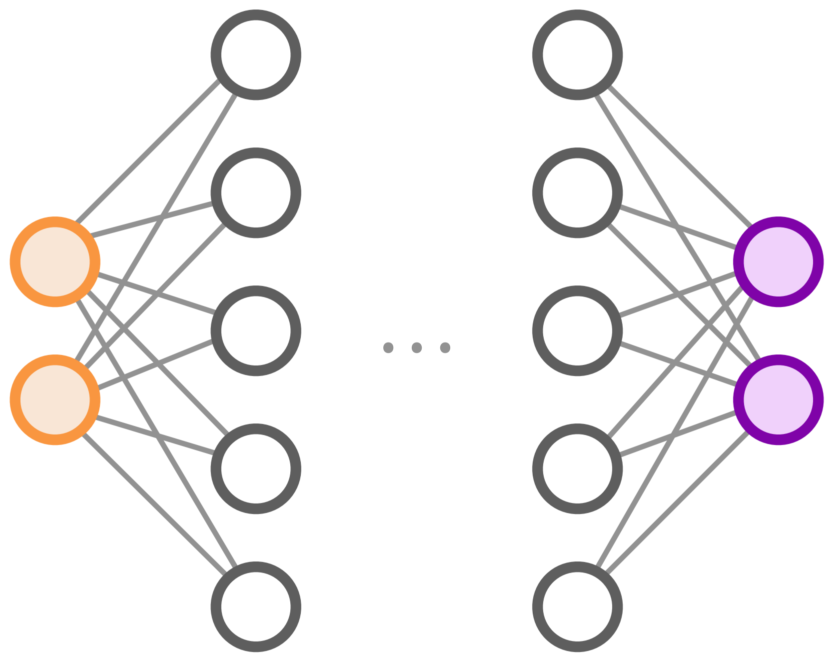
[I. Goodfellow et al 1406.2661]

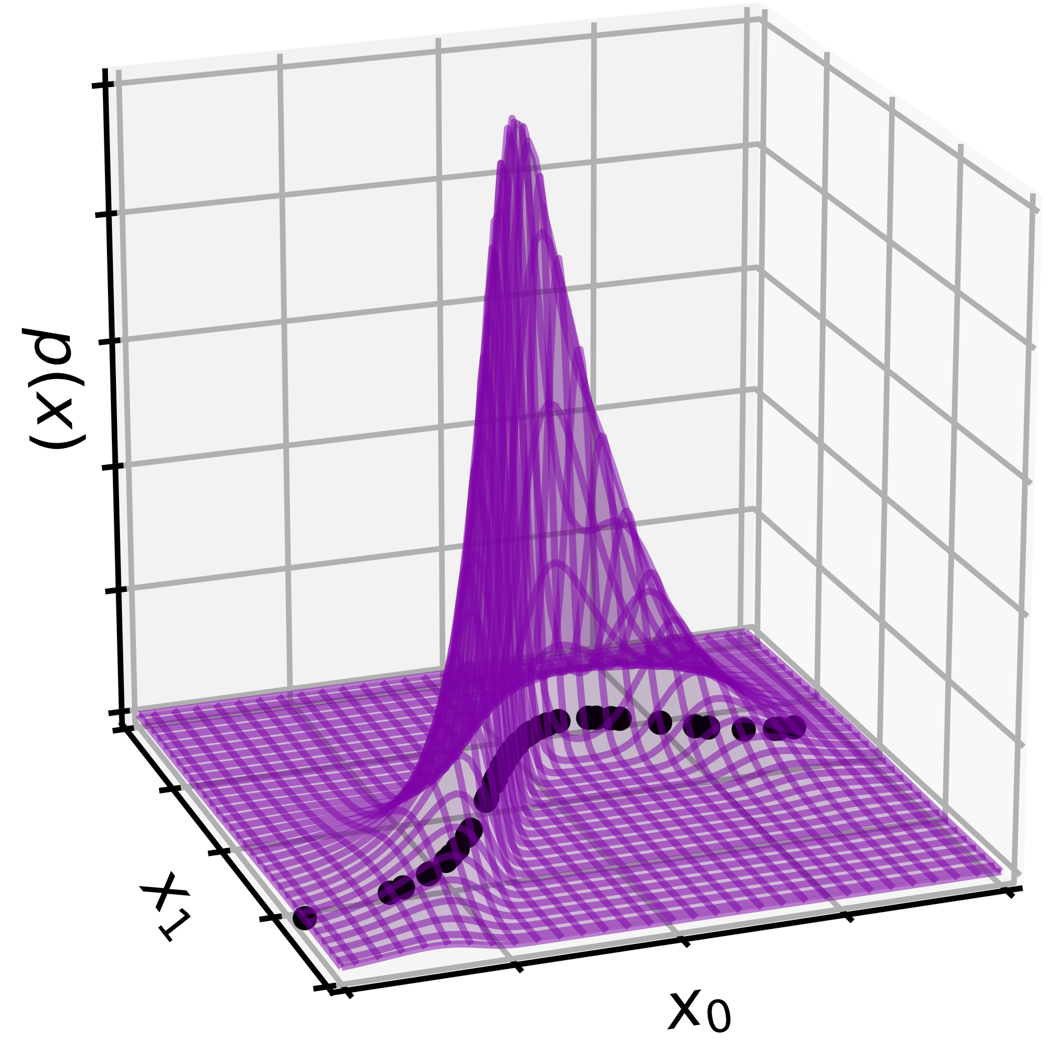


Normalizing flows in the ambient data space

[G. Papamakarios et al 1912.02762]







$u \sim p_u(u)$

f

x

d-dim. latent variables

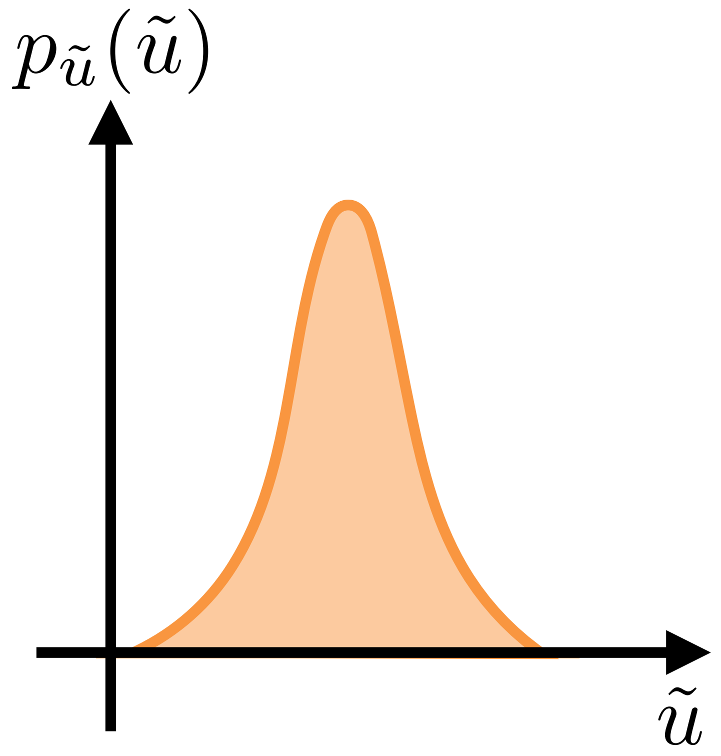
invertible NN

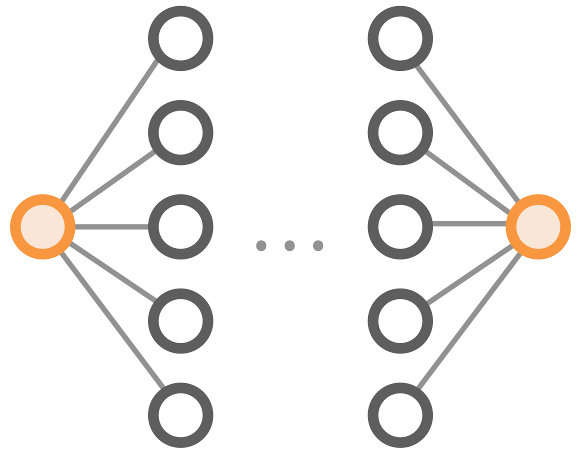
tractable density over ambient data space

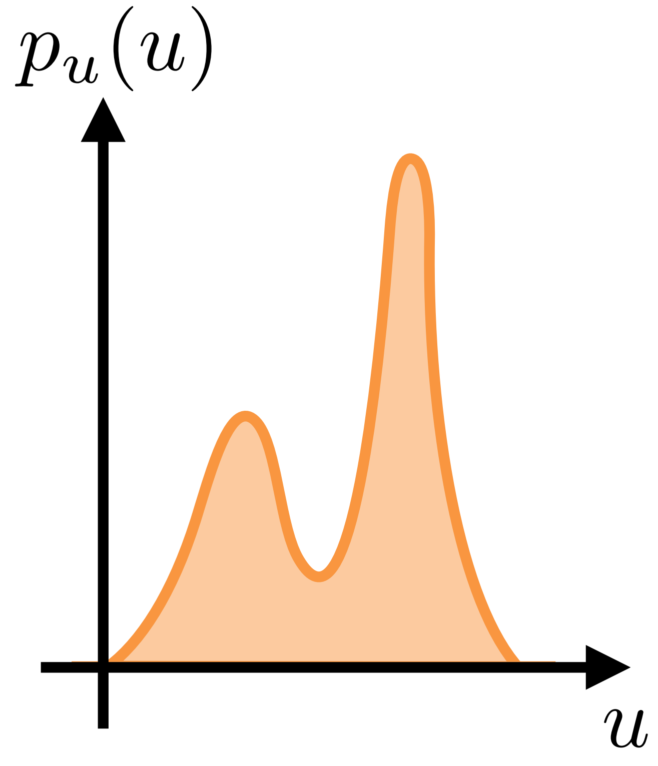
$$p_x(x) = p_u(f^{-1}(x))\ |\det J_f(f^{-1}(x))|^{-1}$$

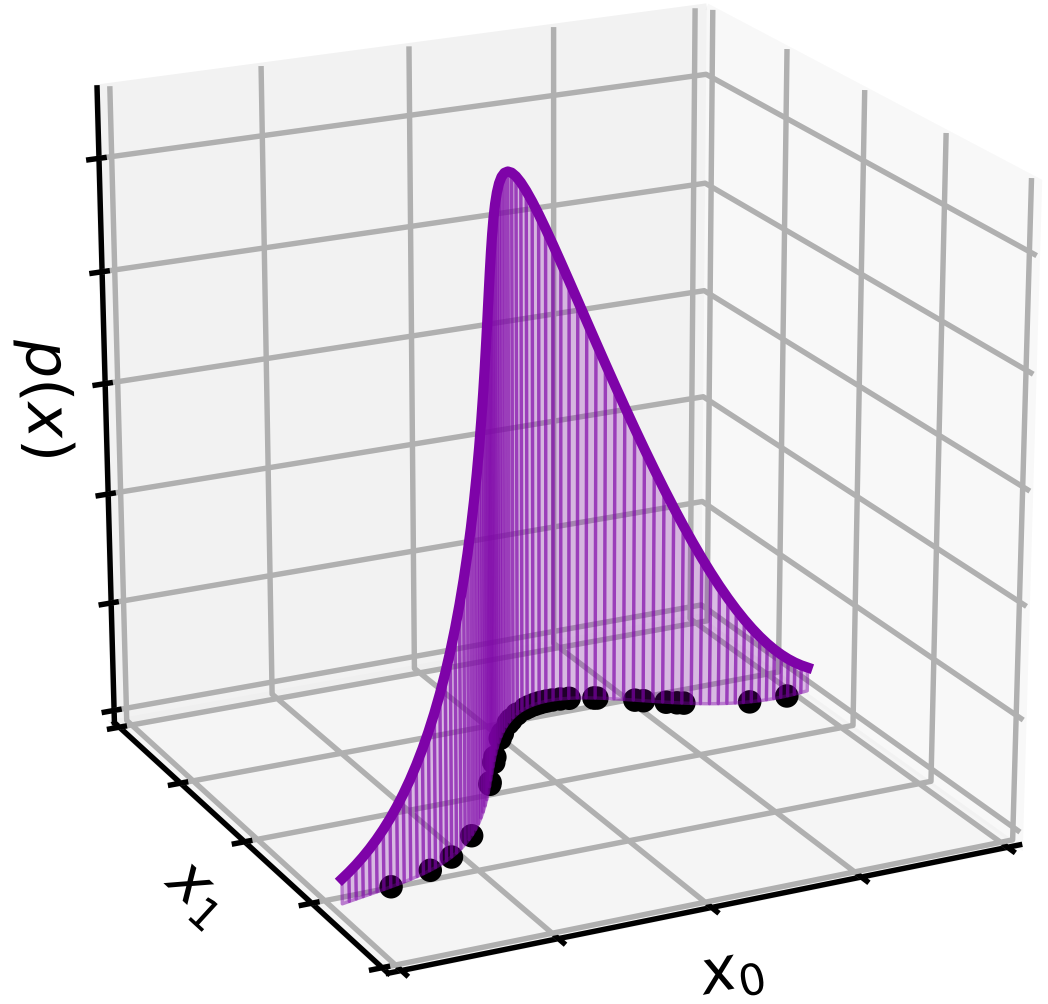
Flows on a prescribed manifold

[M. Gemici et al 1611.02304;
D. Rezende, …, M. Albergo, …, K. Cranmer 2002.02428]



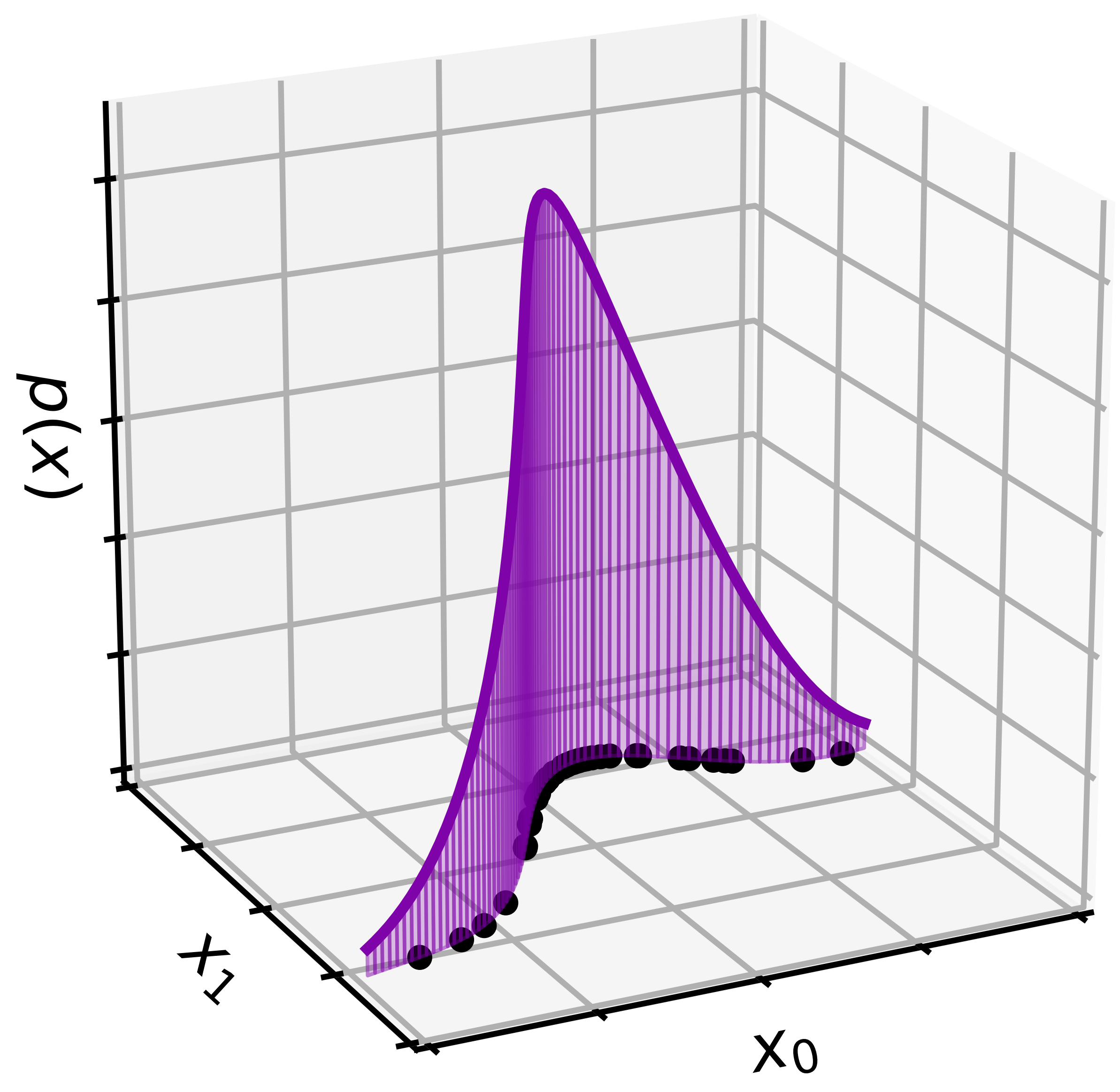






$\tilde{u} \sim p_{\tilde{u}}(\tilde{u})$	$\overset{}{\longleftrightarrow}$ h	u	$\longleftrightarrow$ g^*	x
n-dim. latents	invertible NN	n-dim. latents	prescribed chart	tractable density over $\mathcal{M}^*$

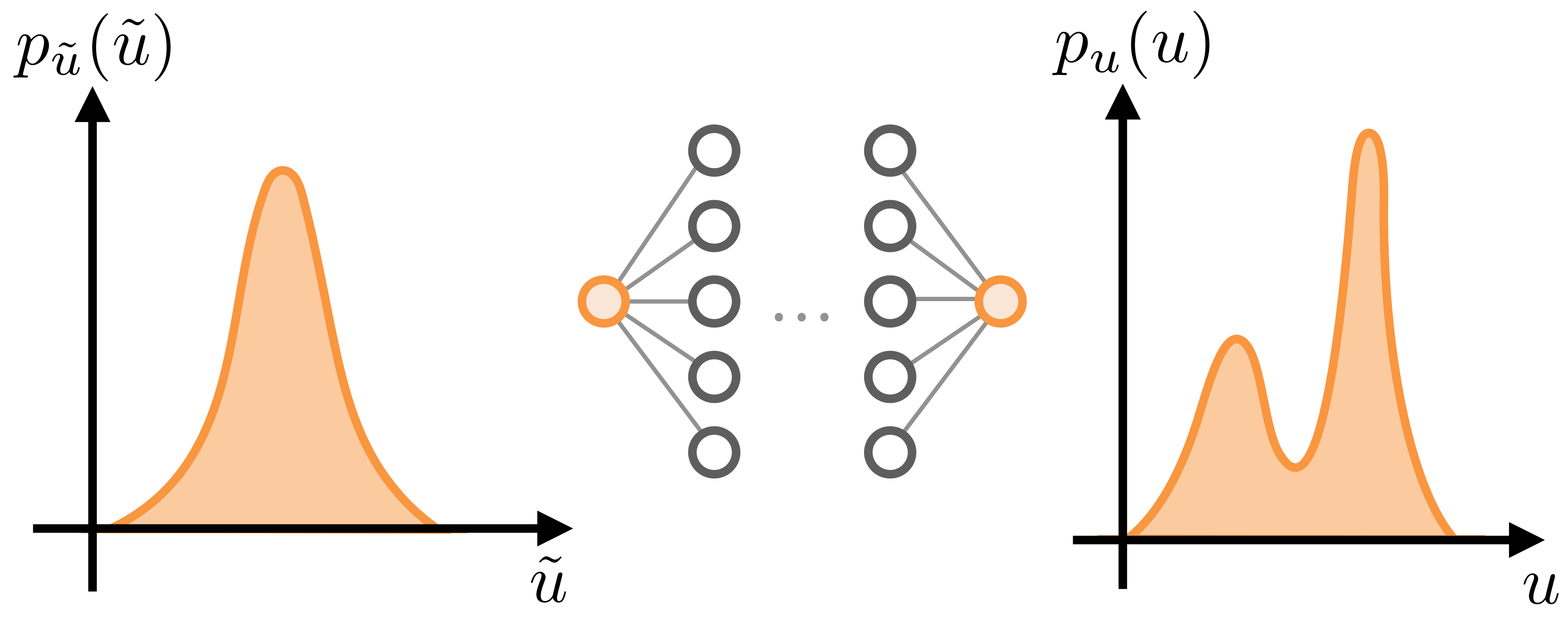
$$p_{\mathcal{M}^*}(x) = p_{\tilde{u}}(\tilde{u}) \; |\det J_h(\tilde{u})|^{-1} \cdot \left|\det[J_{g^*}^T(u) J_{g^*}(u)]\right|^{-\frac{1}{2}}$$



$\mathcal{M}$-flows

$\mathcal{M}$-flows

[JB, Kyle Cranmer 2003.13913]

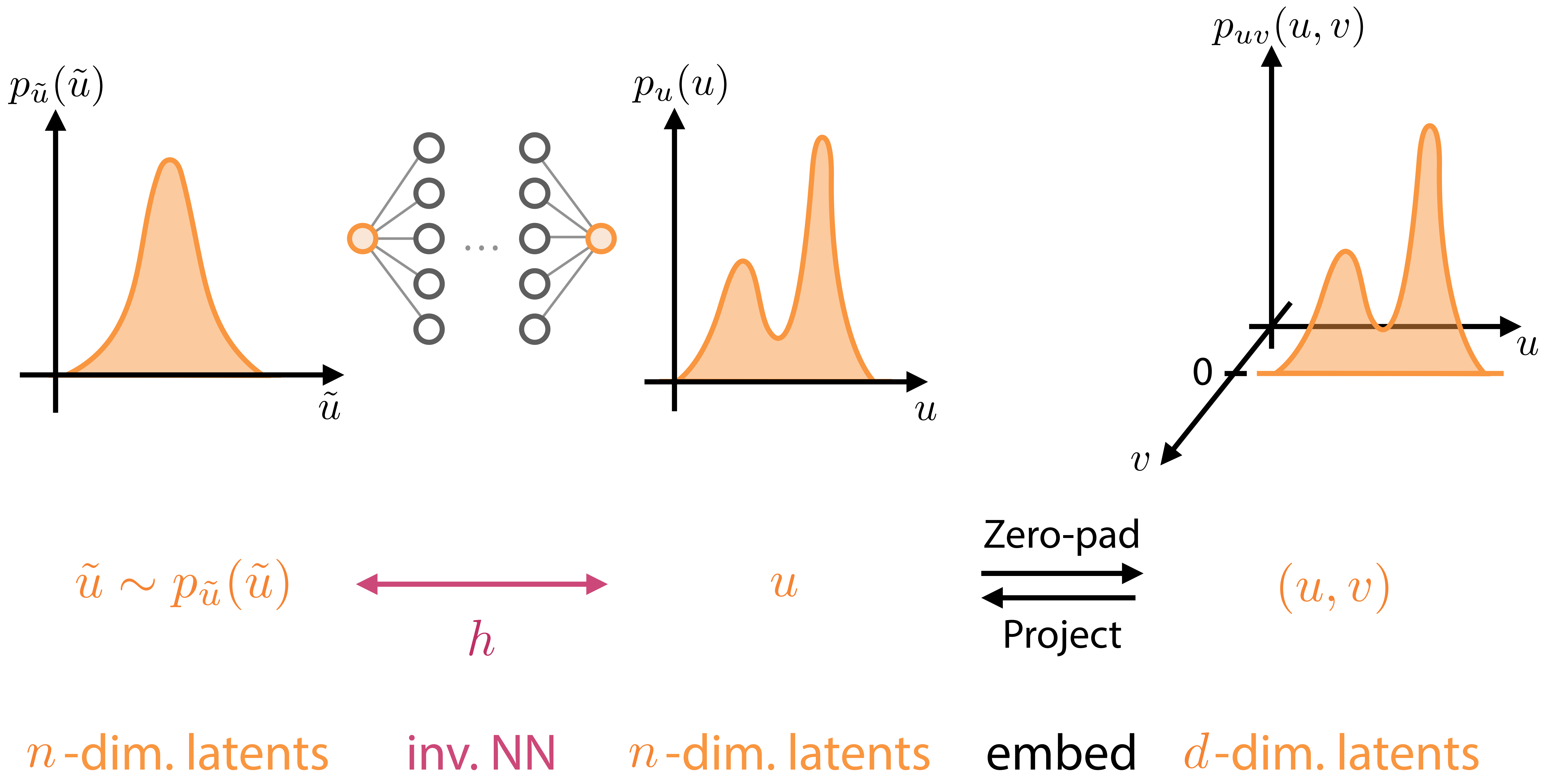


$\tilde{u} \sim p_{\tilde{u}}(\tilde{u})$ $\xleftrightarrow[h]{}$ u

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

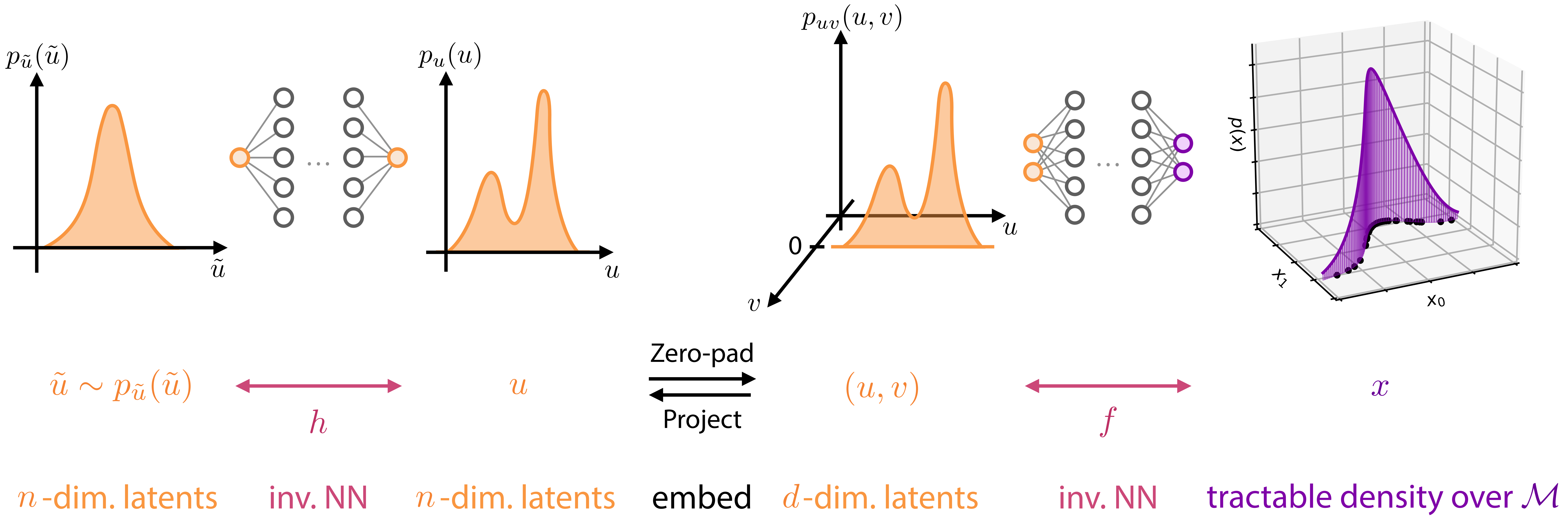
$\mathcal{M}$-flows

[JB, Kyle Cranmer 2003.13913]



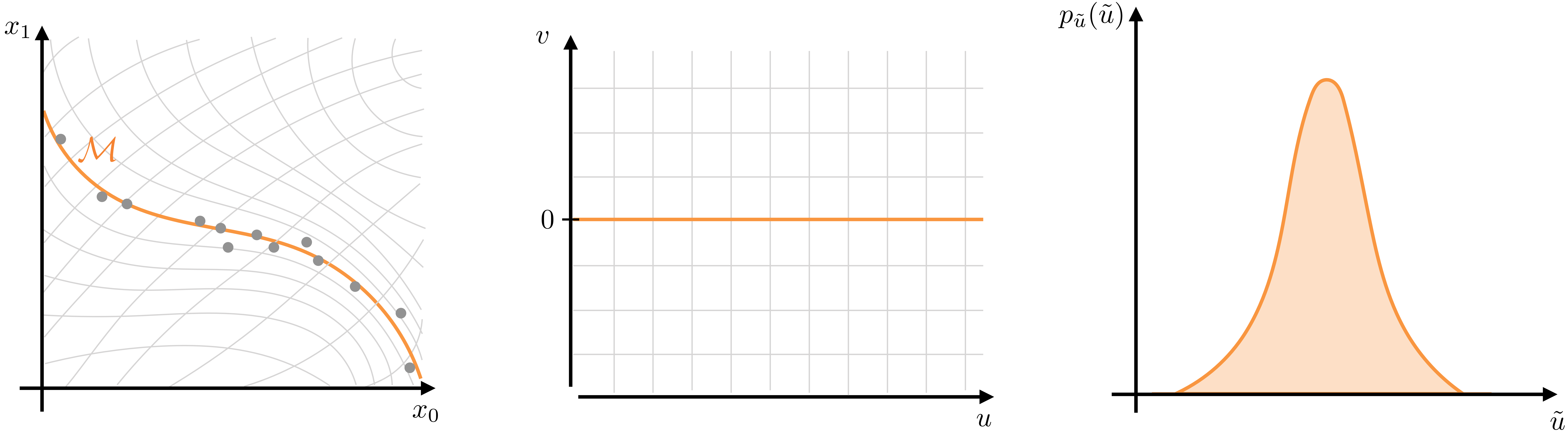
$\mathcal{M}$-flows

[JB, Kyle Cranmer 2003.13913]

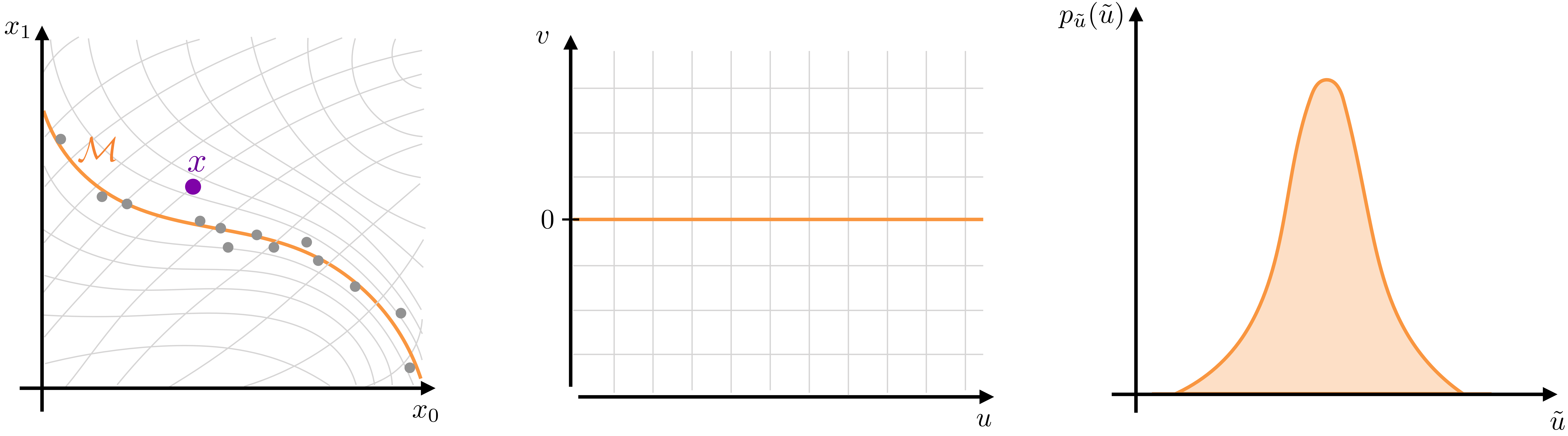


$$p_{\mathcal{M}}(x) = p_{\tilde u}(\tilde u)\,|\det J_h(\tilde u)|^{-1} \cdot \left|\det\left[\begin{pmatrix}\mathbb{1} & 0\end{pmatrix} J_f(u)^T J_f(u) \begin{pmatrix}\mathbb{1}\\ 0\end{pmatrix}\right]\right|^{-\frac{1}{2}}$$

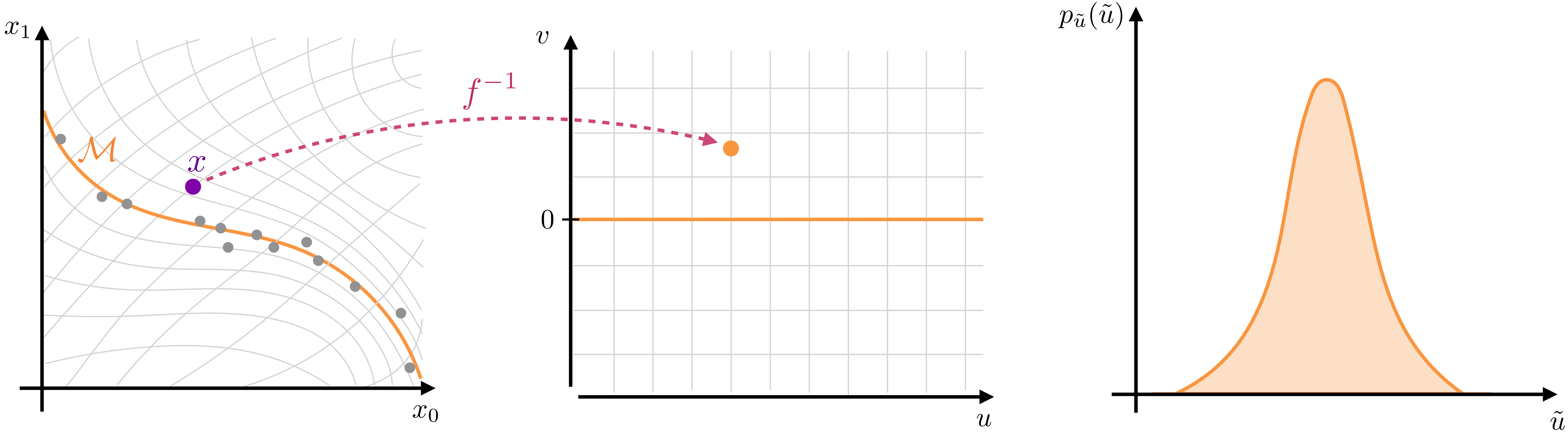
Evaluating data on or off the manifold



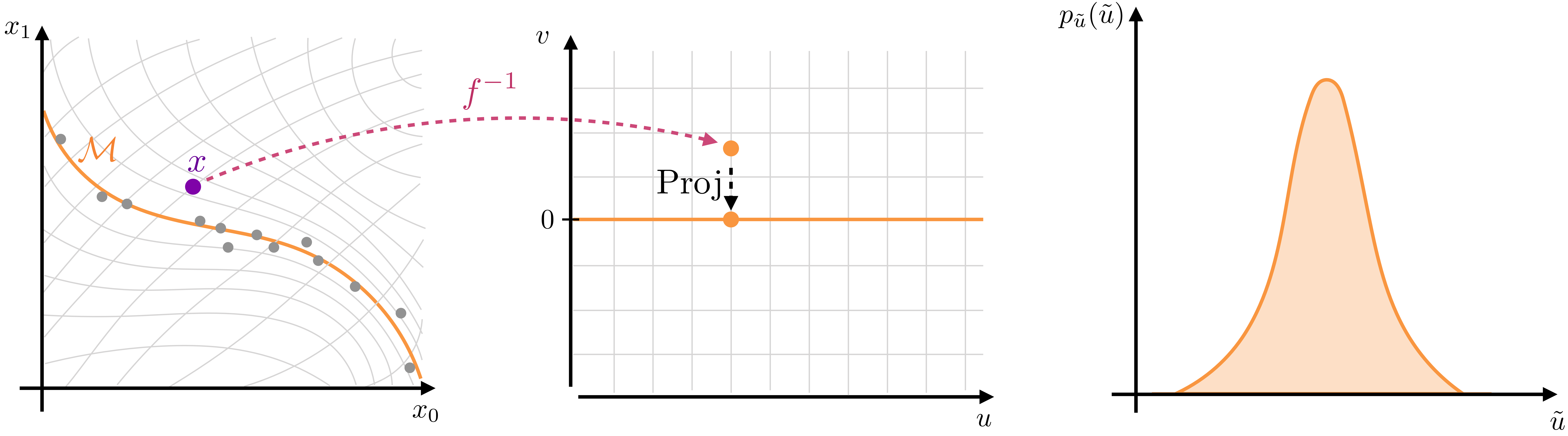
Evaluating data on or off the manifold



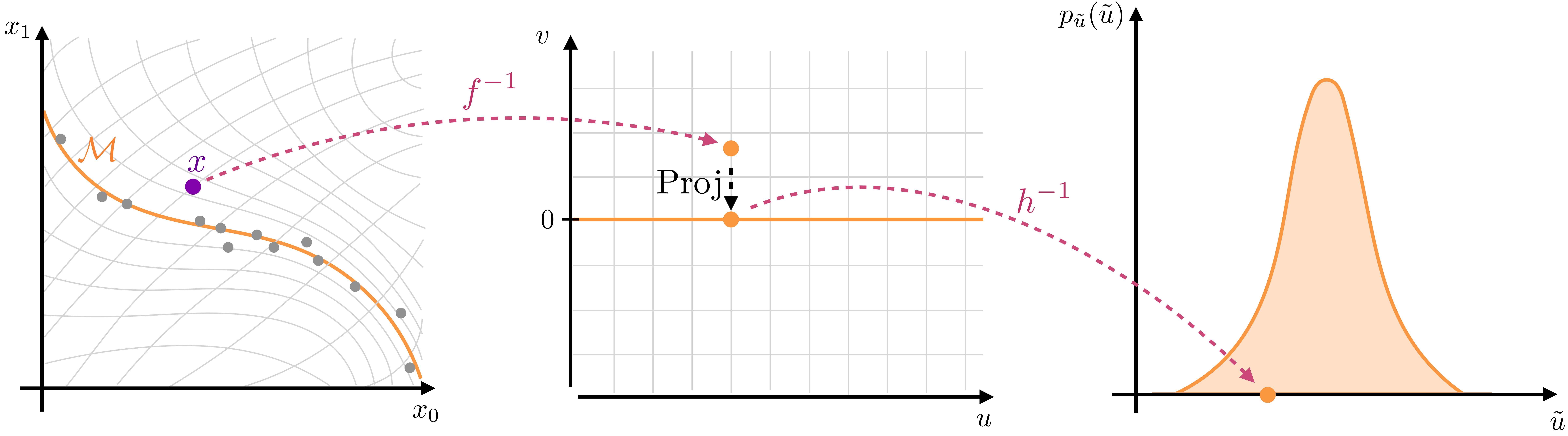
Evaluating data on or off the manifold



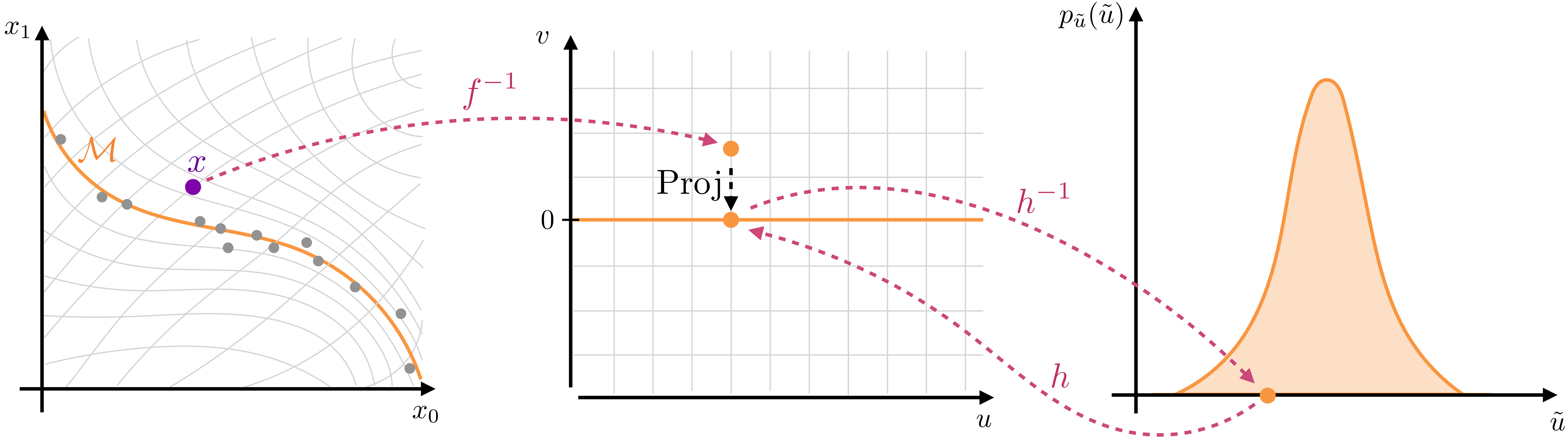
Evaluating data on or off the manifold



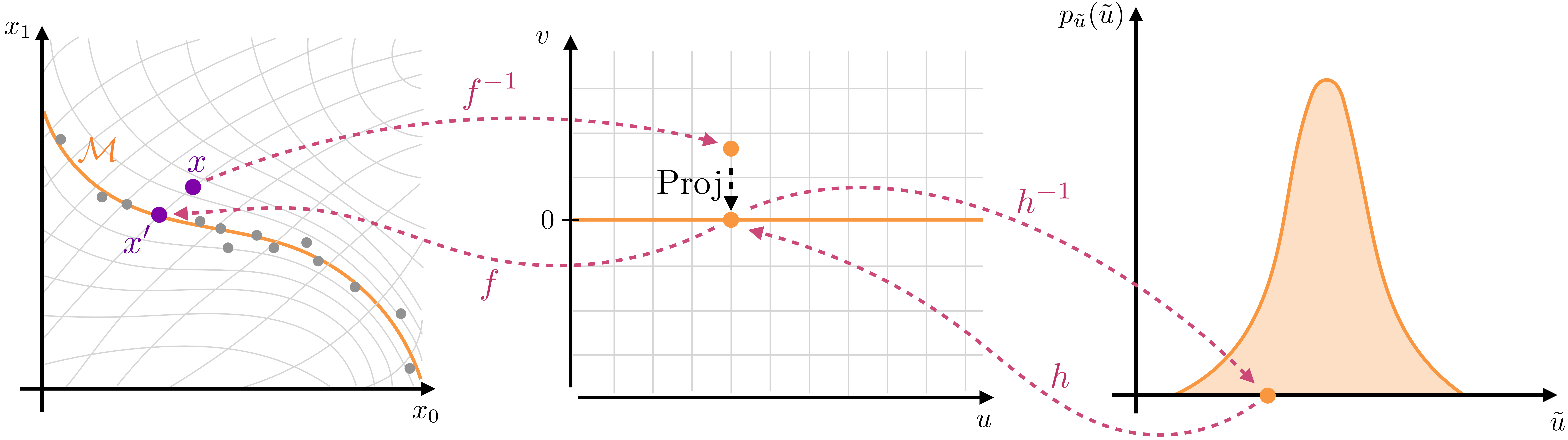
Evaluating data on or off the manifold



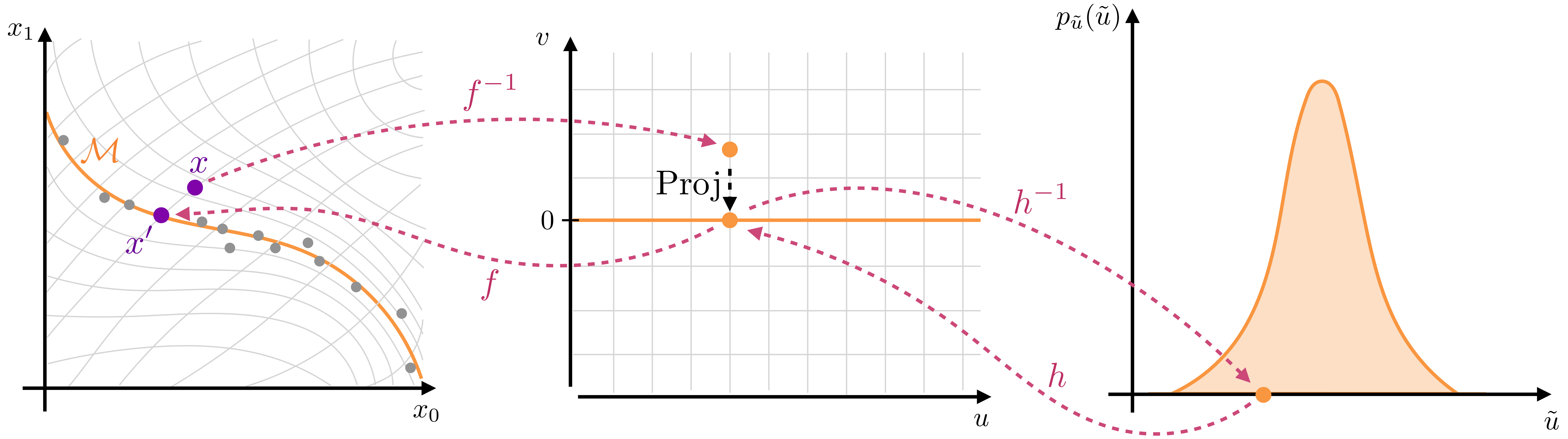
Evaluating data on or off the manifold



Evaluating data on or off the manifold



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

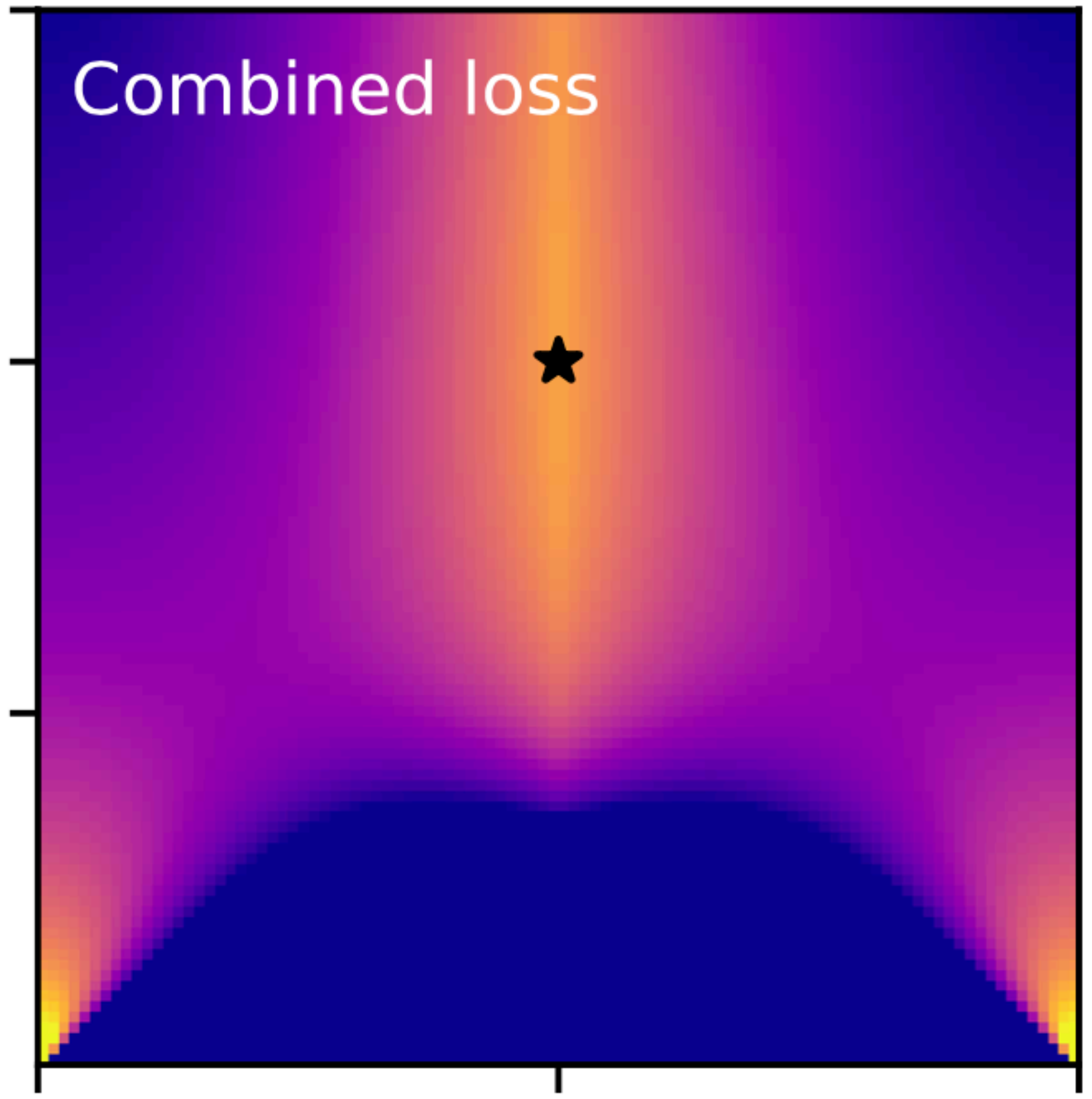
- Pseudo-Invertible Encoder (PIE):
  add "narrow" base density $p(v)$ for off-the-manifold latents,
  likelihood over ambient space (inconsistent with sampling)
  [J. J. Beitler, I. Sosnovik, A. Smeulders 2018]

- 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 ϕ_f

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

$\Rightarrow$ Learning ϕ_f by maximum

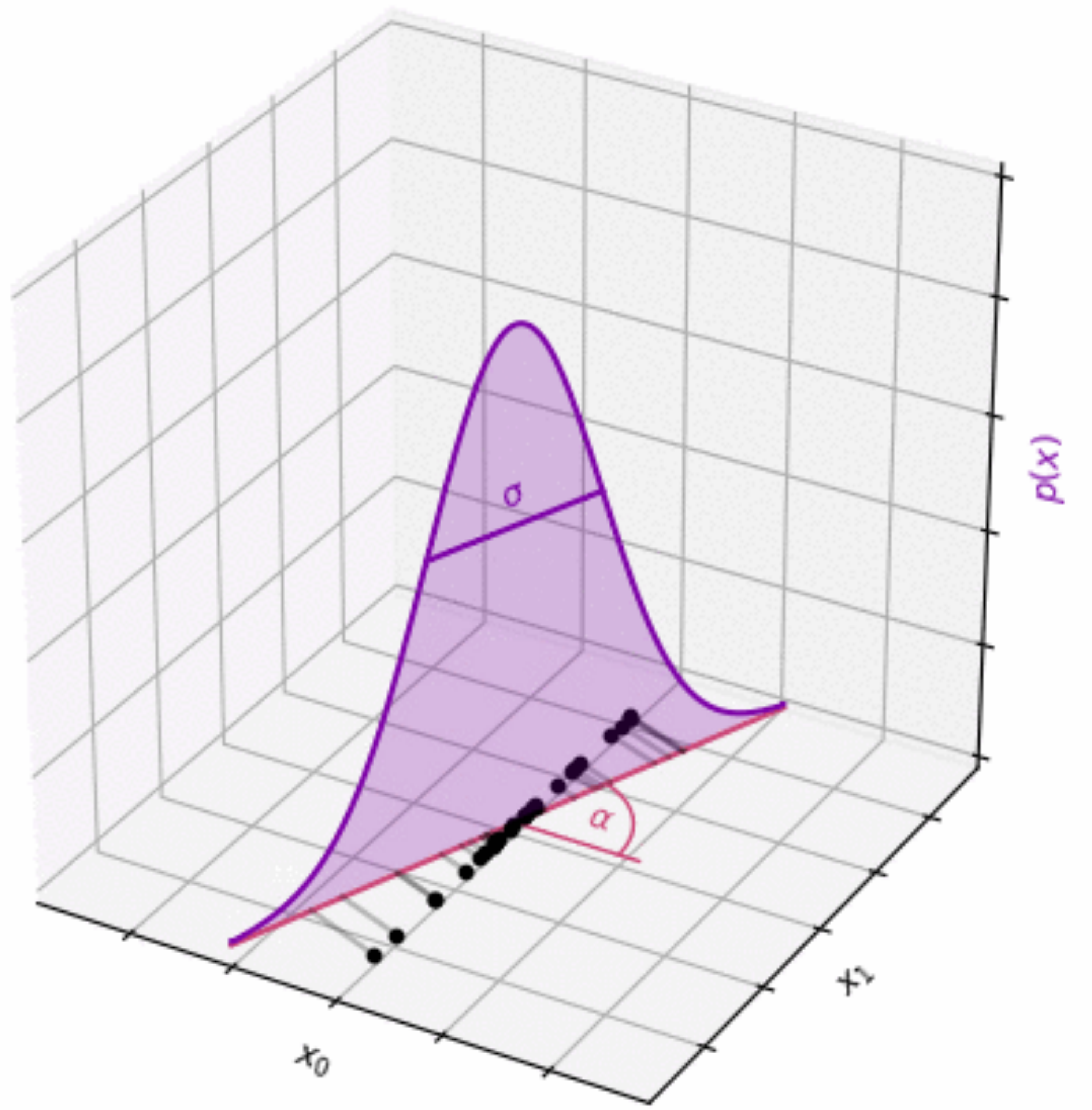
likelihood is unstable

Maximum likelihood is not enough

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

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

⇒ Learning ϕ_f by maximum likelihood is unstable

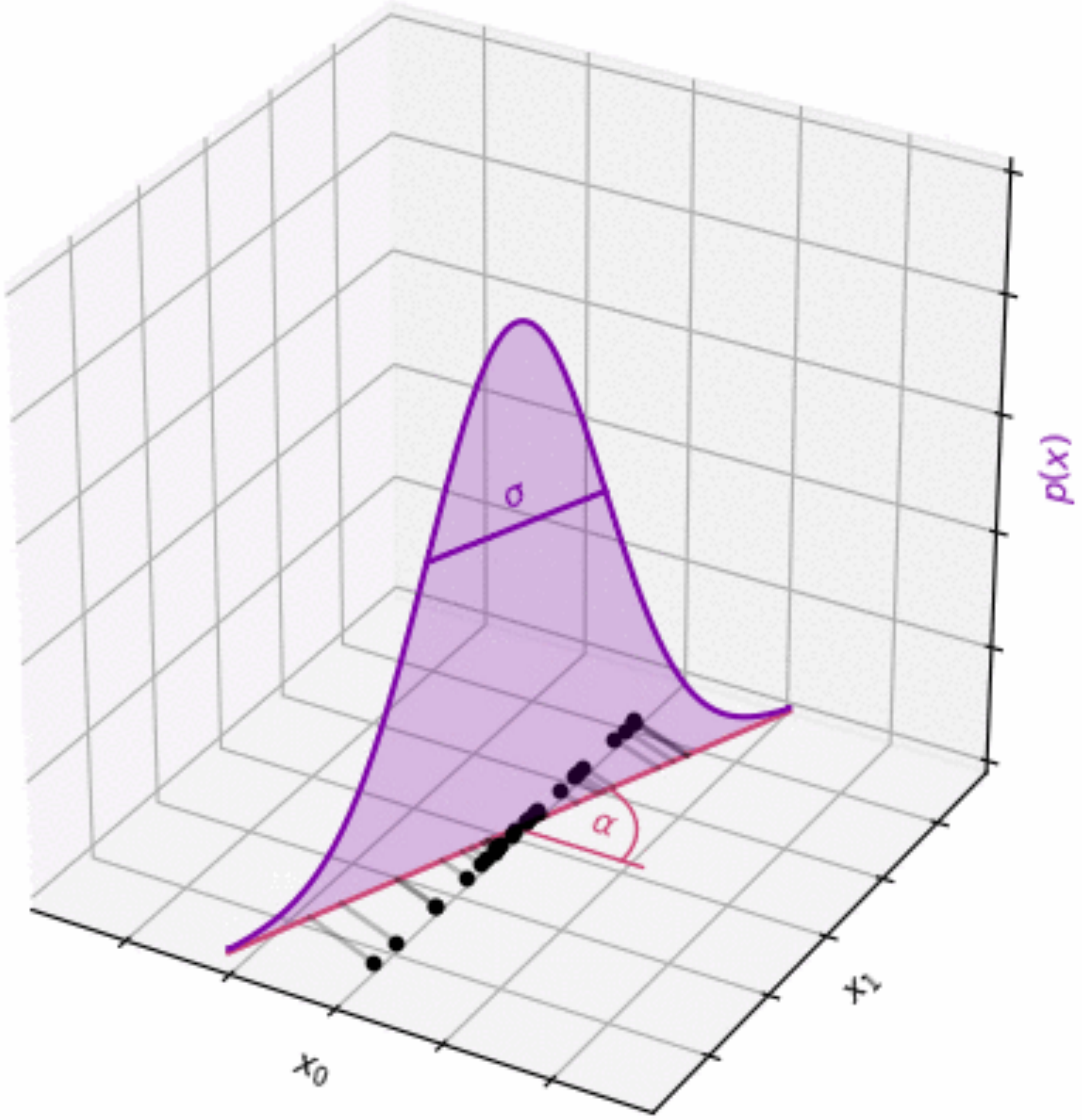


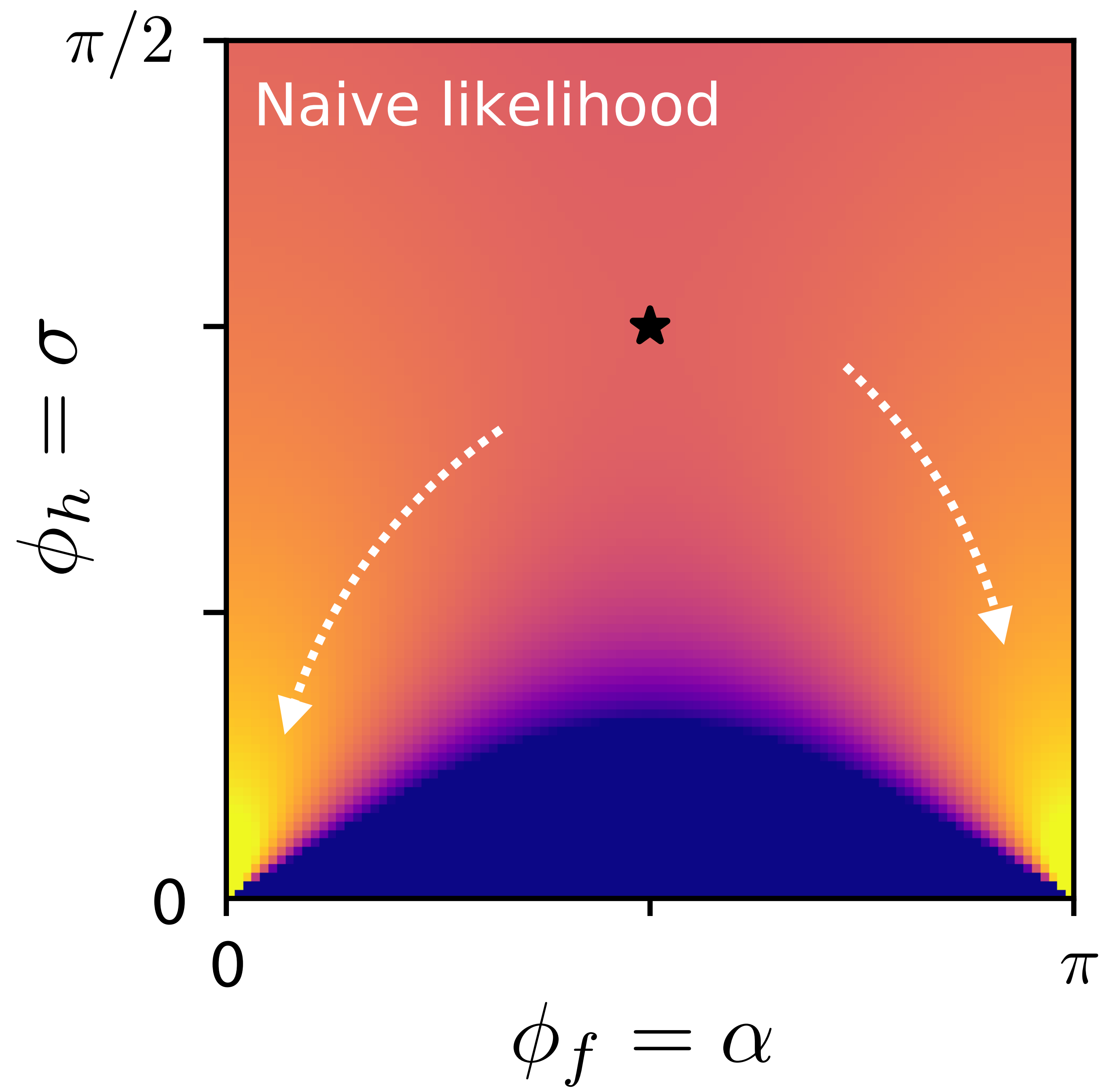
Maximum likelihood is not enough

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

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

$\Rightarrow$ Learning ϕ_f by maximum likelihood is unstable

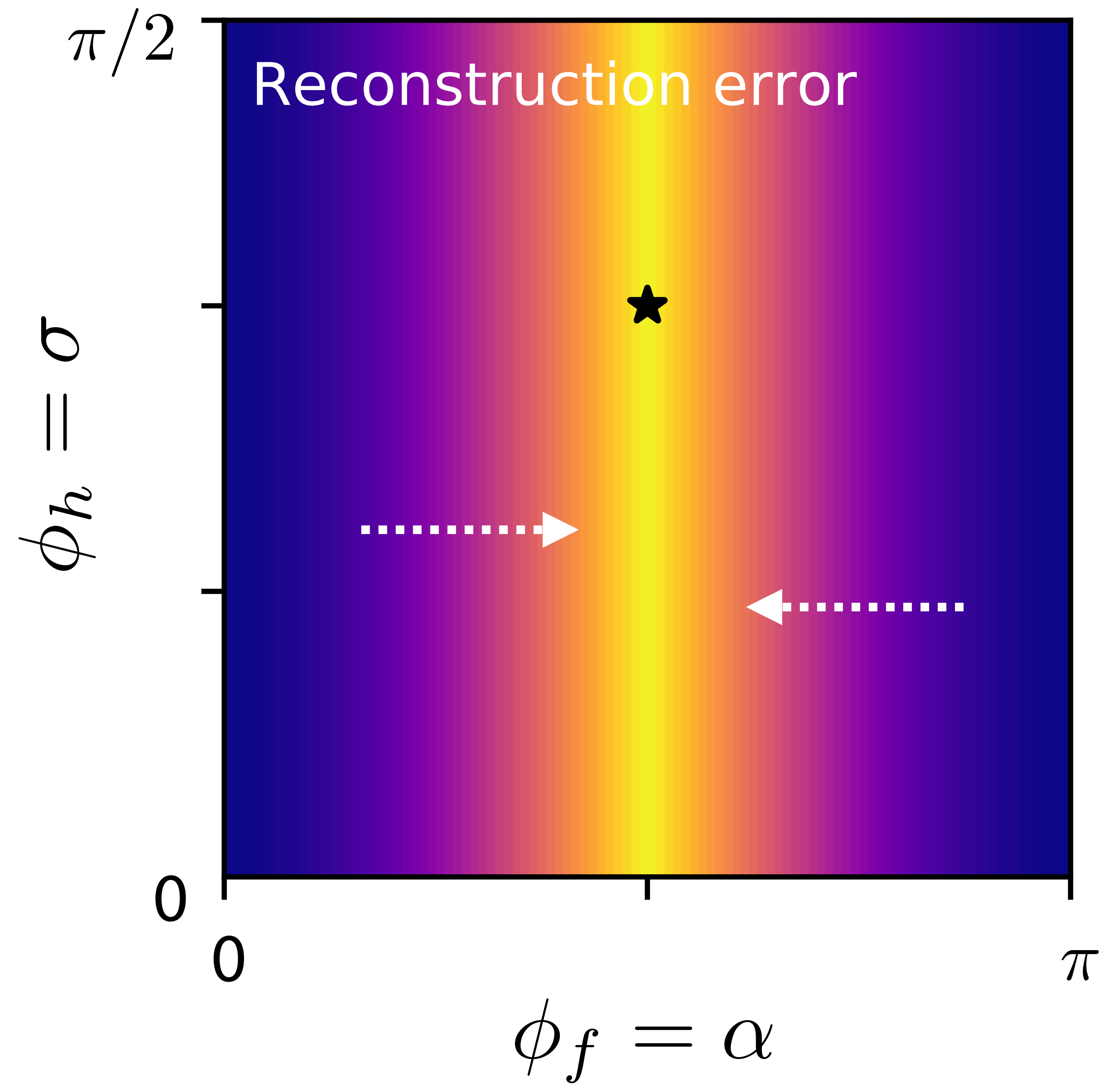




M/D training

Solution: separate training in two phases!

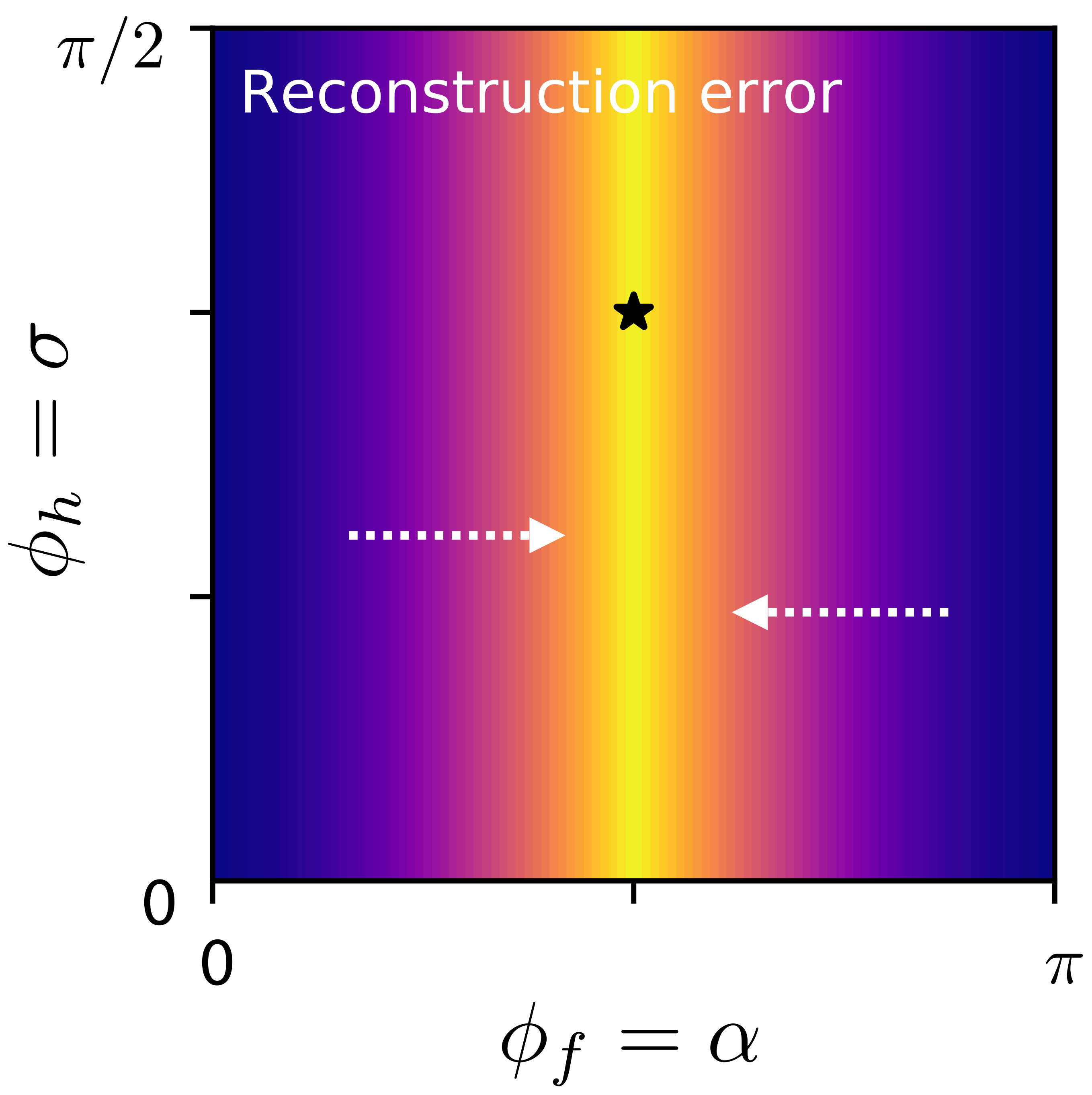
- Manifold phase:
  update ϕ_f (and thus $\mathcal{M}$) by minimizing $\|x - x'\|$



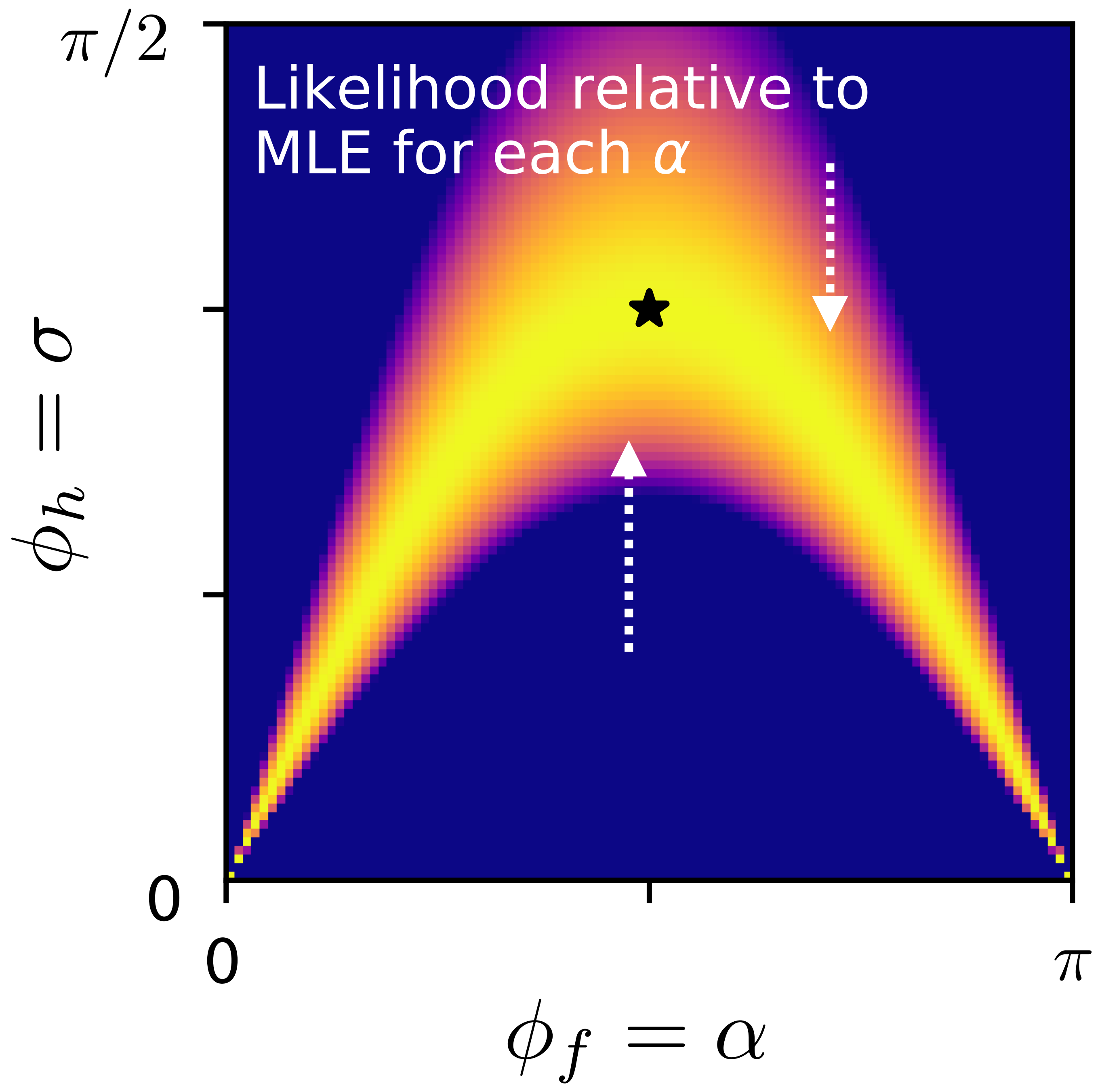
M/D training

Solution: separate training in two phases!

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



- Density phase:
  update ϕ_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:

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

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:

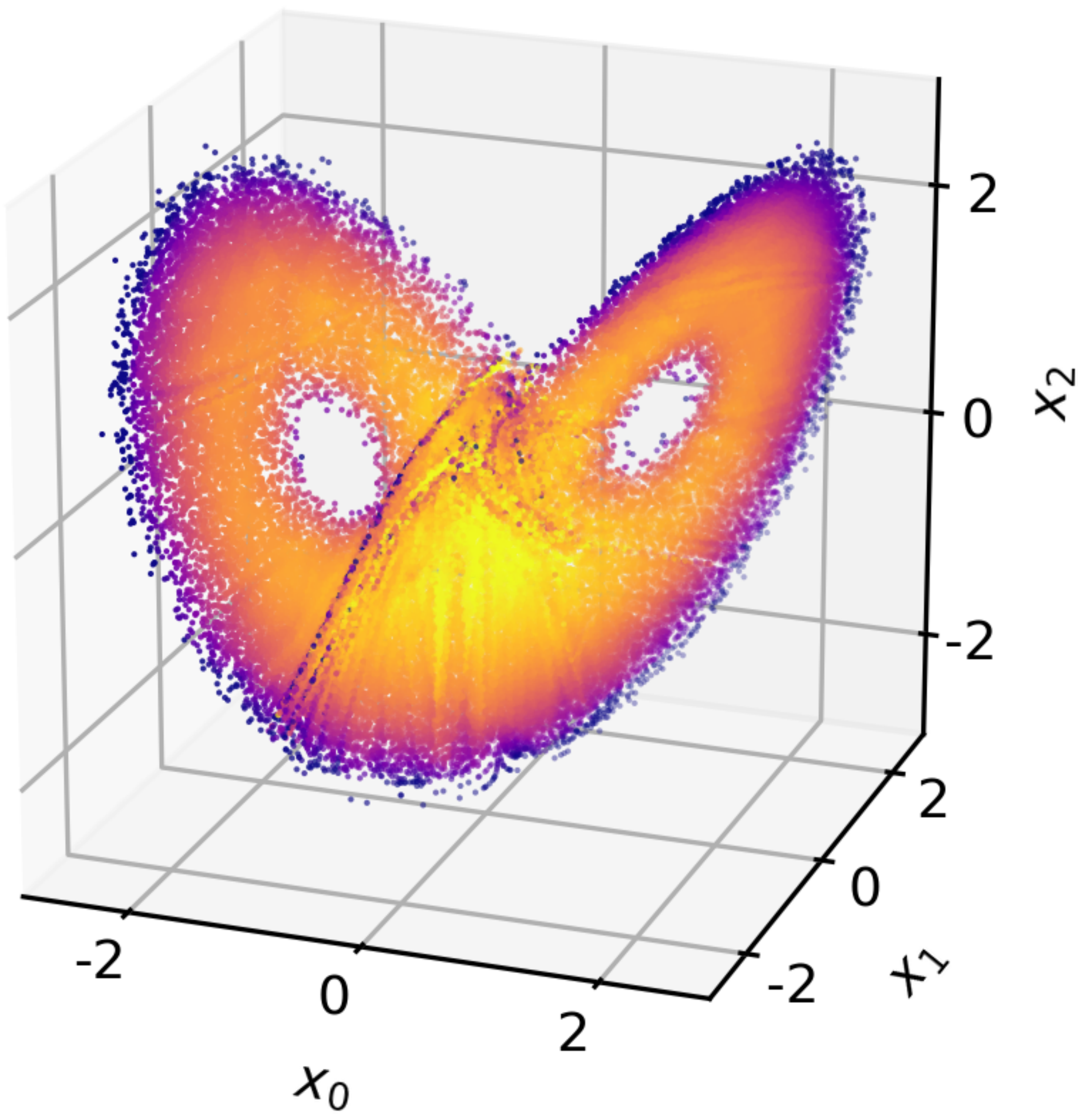
$$\log p_{\mathcal{M}}(x) = \log p_{\tilde{u}}(h^{-1}(u)) - \log\det J_h(h^{-1}(u)) - \frac{1}{2}\log\det\left[\begin{pmatrix}\mathbb{1} & 0\end{pmatrix} J_f^T(u)\, J_f(u) \begin{pmatrix}\mathbb{1}\\ 0\end{pmatrix}\right]$$

Cannot separate determinant of product of non-square matrices

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

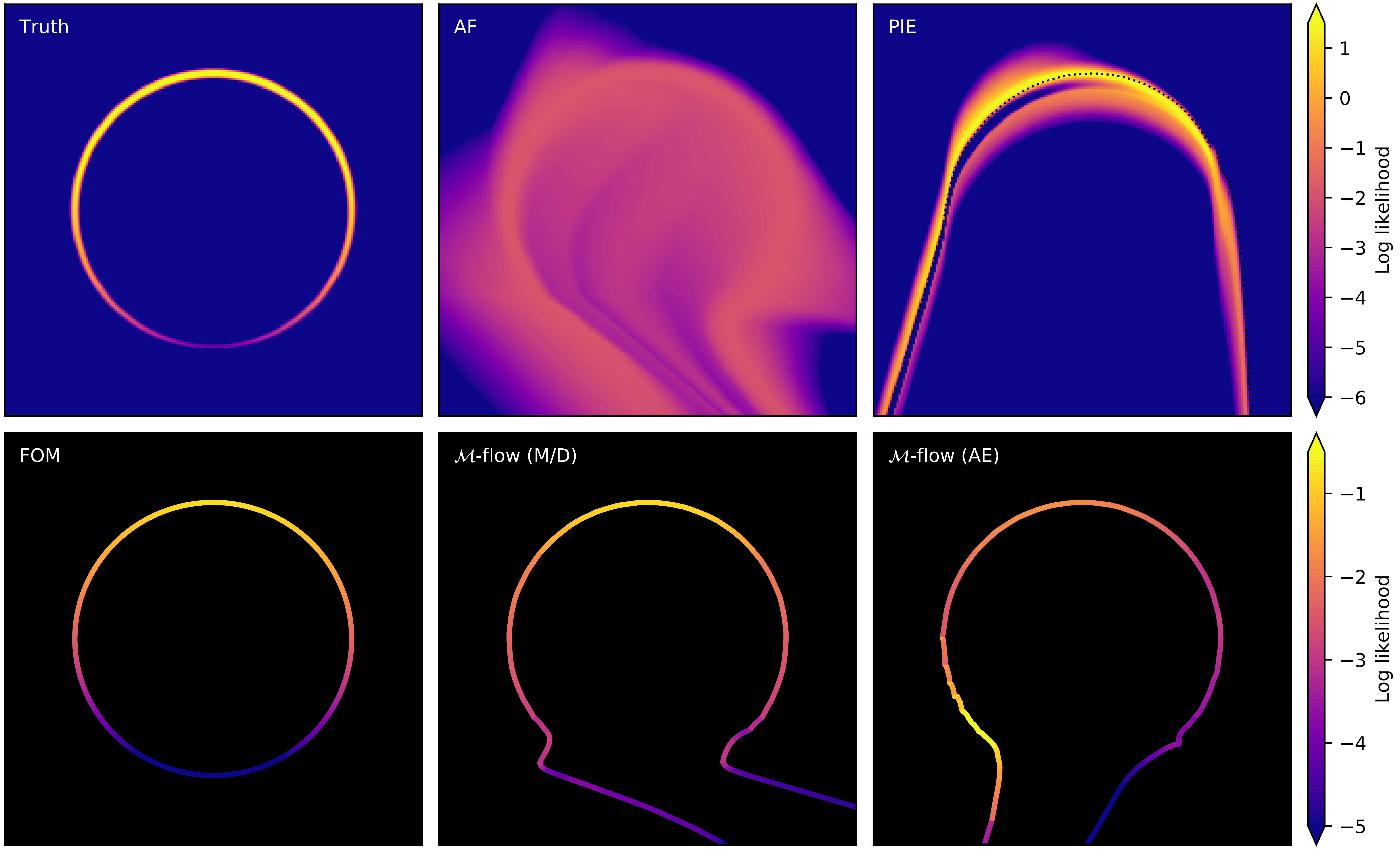
$$\nabla_{\phi_h}\left(\log p_{\mathcal{M}}(x)\right) = \nabla_{\phi_h}\left(\log p_{\tilde{u}}(h^{-1}(u))\right) - \nabla_{\phi_h}\left(\log\det J_h(h^{-1}(u))\right) - \underbrace{\nabla_{\phi_h}\frac{1}{2}\log\det\left[\begin{pmatrix}\mathbb{1} & 0\end{pmatrix} J_f^T(u)\, J_f(u)\begin{pmatrix}\mathbb{1}\\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

which can be computed efficiently!

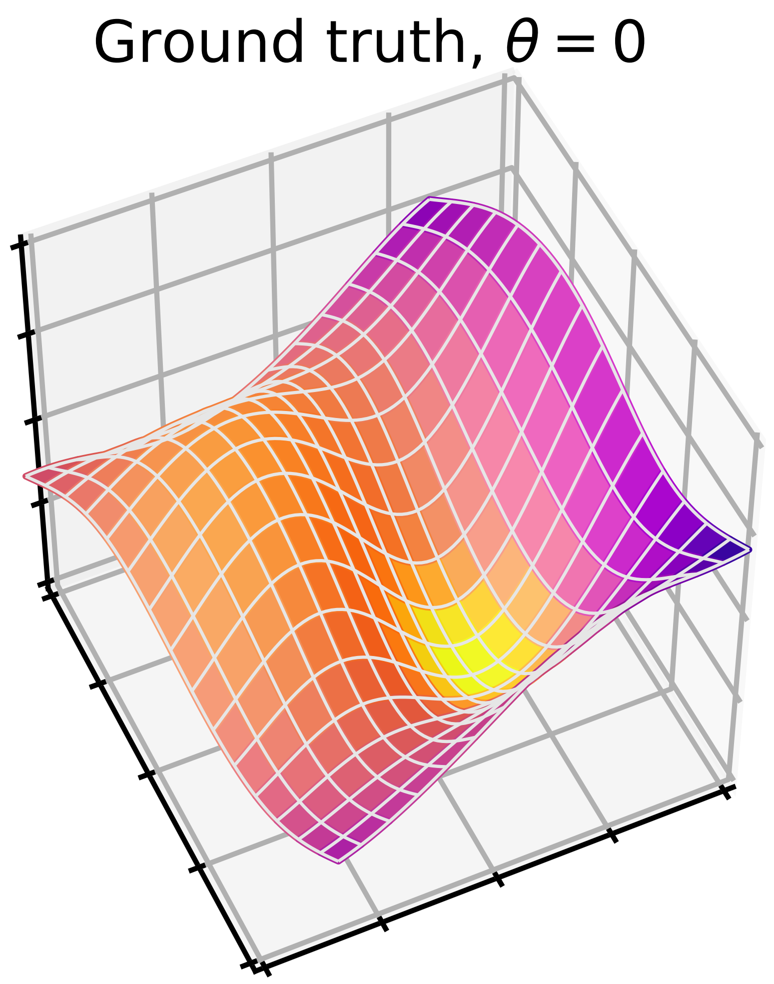


Experiments

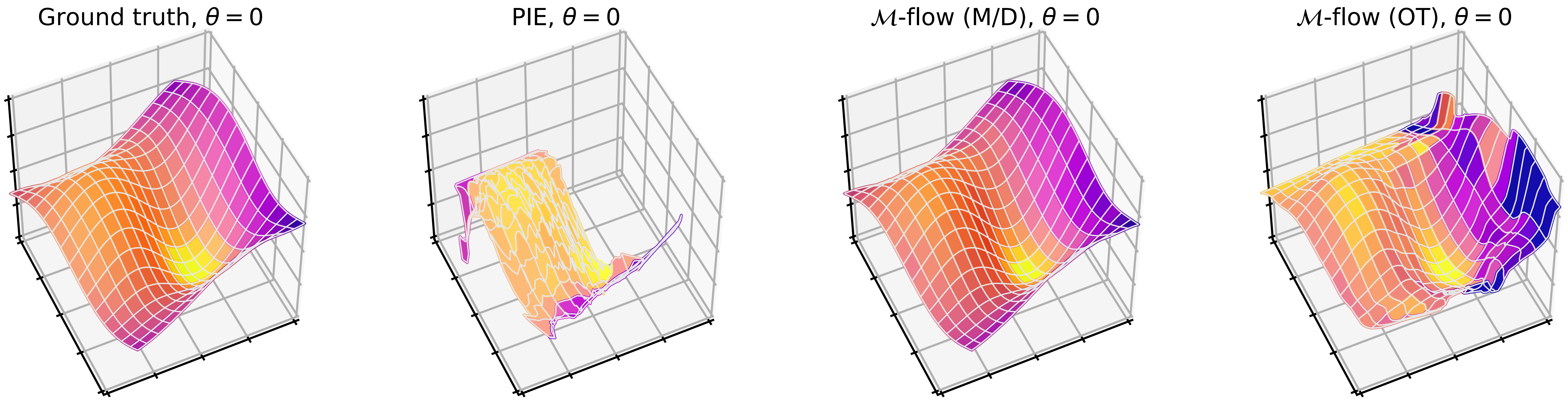
Gaussian on a circle



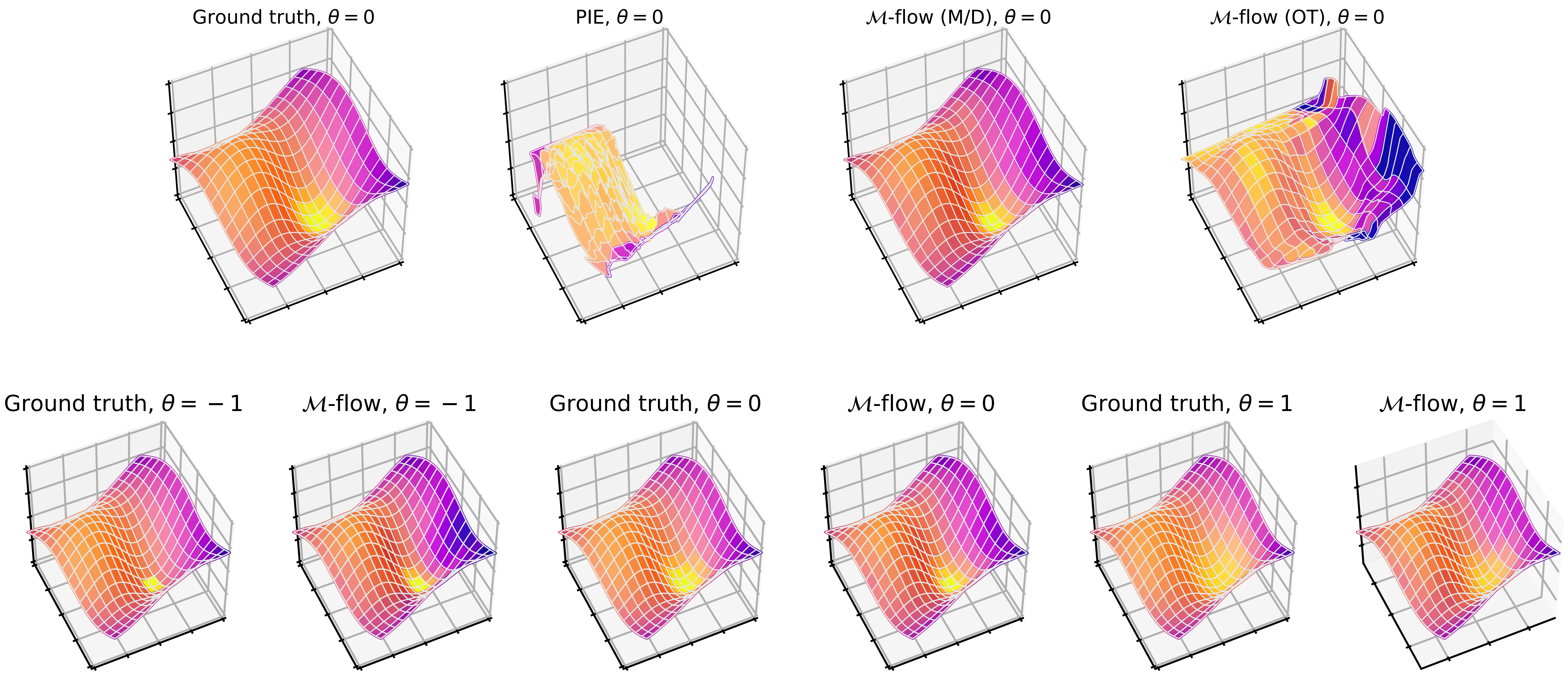
Mixture model on a polynomial surface



Mixture model on a polynomial surface

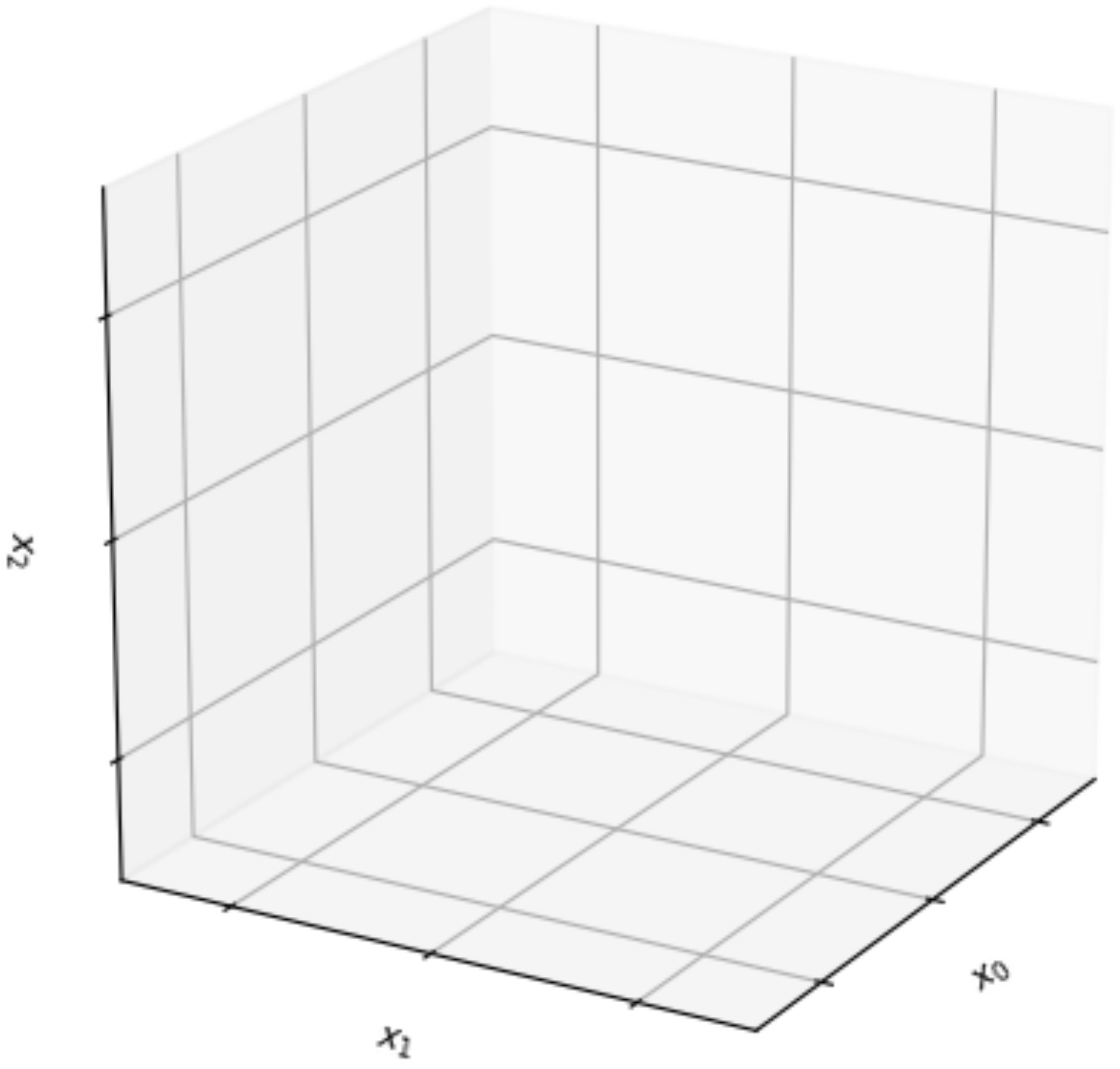


Mixture model on a polynomial surface



Lorenz attractor

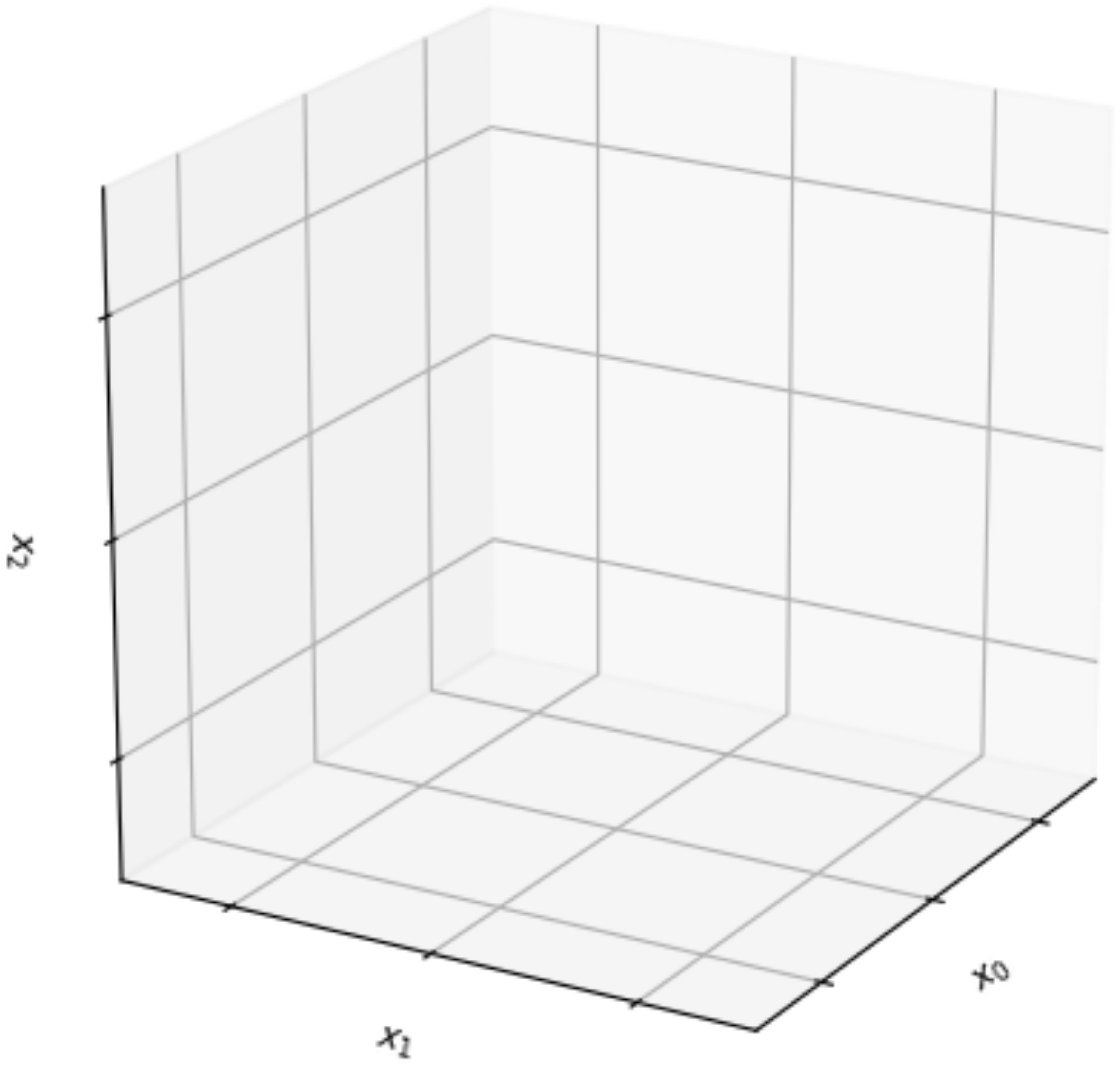
[E. Lorenz 1963]



$$\frac{\mathrm{d}x_0}{\mathrm{d}t} = \sigma(x_1 - x_0)\,, \quad \frac{\mathrm{d}x_1}{\mathrm{d}t} = x_0(\rho - x_2) - x_1\,, \quad \frac{\mathrm{d}x_2}{\mathrm{d}t} = x_0 x_1 - \beta x_2\,.$$

Lorenz attractor

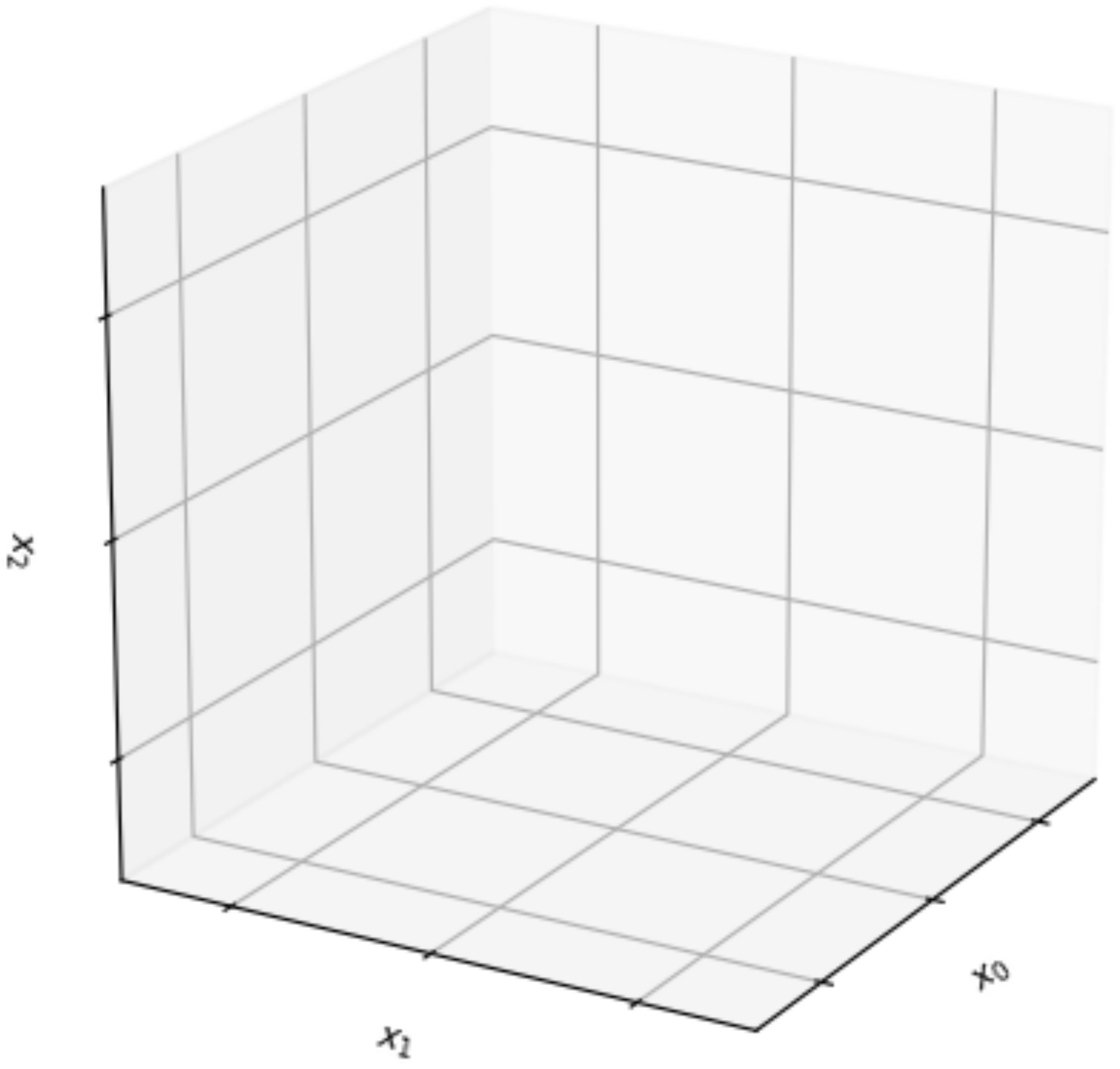
[E. Lorenz 1963]



$$\frac{\mathrm{d}x_0}{\mathrm{d}t} = \sigma(x_1 - x_0)\,, \quad \frac{\mathrm{d}x_1}{\mathrm{d}t} = x_0(\rho - x_2) - x_1\,, \quad \frac{\mathrm{d}x_2}{\mathrm{d}t} = x_0 x_1 - \beta x_2\,.$$

Lorenz attractor

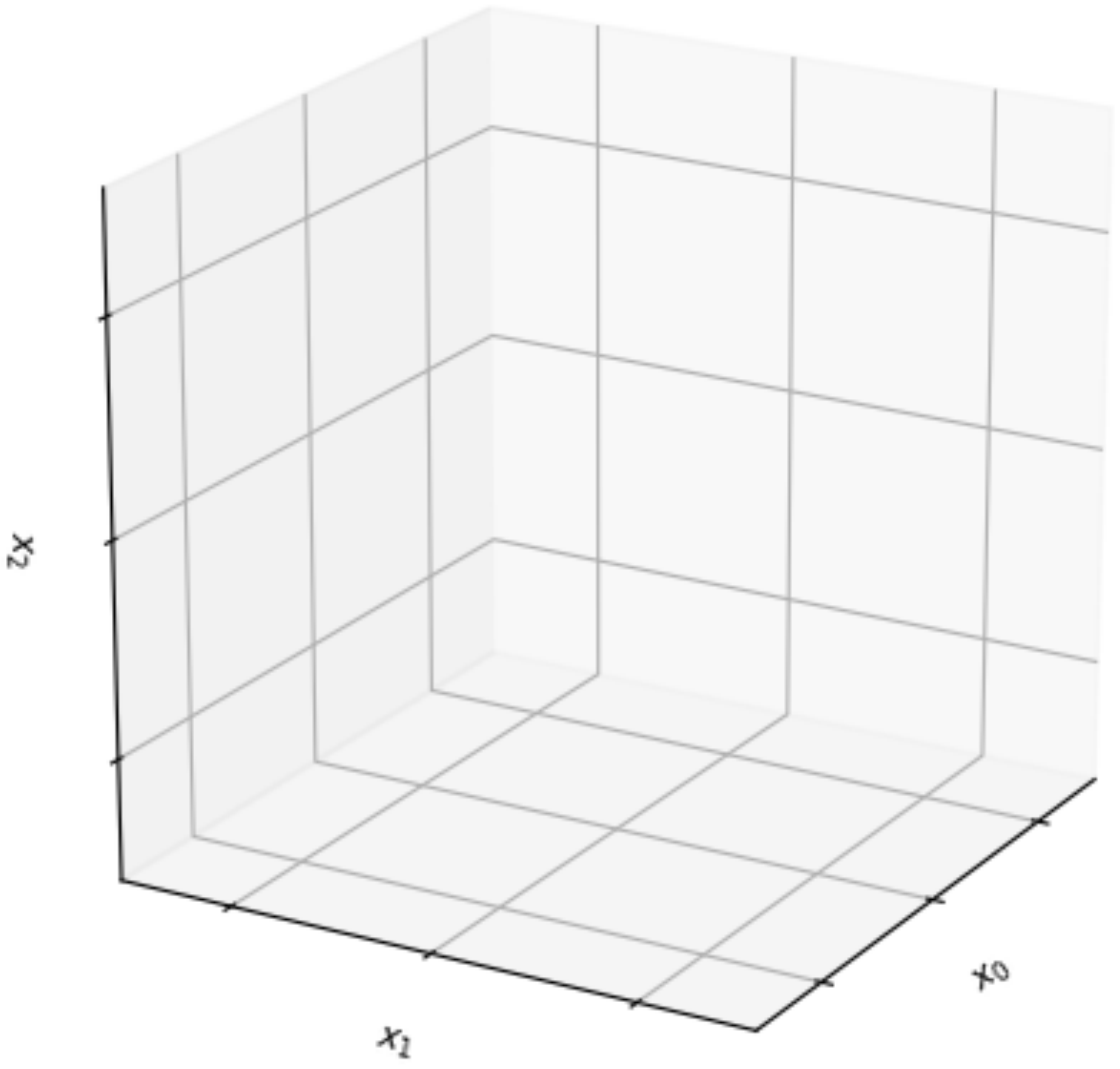
[E. Lorenz 1963]

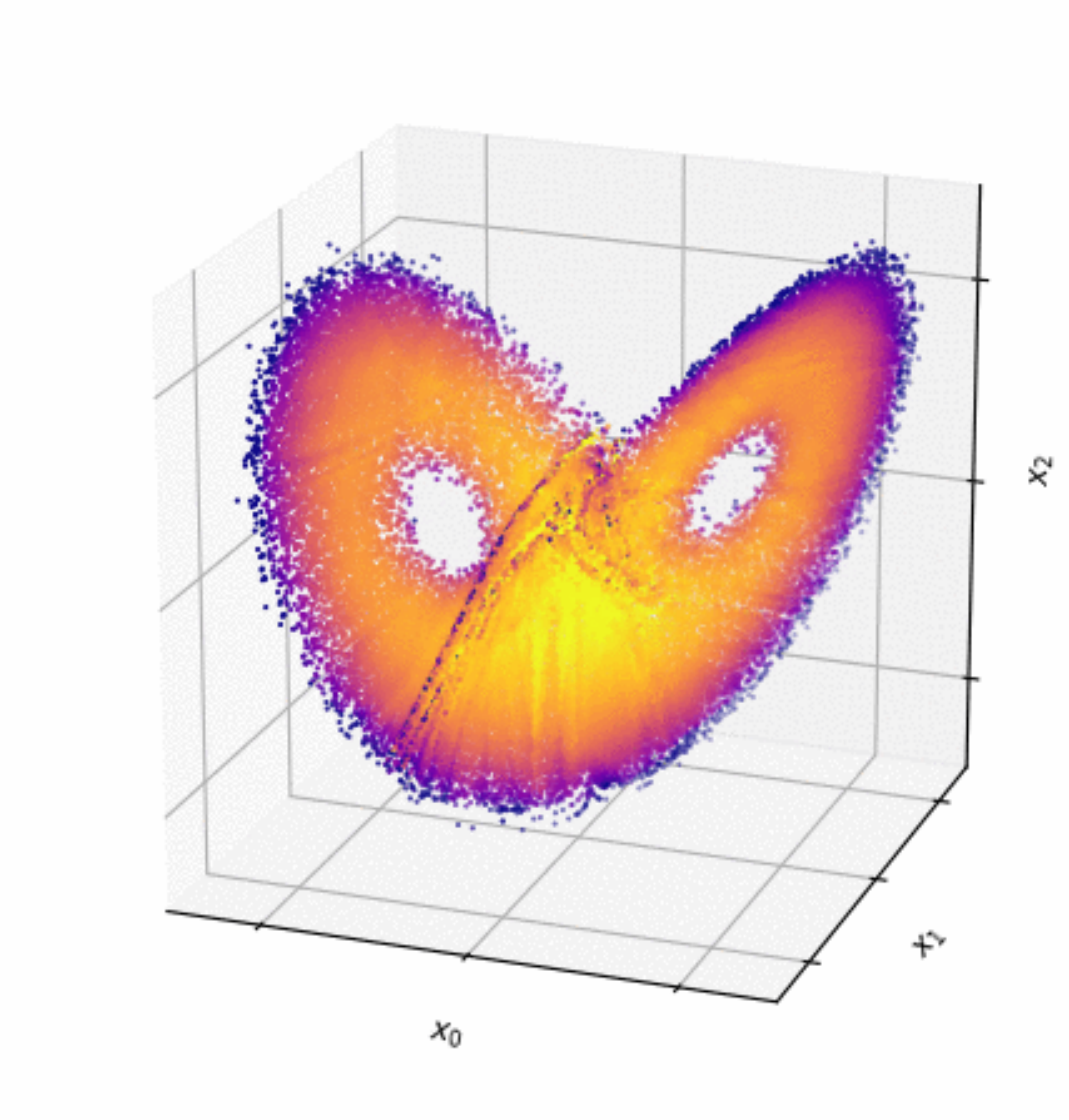


$$\frac{\mathrm{d}x_0}{\mathrm{d}t} = \sigma(x_1 - x_0)\,, \quad \frac{\mathrm{d}x_1}{\mathrm{d}t} = x_0(\rho - x_2) - x_1\,, \quad \frac{\mathrm{d}x_2}{\mathrm{d}t} = x_0 x_1 - \beta x_2\,.$$

Lorenz attractor

[E. Lorenz 1963]





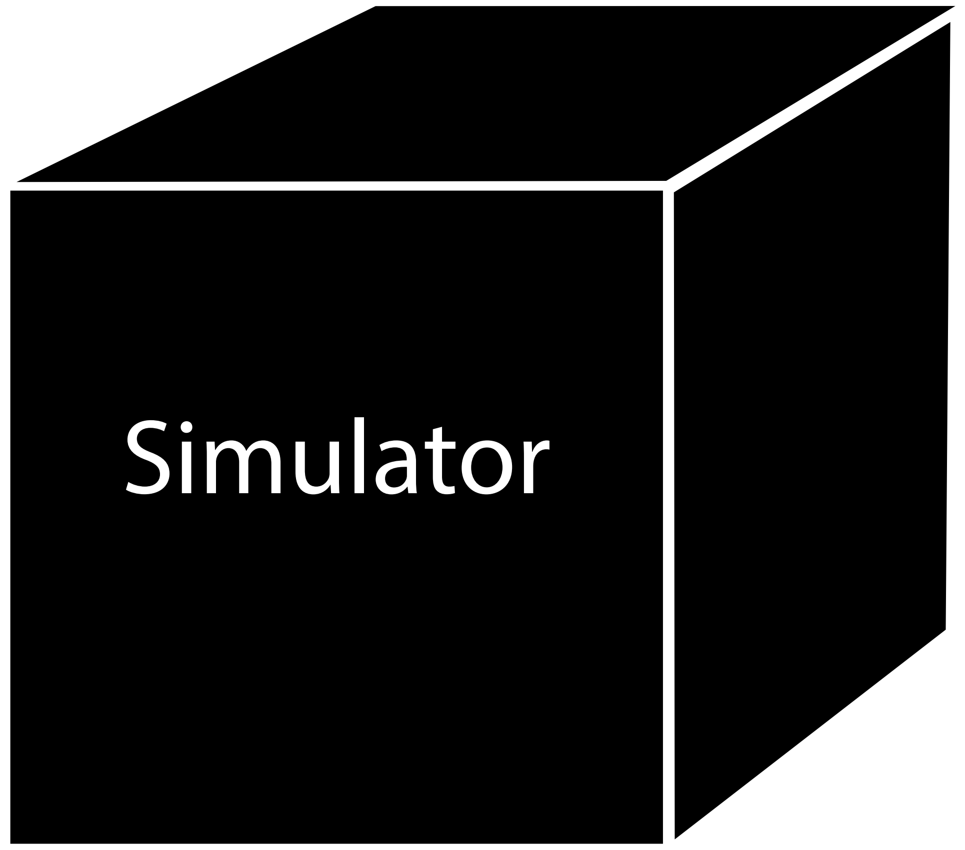
$$\frac{\mathrm{d}x_0}{\mathrm{d}t} = \sigma(x_1 - x_0)\,, \quad \frac{\mathrm{d}x_1}{\mathrm{d}t} = x_0(\rho - x_2) - x_1\,, \quad \frac{\mathrm{d}x_2}{\mathrm{d}t} = x_0 x_1 - \beta x_2\,.$$

Particle physics as a likelihood-free inference problem

$\mathcal{O}(10)$ parameters θ

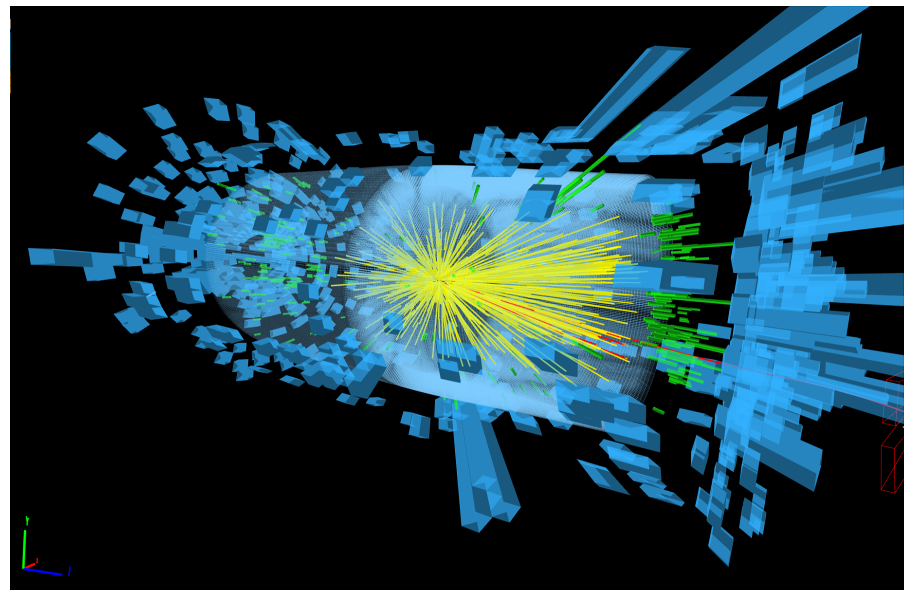
$$\begin{aligned} S = \int \mathrm{d}^4x \Bigg[& \mathcal{L}_{\mathrm{SM}} + \frac{f_{\phi,2}}{\Lambda^2}\,\frac{1}{2}\,\partial_\mu(\phi^\dagger\phi)\,\partial^\mu(\phi^\dagger\phi) + \frac{f_{\phi,3}}{\Lambda^2}\,\frac{1}{3}(\phi^\dagger\phi)^3 \\ & + \frac{f_{GG}}{\Lambda^2}\,(\phi^\dagger\phi)\,G^a_{\mu\nu}\,G^{\mu\nu\,a} - \frac{f_{BB}}{\Lambda^2}\,\frac{g'^2}{4}\,(\phi^\dagger\phi)\,B_{\mu\nu}\,B^{\mu\nu} - \frac{f_{WW}}{\Lambda^2}\,\frac{g^2}{4}\,(\phi^\dagger\phi)\,W^a_{\mu\nu}\,W^{\mu\nu\,a} \\ & + \frac{f_B}{\Lambda^2}\,\frac{\mathrm{i}g'}{2}\,(D^\mu\phi)^\dagger\,D^\nu\phi\,B_{\mu\nu} + \frac{f_W}{\Lambda^2}\,\frac{\mathrm{i}g}{2}\,(D^\mu\phi)^\dagger\,\sigma^a\,D^\nu\phi\,W^a_{\mu\nu} \\ & + \frac{f_\ell}{\Lambda^2}\,(\phi^\dagger\phi)\,\overline{L}_L\phi\,\ell_R + \frac{f_u}{\Lambda^2}\,(\phi^\dagger\phi)\,\overline{Q}_L\tilde{\phi}\,u_R + \frac{f_d}{\Lambda^2}\,(\phi^\dagger\phi)\,\overline{Q}_L\phi\,d_R \\ & + \frac{f_{G\widetilde{G}}}{\Lambda^2}\,(\phi^\dagger\phi)\,G^a_{\mu\nu}\,\widetilde{G}^{\mu\nu\,a} - \frac{f_{B\widetilde{B}}}{\Lambda^2}\,\frac{g'^2}{4}(\phi^\dagger\phi)\,B_{\mu\nu}\,\widetilde{B}^{\mu\nu} - \frac{f_{W\widetilde{W}}}{\Lambda^2}\,\frac{g^2}{4}(\phi^\dagger\phi)\,W^a_{\mu\nu}\,\widetilde{W}^{\mu\nu\,a} \Bigg] \end{aligned}$$

$\longrightarrow$



$\longrightarrow$

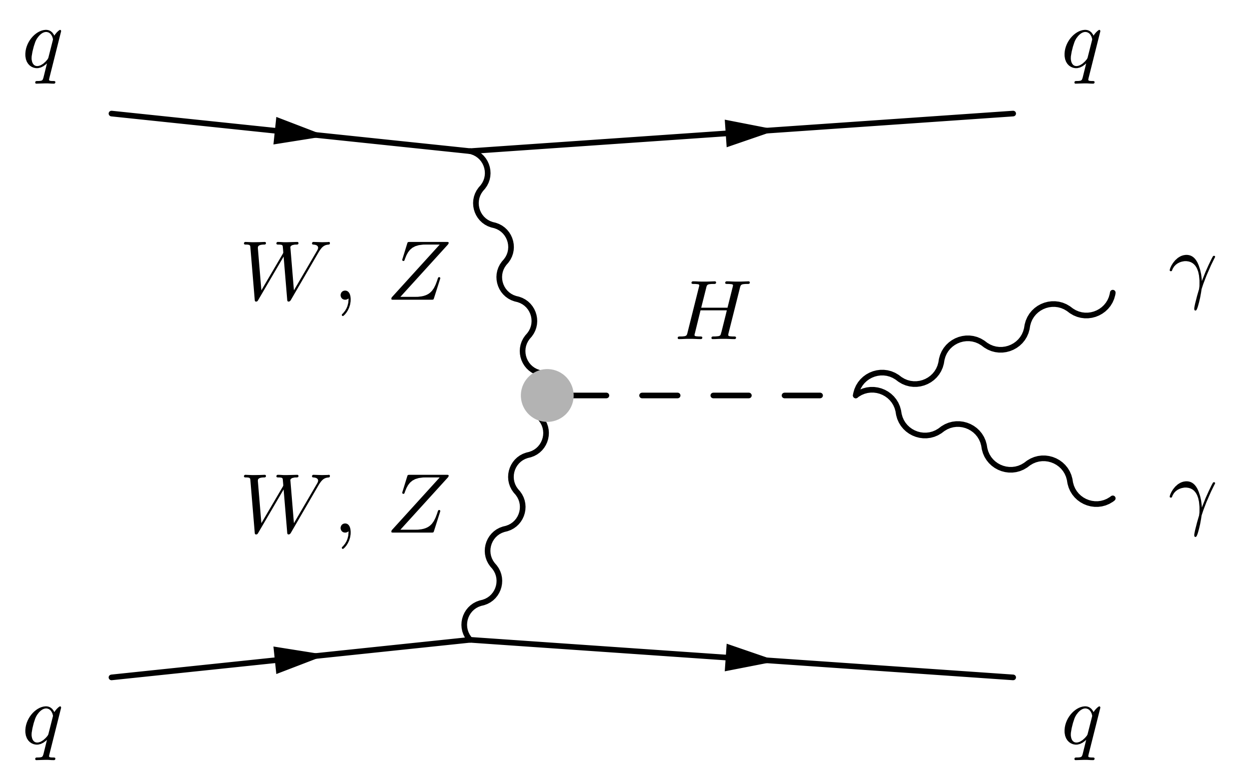
$\mathcal{O}(10\ldots1000)$ observables x



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

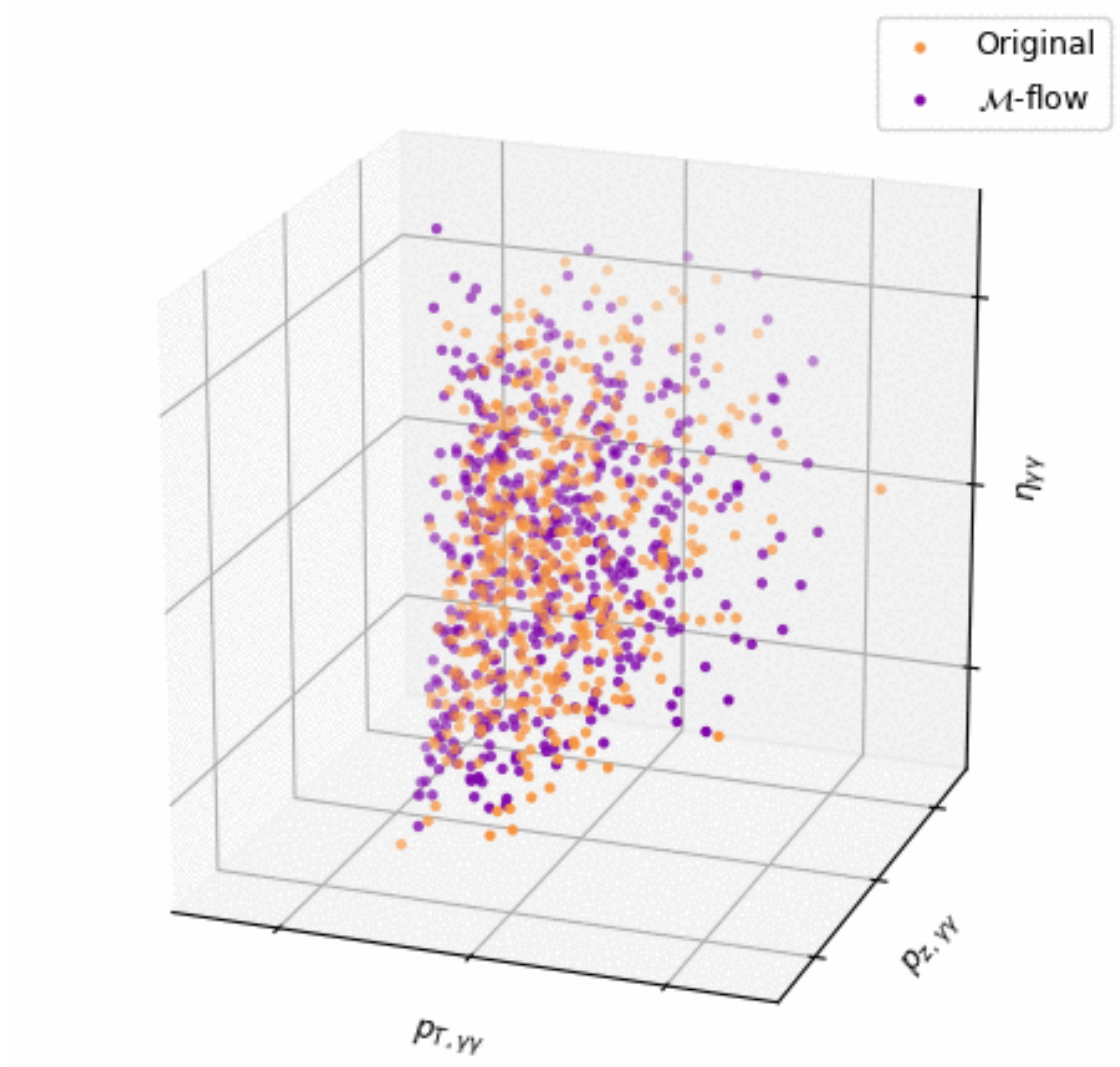


particle masses
("on-shell condition")

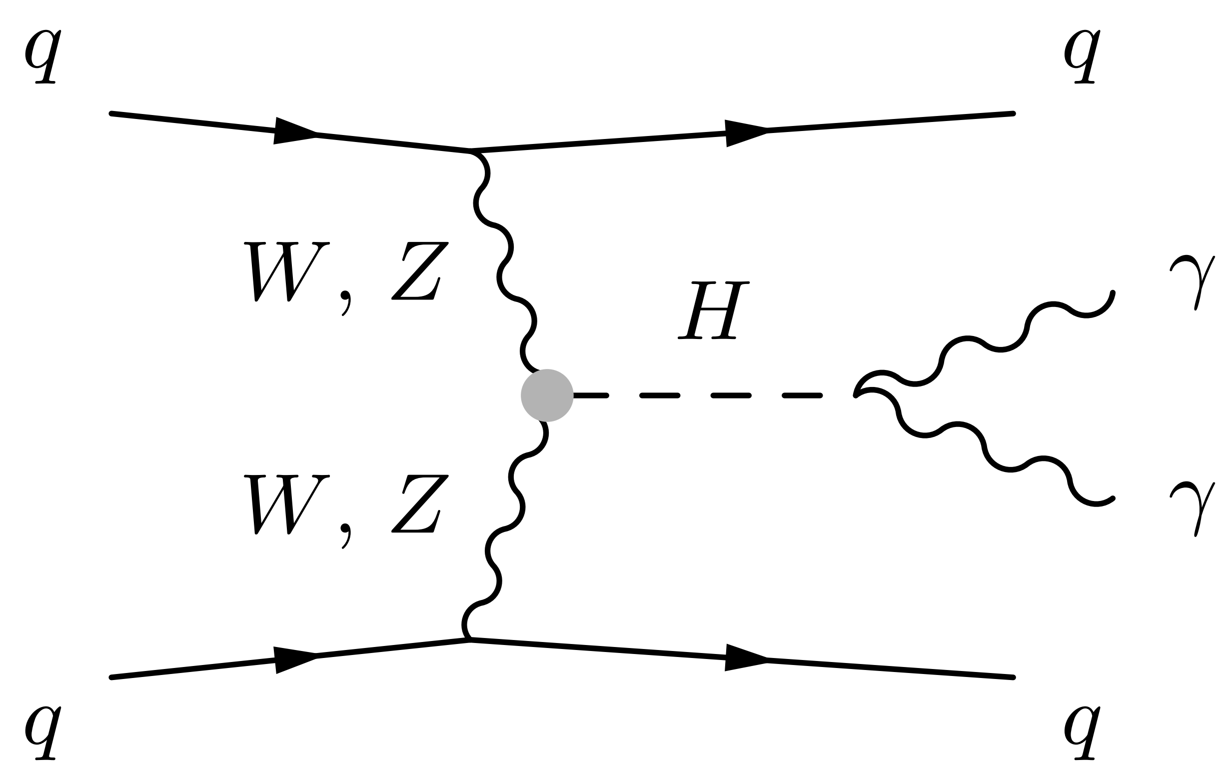
redundant
observables

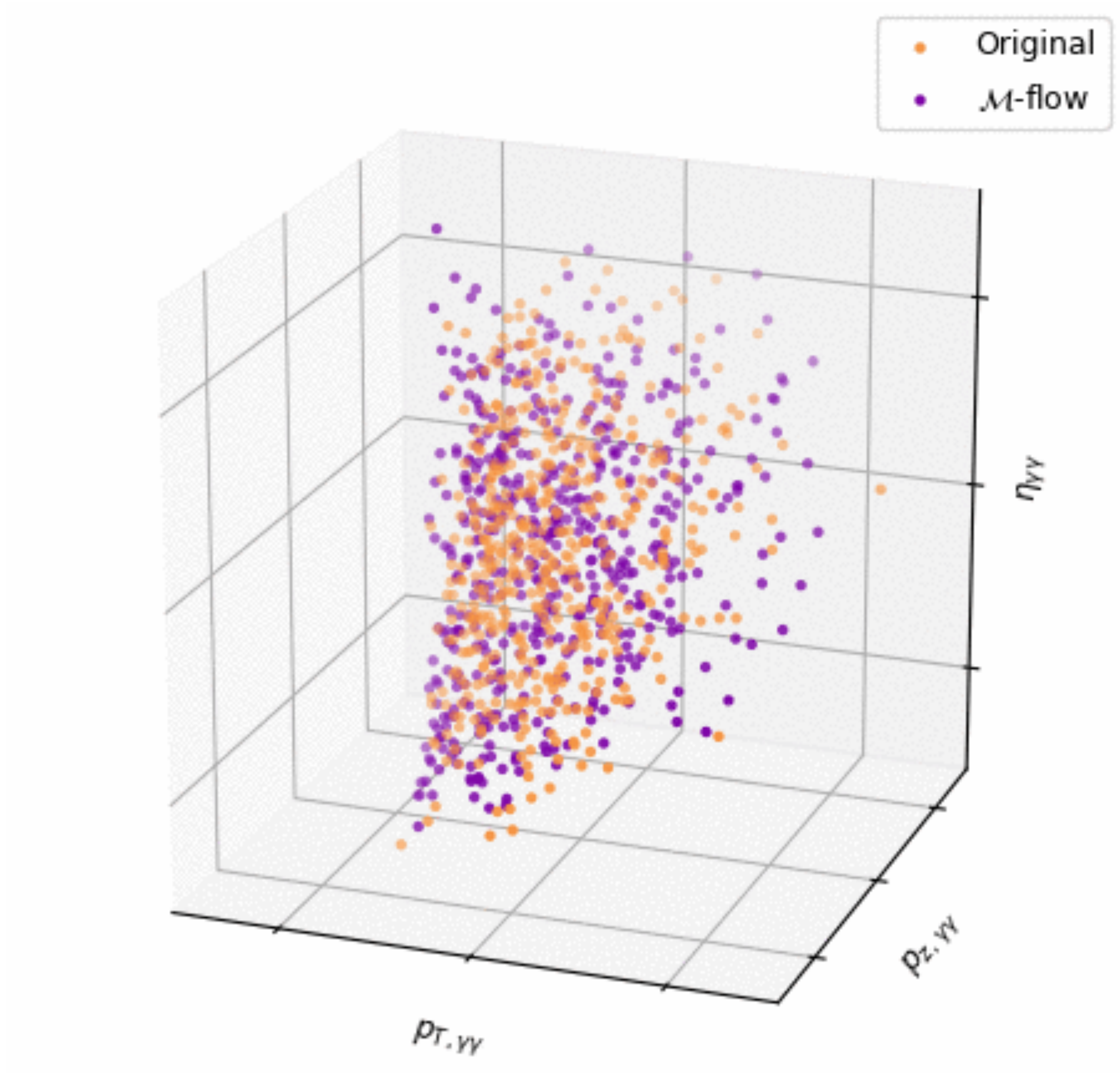
energy-momentum
conservation

14-dimensional manifold
embedded in 40-dimensional
data space



Particle physics: structure





particle masses
("on-shell condition")

redundant
observables

energy-momentum
conservation

14-dimensional manifold
embedded in 40-dimensional
data space

Particle physics: results

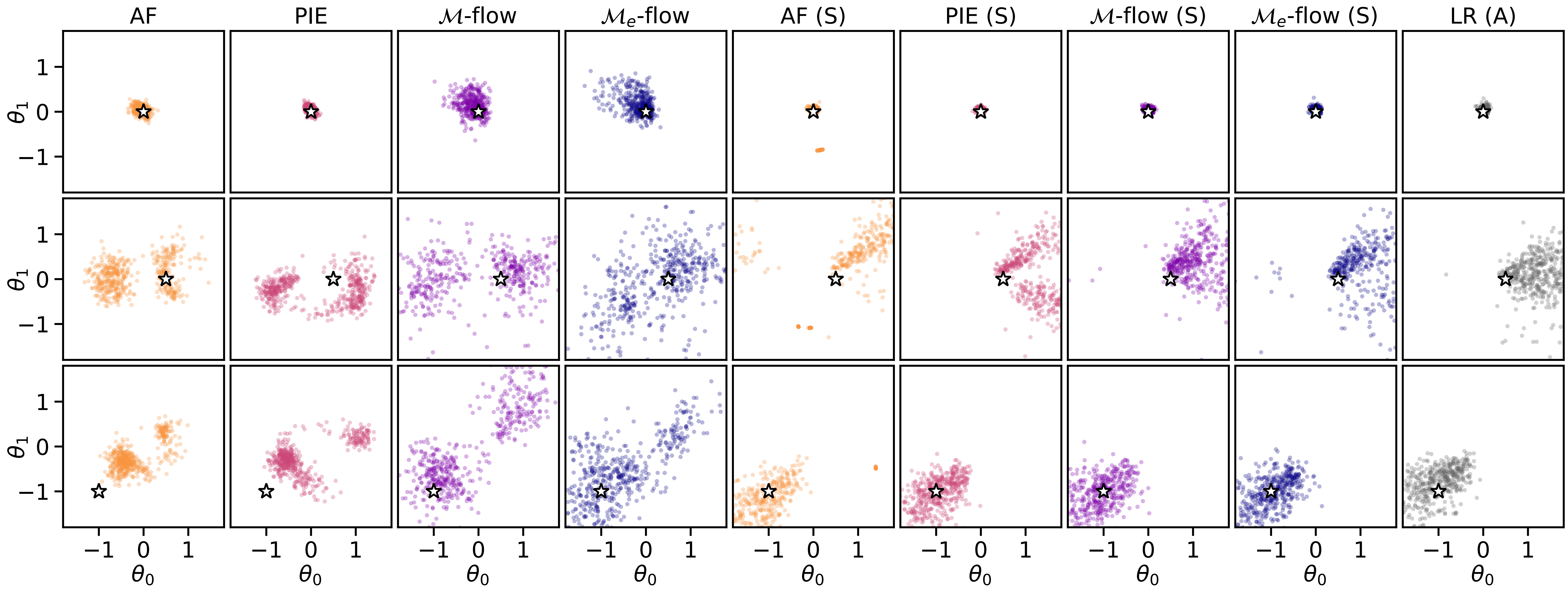
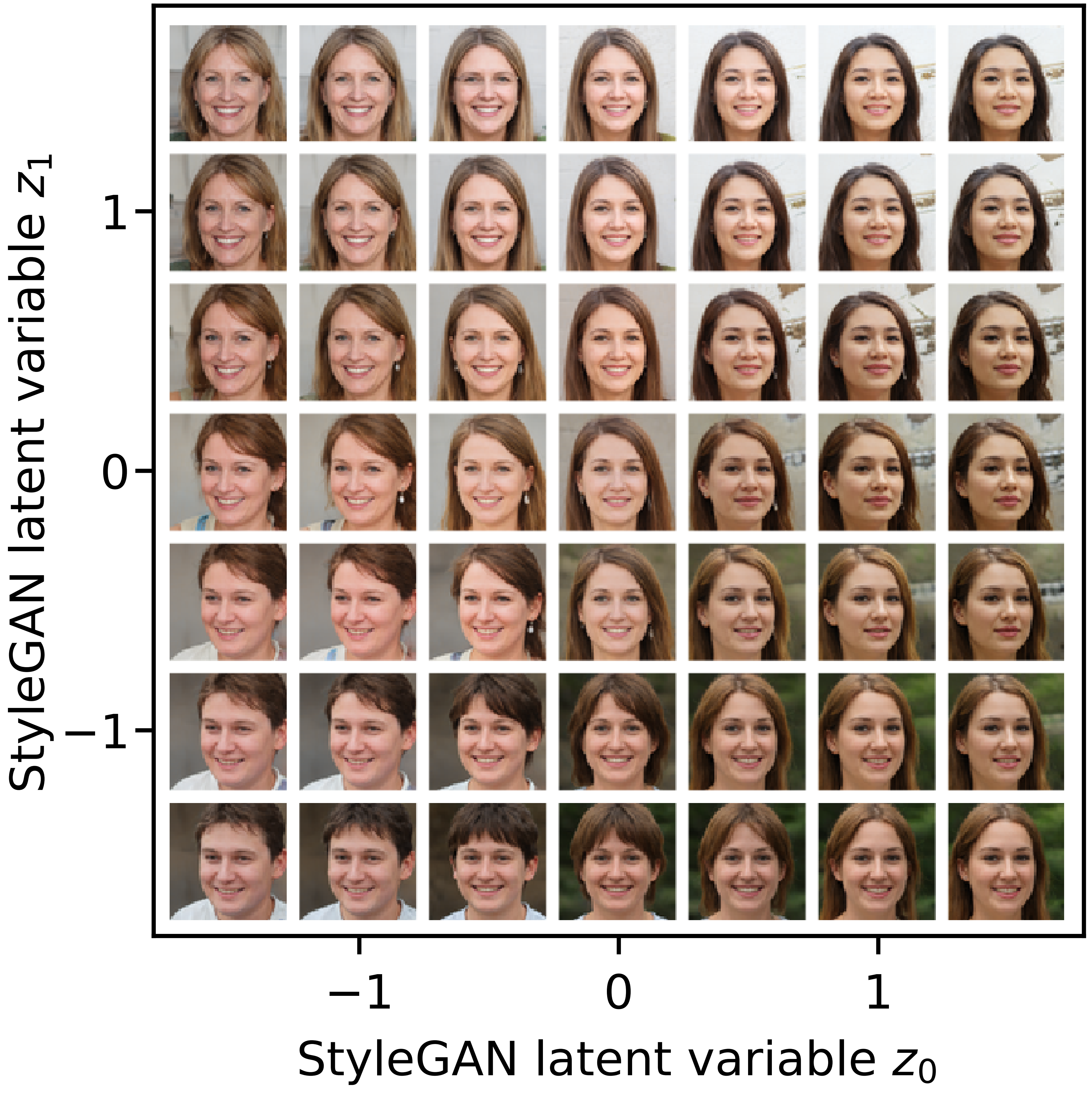


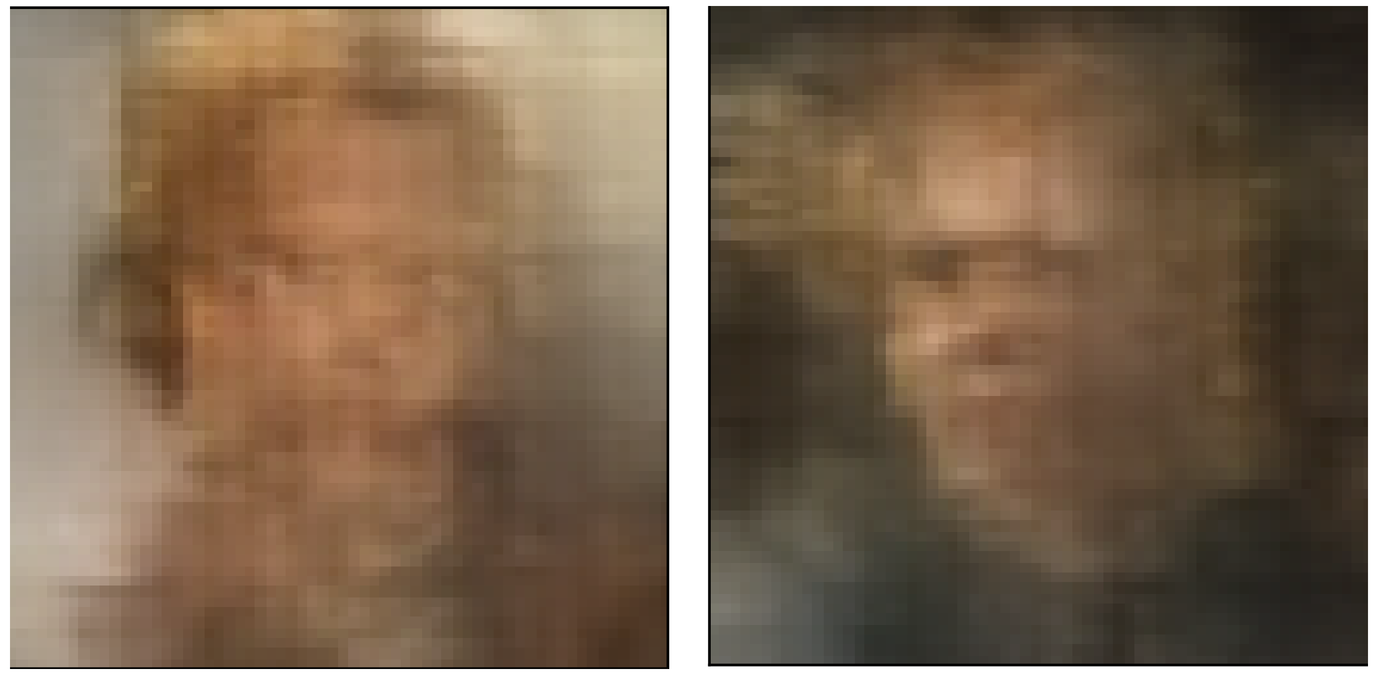
Image manifolds

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

$n = 2$





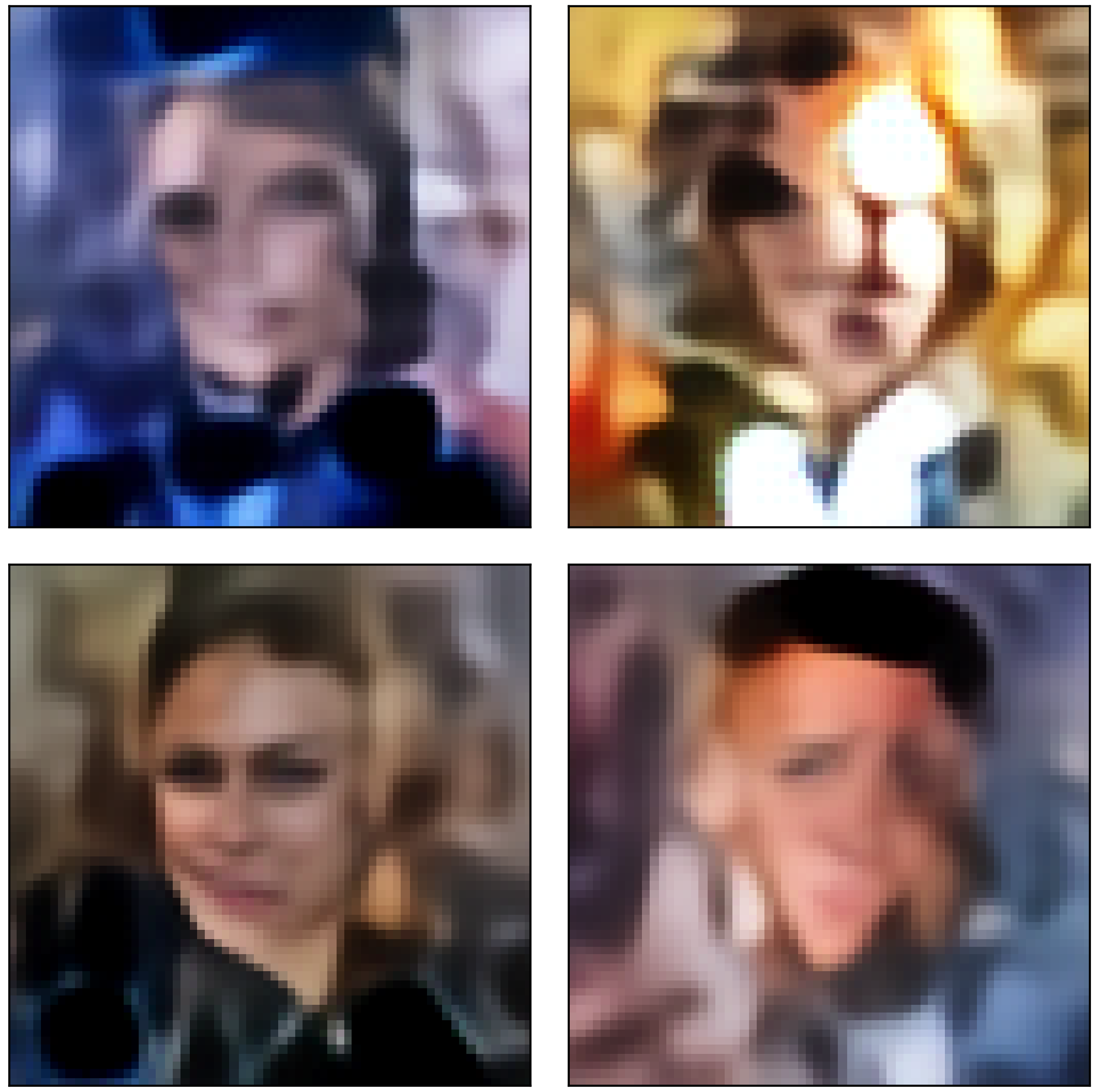


Image manifold results

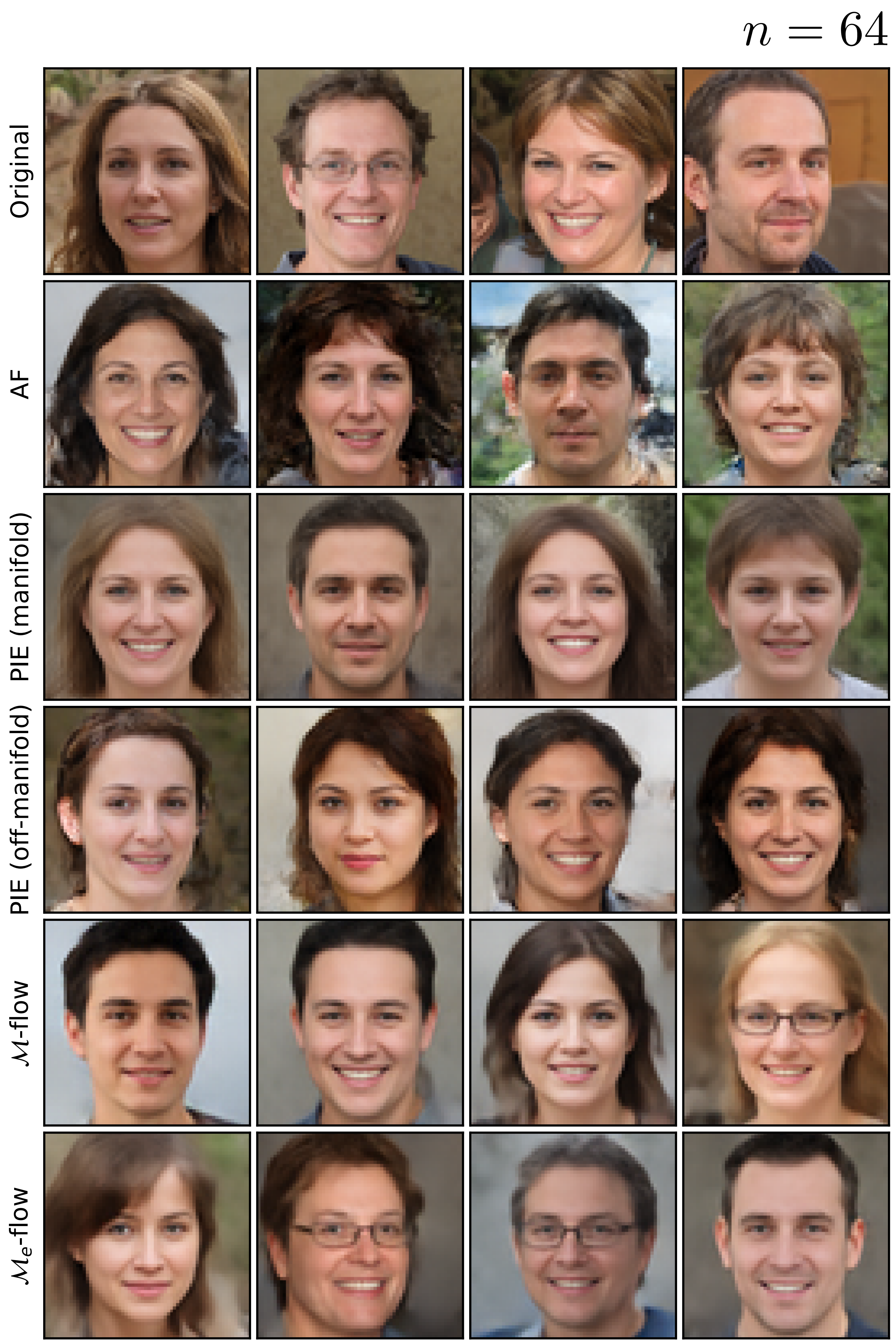
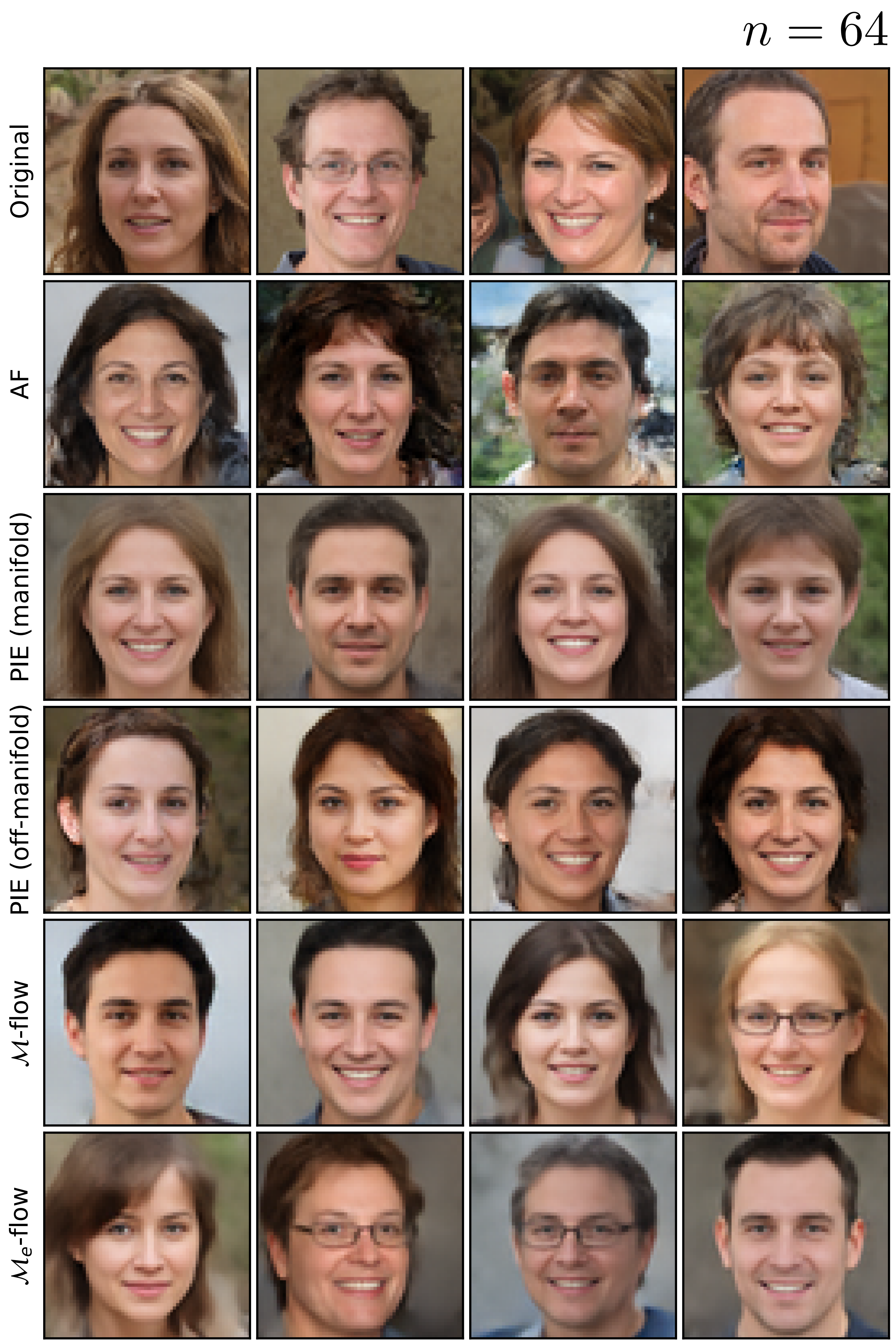


Image manifold results



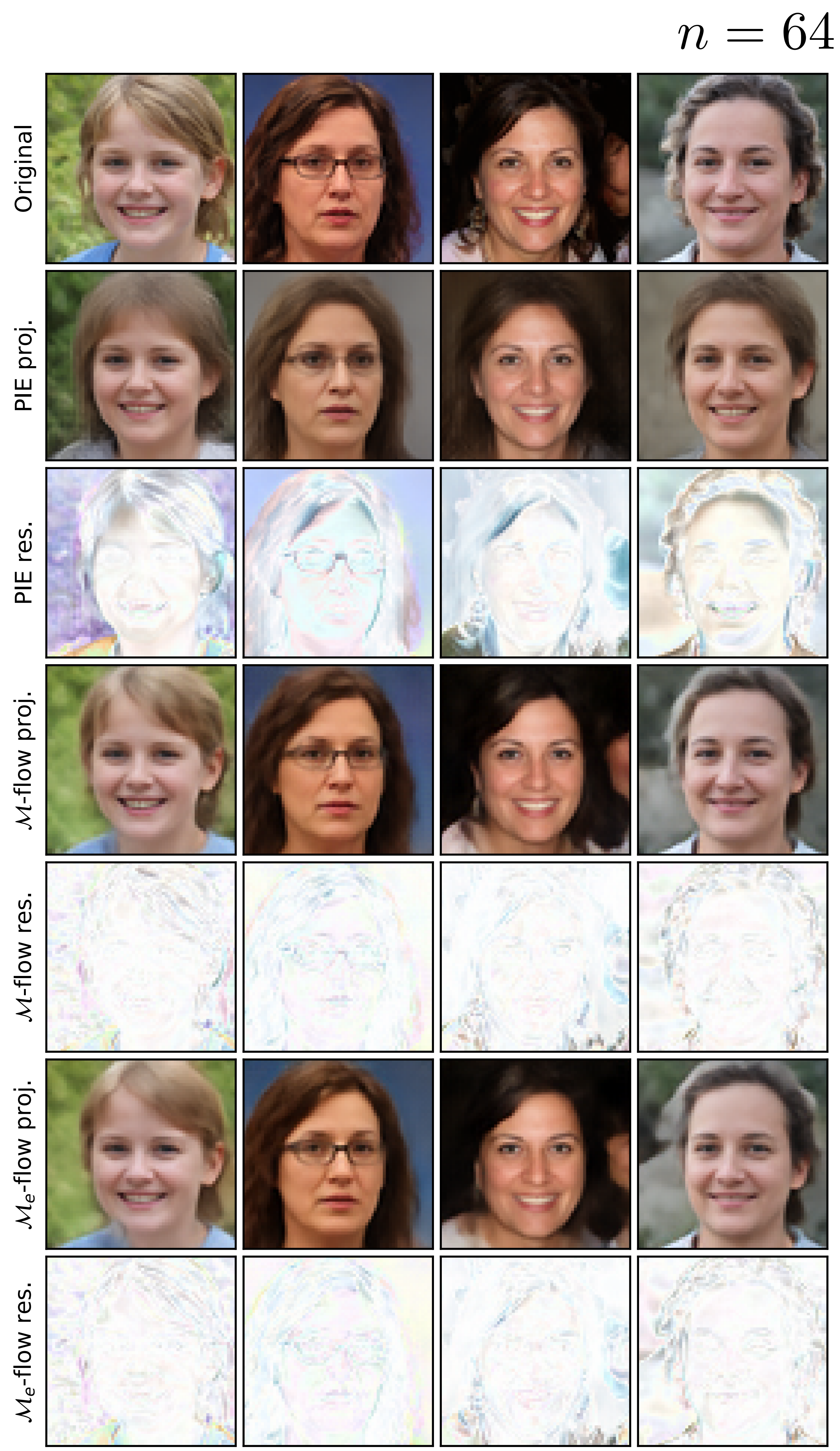
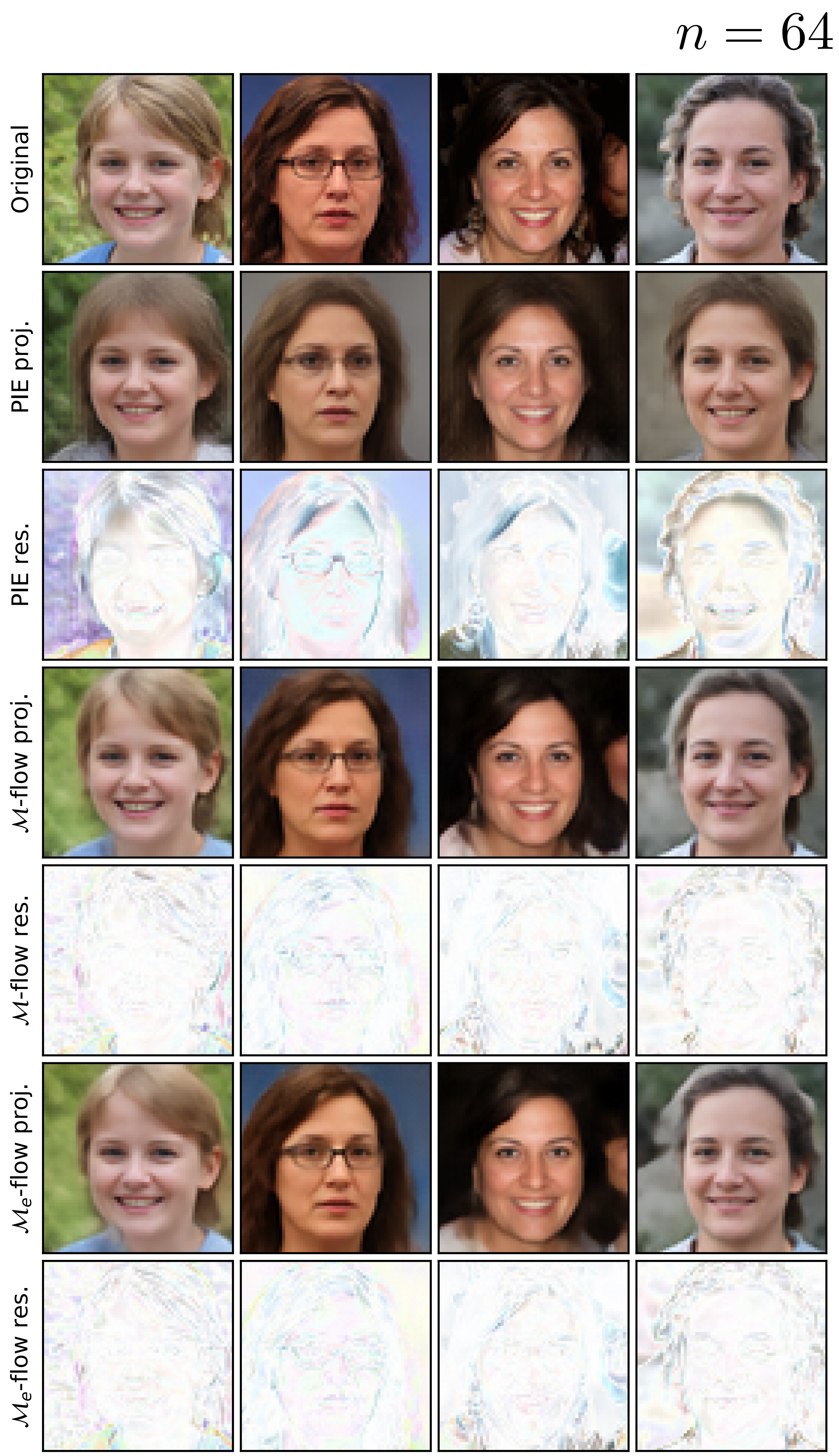


Image manifold results



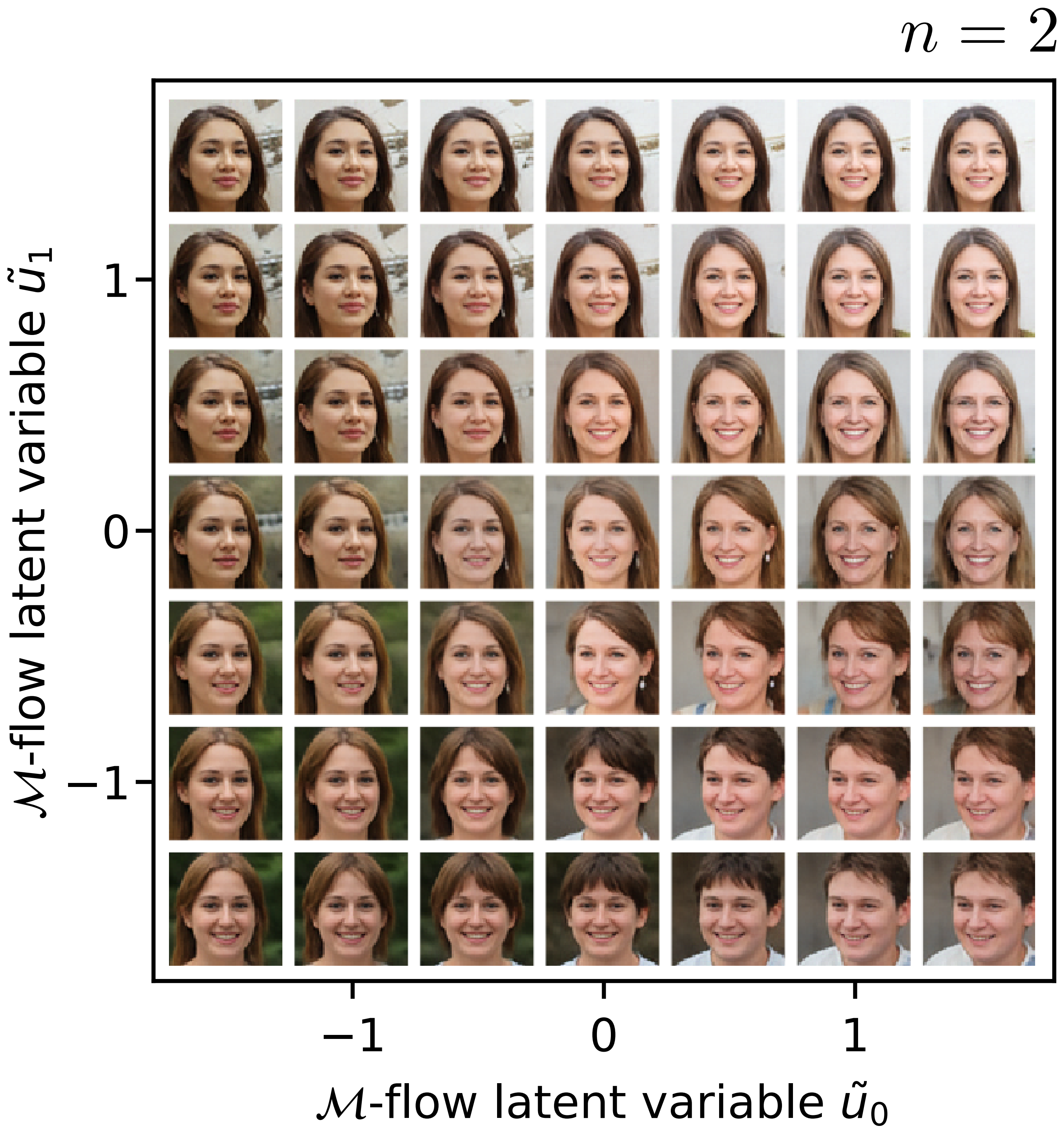
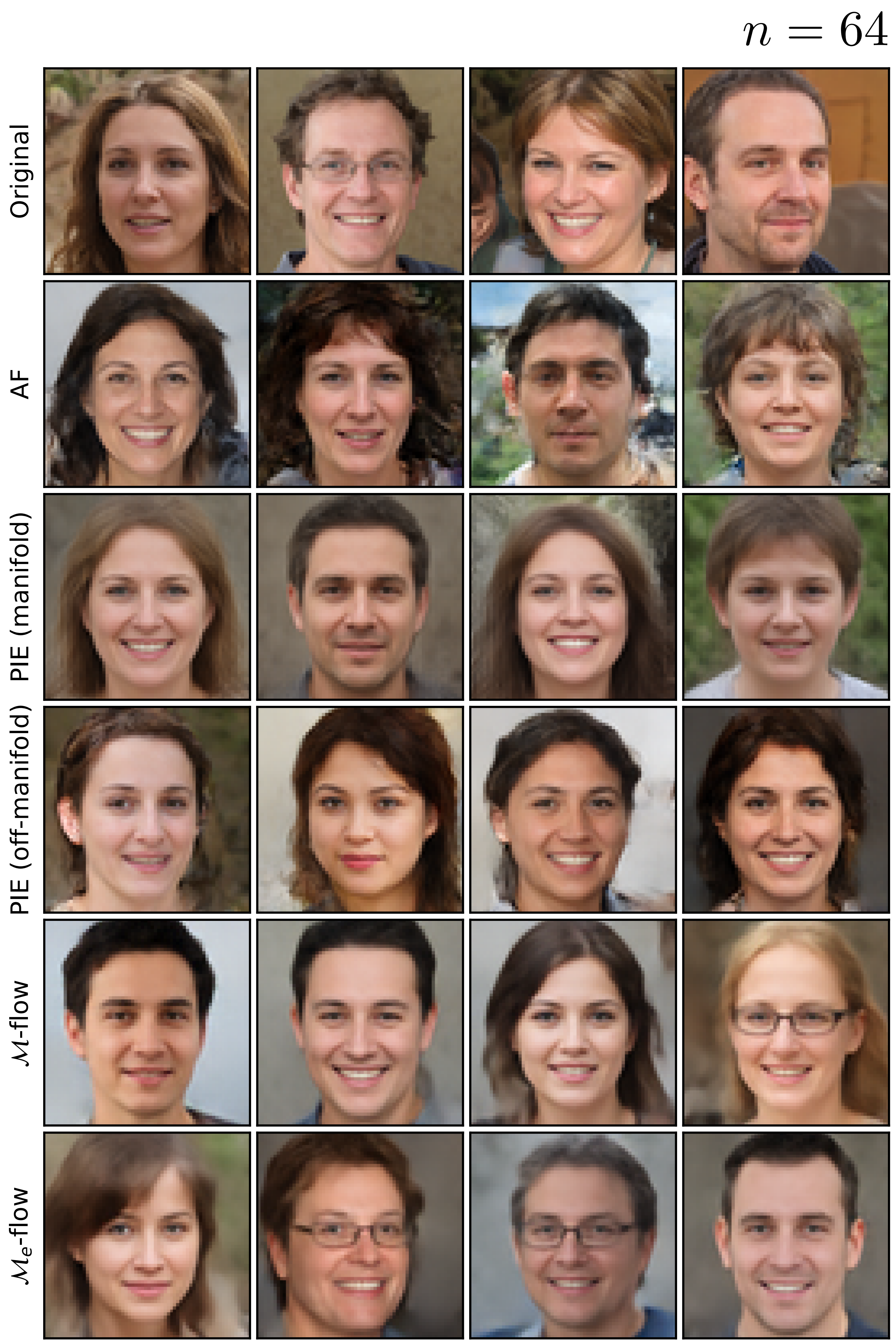
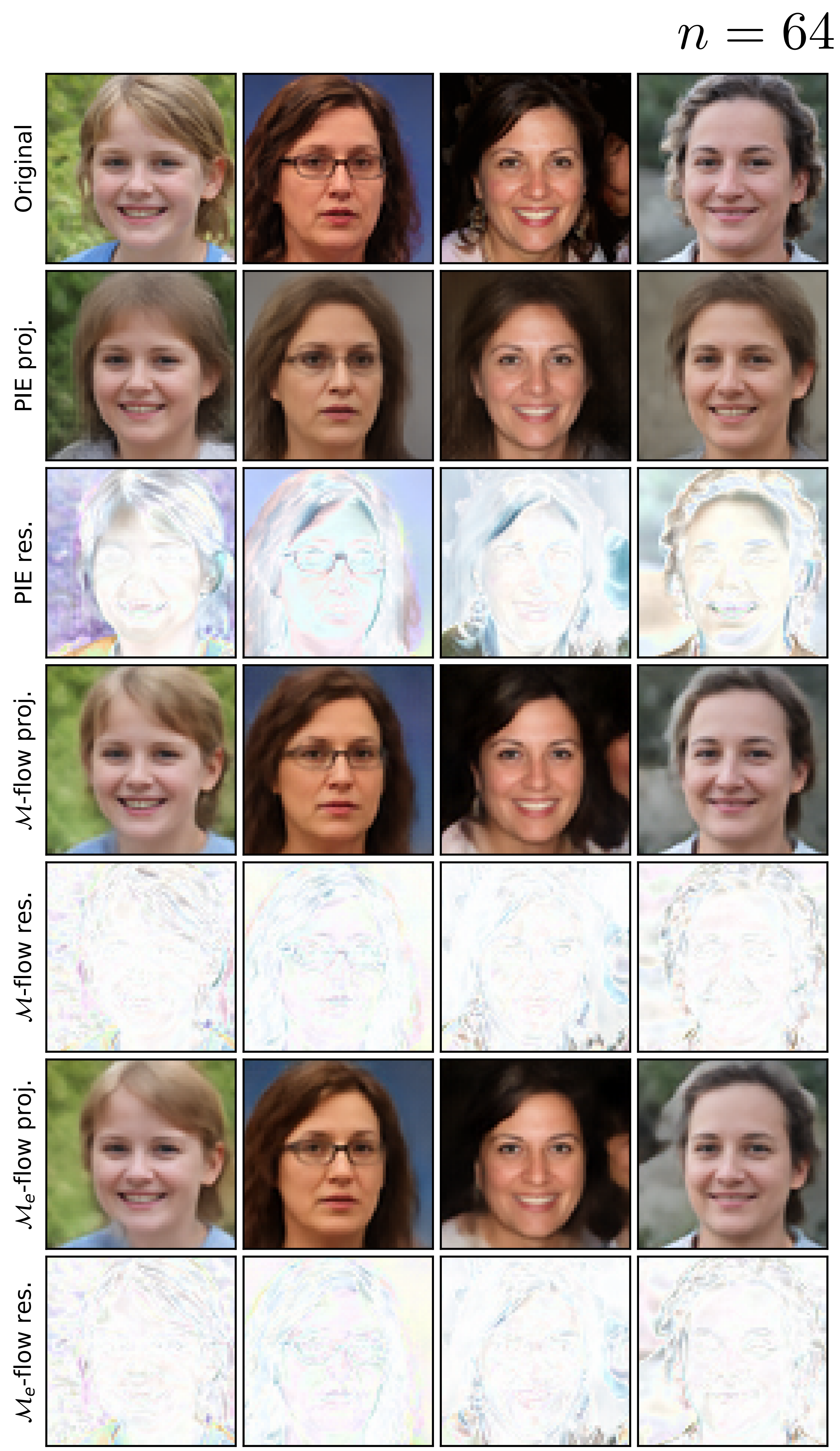


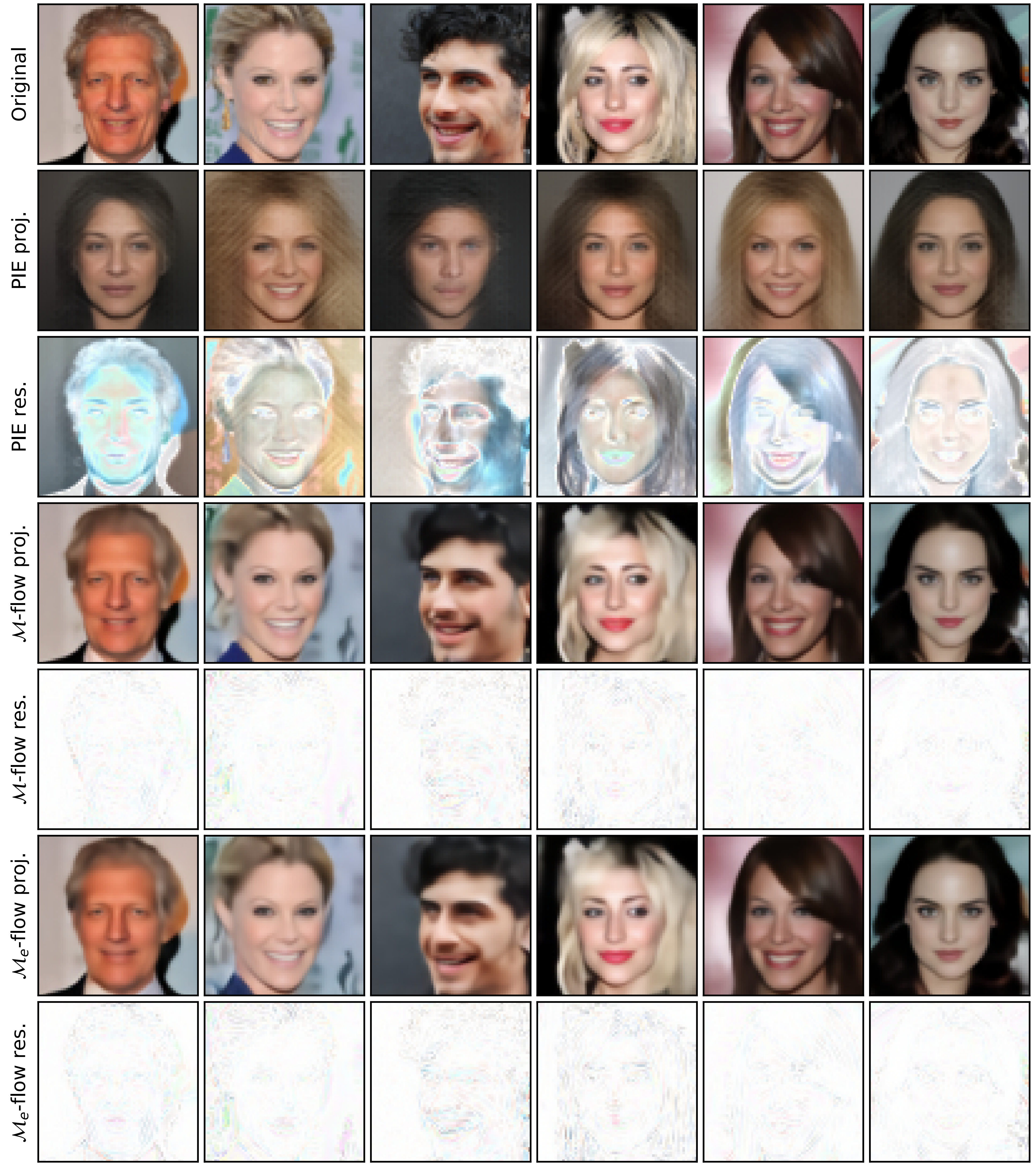
Image manifold results





Real-world images: CelebA

Real-world images: CelebA



Metrics

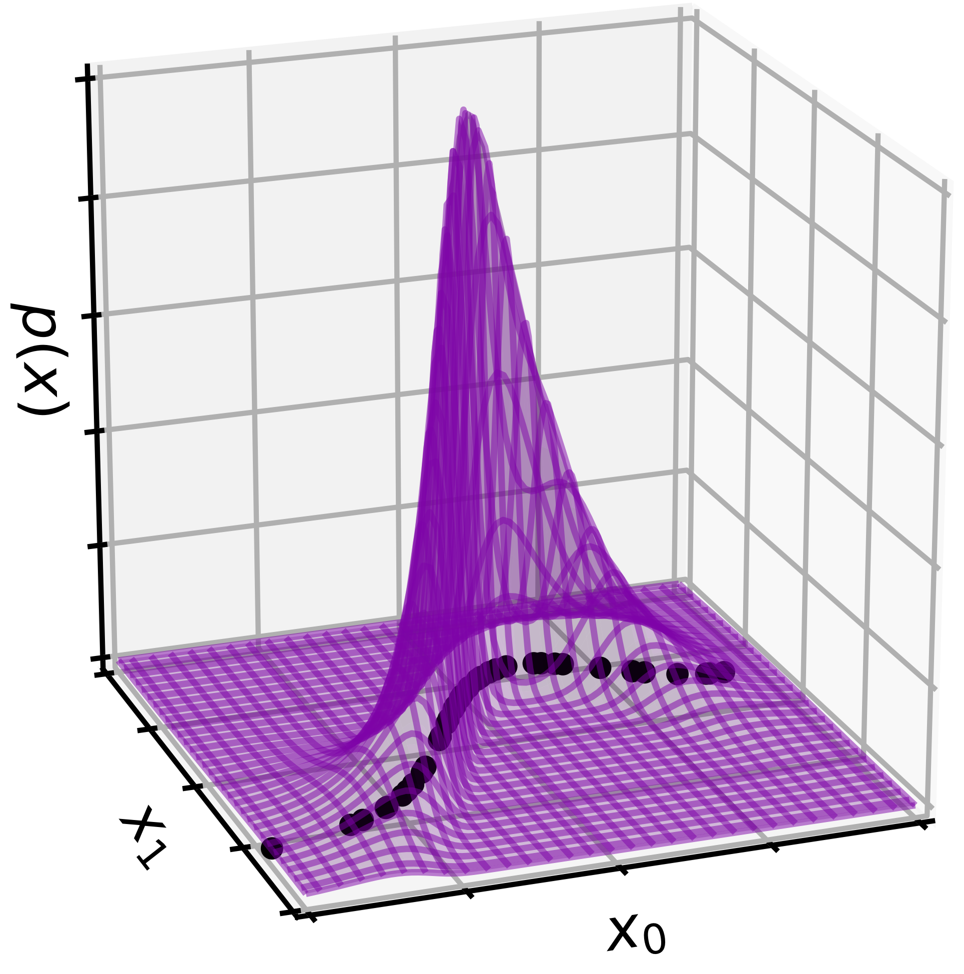
Model	Polynomial surface			Particle physics			Images			
	Distance	RE	MMD	Closure	RE	$\log p(\theta^*)$	$n = 2$ FID	$n = 64$ FID	$n = 64$ $\log p(\theta^*)$	CelebA FID
AF	0.005	–	0.071	**0.0019**	–	-3.94	58.3	24.0	0.17	**33.6**
PIE	0.035	1.278	0.131	0.0023	2.054	-4.68	139.5	32.2	-6.40	75.7
$\mathcal{M}$-flow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

Open questions

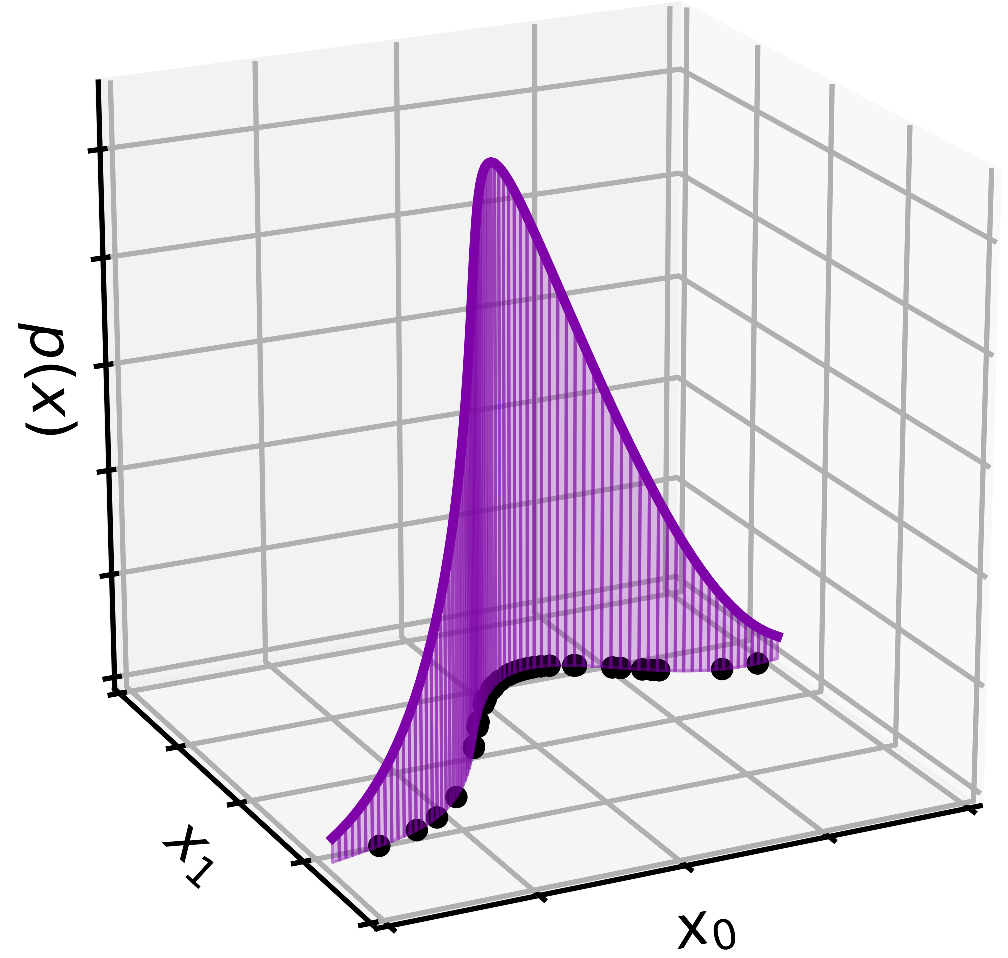
How do you learn n ?

What are good architectures for image data?

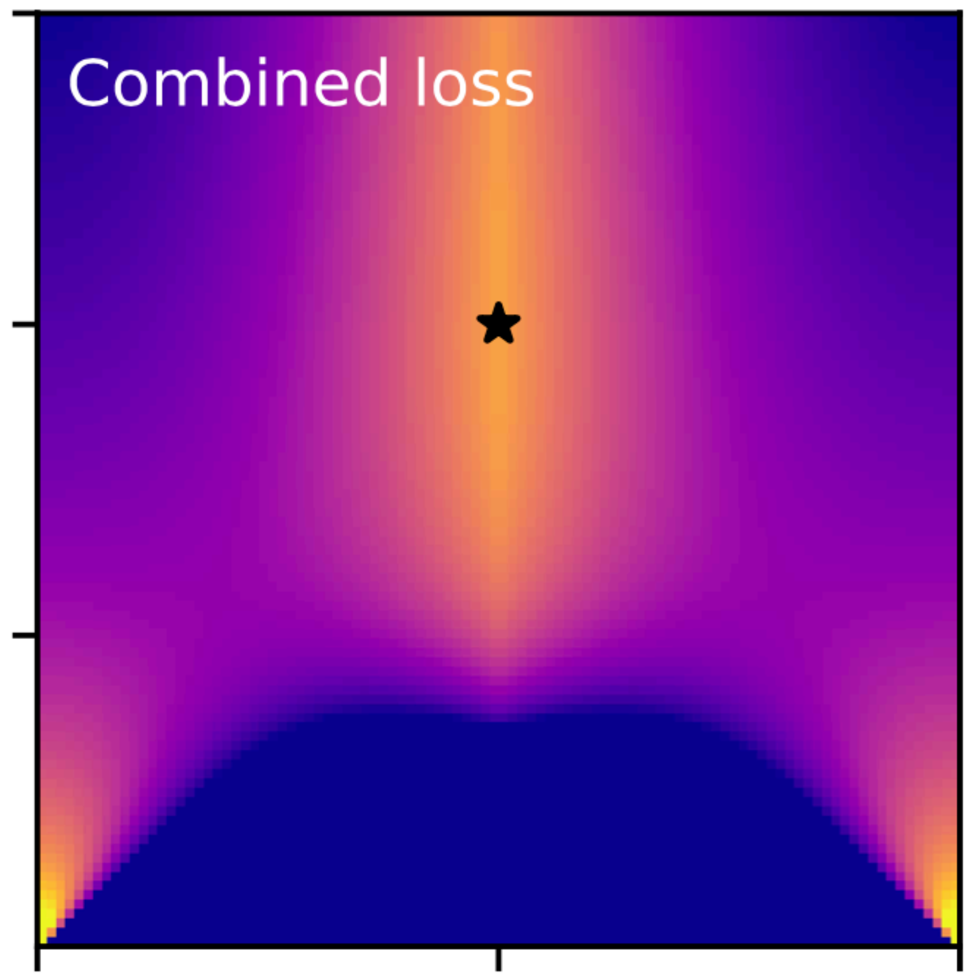
Are there transformations f for which the likelihood can be computed faster?



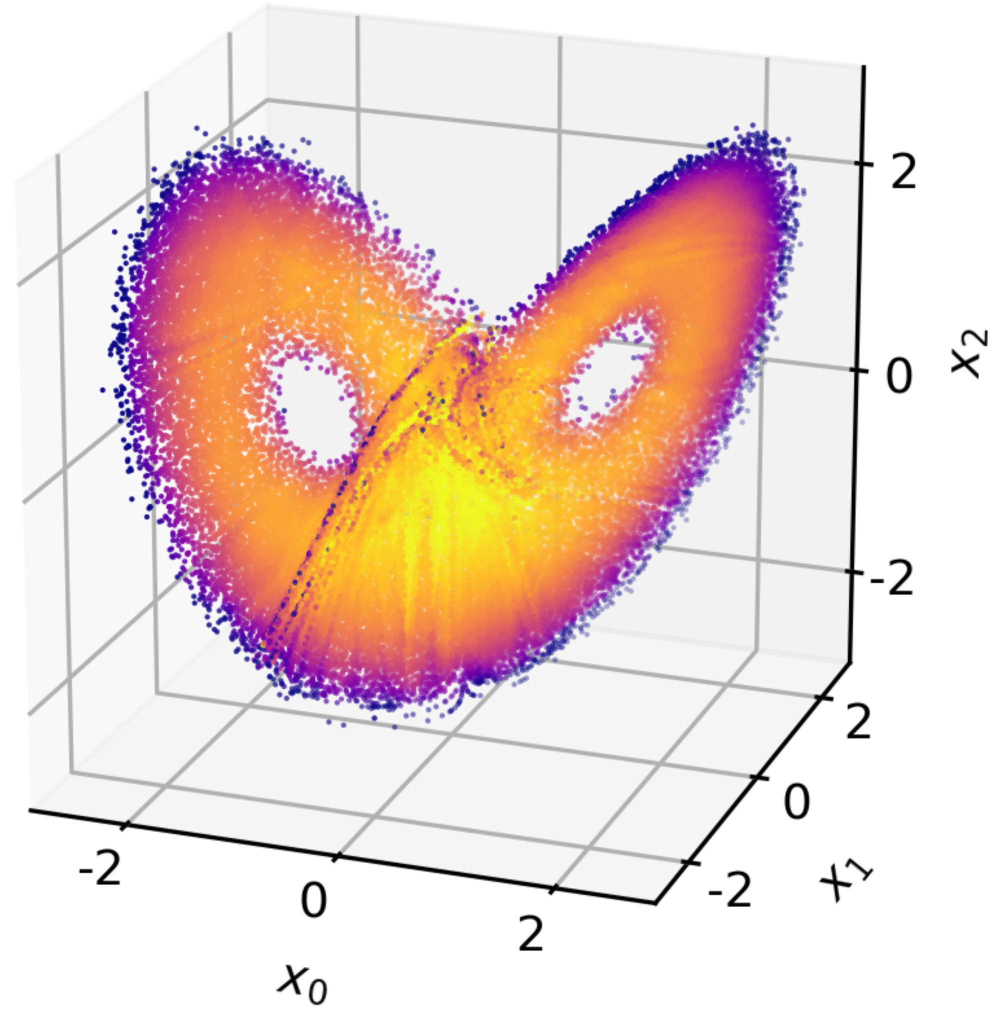
Standard ambient flows cannot represent lower-dim. data manifolds



$\mathcal{M}$-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

[JB, Kyle Cranmer 2003.13913]