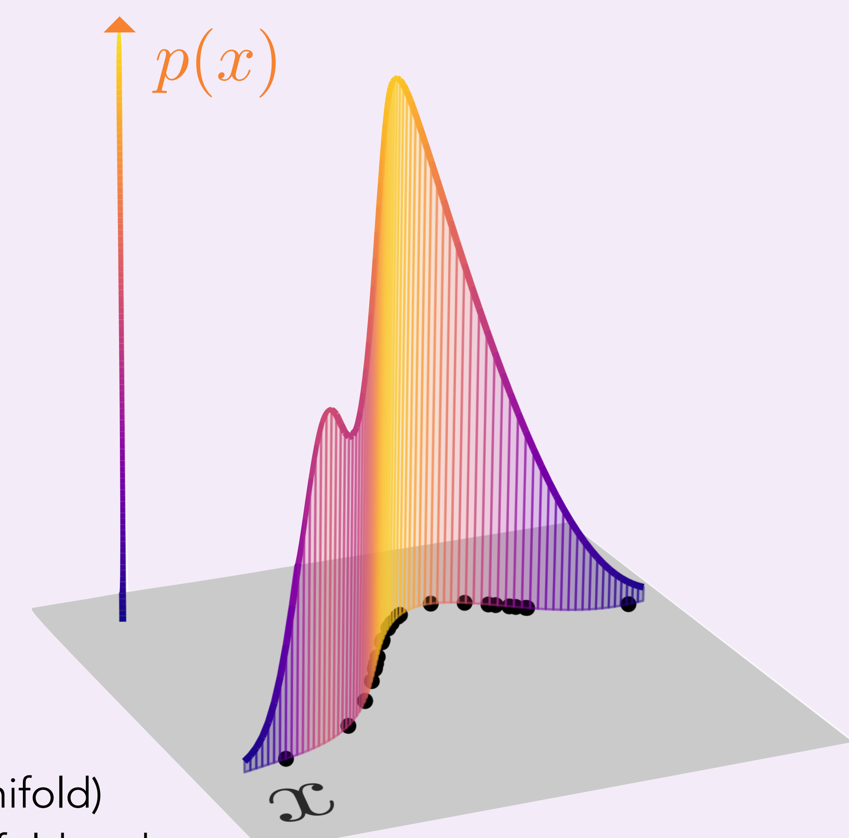
Flows for simultaneous manifold learning and density estimation

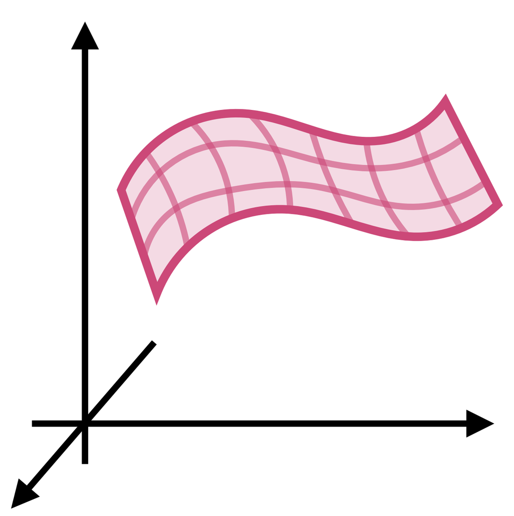
Johann Brehmer and Kyle Cranmer, New York University

Introducing $\mathcal{M}$-flows

- We present a new type of normalizing flow in which the probability density is consistently restricted to a lower-dimensional Riemannian manifold embedded in the data space
- Manifold, chart, and density are all learned from data
- The density is tractable, making $\mathcal{M}$-flows perfectly suited for inference tasks
- $\mathcal{M}$-flows evaluate any data point (which may be off the manifold) by first projecting it to the manifold and returning
  - the log likelihood on the manifold and
  - the reconstruction error from this projection



Motivation



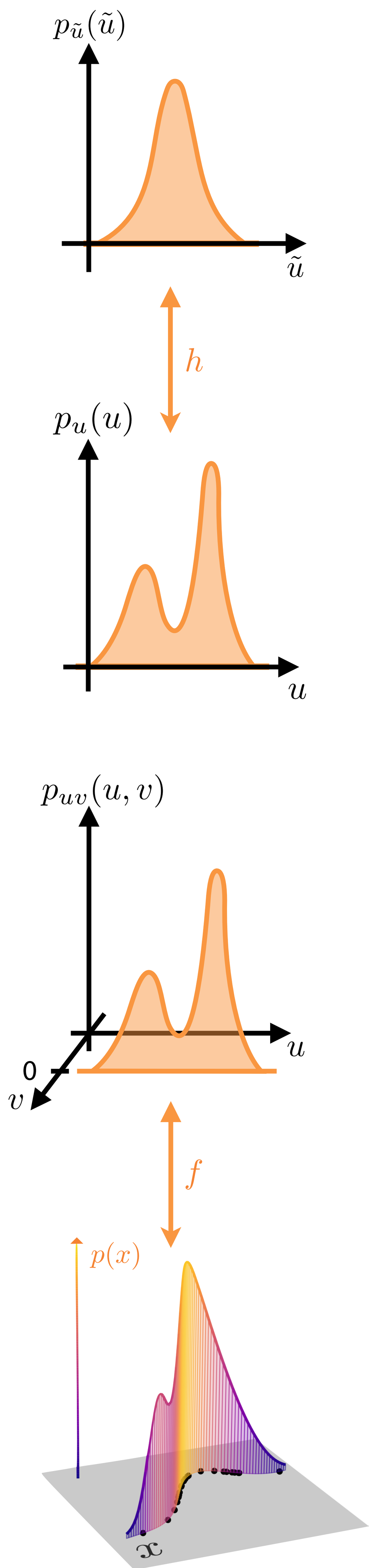
- Many data sets do not fill out the full ambient data space, but are restricted to a lower-dimensional data manifold embedded in the ambient space
- In some scientific problems, the manifold structure is explicit and its dimensionality known, in other cases there is empirical evidence for an (approximate) data manifold
- GANs and VAEs model data manifolds, but do not have a tractable density, which makes them unsuitable for some inference tasks
- Standard Euclidean normalizing flows [3] cannot represent a data manifold exactly and will always learn a smeared-out approximate density with support off the data manifold
- Previously, flows have been generalized to manifolds [4], but this approach has so far been limited to the case where the chart for the manifold is prescribed

Model	Manifold	Chart	Tractable density	Consistently restricted to $\mathcal{M}$
Standard ambient flow	no manifold	×	✓	×
Flow on prescribed manifold	prescribed	✓	✓	✓
GAN	learned	×	×	✓
VAE	learned	×	only ELBO	(×)
PIE [5]	learned	✓	✓	(×)
$\mathcal{M}$-flow	learned	✓	✓	✓

How they work

Generation

1. Sample from lower-dim. base distribution (e.g. Gaussian)
2. Learnable bijection
3. Embedding into higher-dim. space
4. Another learnable bijection
5. Output: data on manifold



Inference

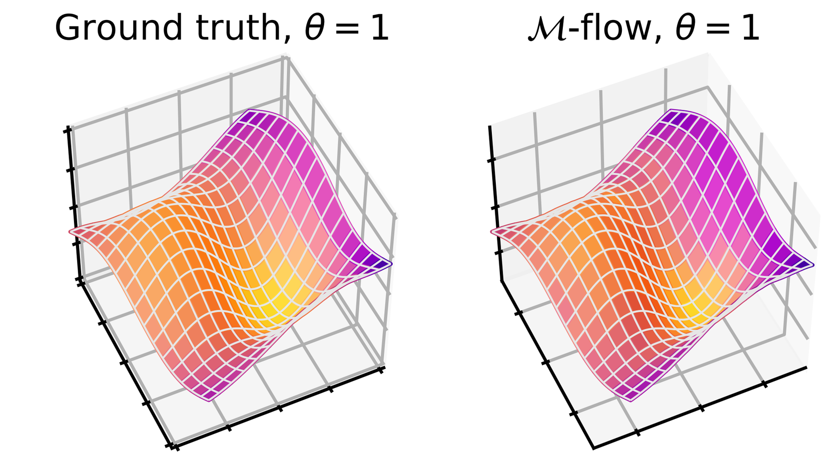
5. Tractable base density, density on manifold follows via change-of-variable formula
4. Inverse transformation
3. Projection to lower-dim. subspace (compute reconstruction error in data space)
2. Inverse transformation
1. Input: data (on or off the manifold)

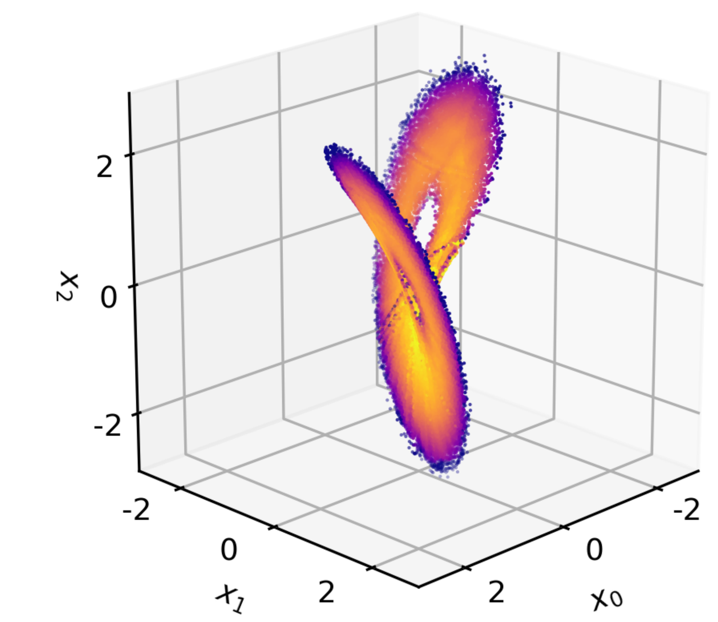
Subtleties in the training

- The $\mathcal{M}$-flow density does not have the properties of a likelihood for the bijection parameters ϕ_f. It is better to think of $p_{\phi_f}(x|\phi_h)$ as a family of likelihoods in ϕ_h, one for each manifold defined by ϕ_f
- While maximum likelihood can be used to learn the density on the manifold, a different strategy must be used to learn the manifold itself
- We introduce the M/D training scheme, which alternates between
  - updating the manifold weights ϕ_f by minimizing the reconstruction error
  - updating the density weights ϕ_h by maximizing the likelihood

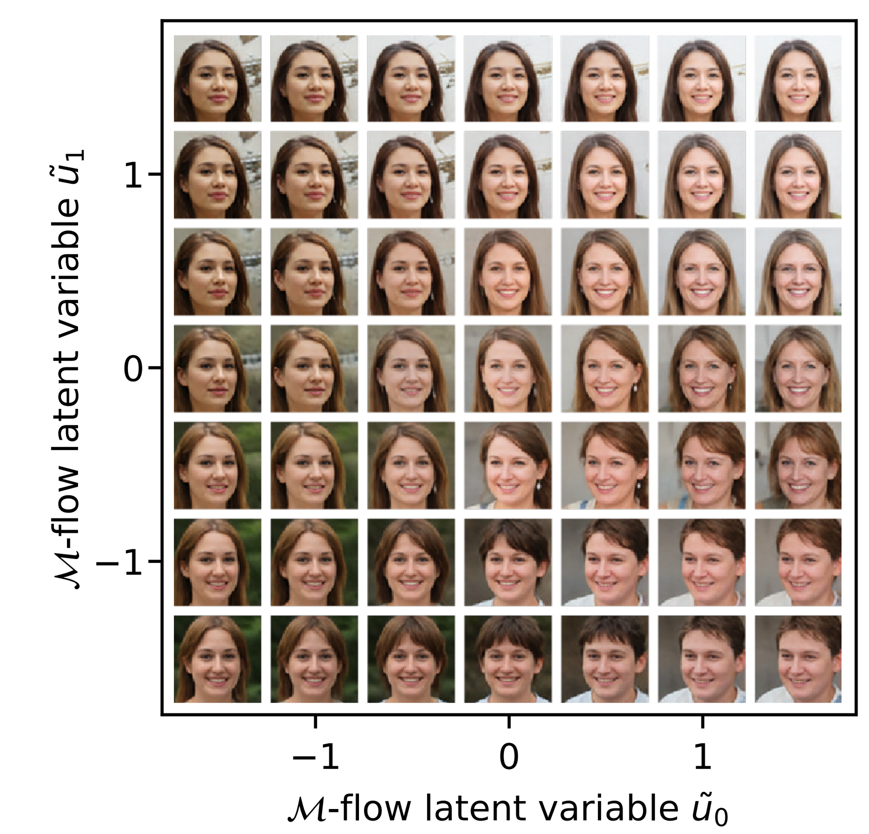
Experiments

- We implement $\mathcal{M}$-flows with neural spline flow transformations [6]
- In toy experiments, $\mathcal{M}$-flows reconstruct ground-truth manifolds and densities with higher precision than baselines





- $\mathcal{M}$-flows can learn the invariant probability density of the Lorenz attractor almost everywhere, despite its non-trivial topology
- We study 40-dimensional particle physics data, where the laws of physics predict a 14-dimensional data manifold. In a simulation-based inference setup [7], $\mathcal{M}$-flows enable higher-quality posteriors than baselines
- On image datasets, $\mathcal{M}$-flows learn high-quality manifolds, smoothly interpolating latent spaces, and outperform baselines on generative and inference tasks



References

[1] See also the extended version of this paper at arXiv:2003.13913

[2] Code at github.com/johannbrehmer/manifold-flow

[3] G. Papamakarios, E. Nalisnick, D. Rezende, S. Mohamed, B. Lakshminarayanan: "Normalizing Flows for Probabilistic Modeling and Inference", arXiv:1912.02762

[4] M. Gemici, D. Rezende, S. Mohamed: "Normalizing Flows on Riemannian Manifolds", NeurIPS 2016 workshop, arXiv:1611.02304

[5] J. Beitler, I. Sosnovik, A. Smeulders: "PIE: Pseudo-Invertible Encoder", openreview.net/forum?id=SkgiX2Aqtm

[6] C. Durkan, A. Bekasov, I. Murray, G. Papamakarios: "Neural Spline Flows", NeurIPS 2019, arXiv:1906.04032

[7] K. Cranmer, J. Brehmer, G. Louppe: "The frontier of simulation-based inference", PNAS 2020, arXiv:1911.01429