Freie Universität Berlin

AI4Science Group

Department of Mathematics and Computer Science

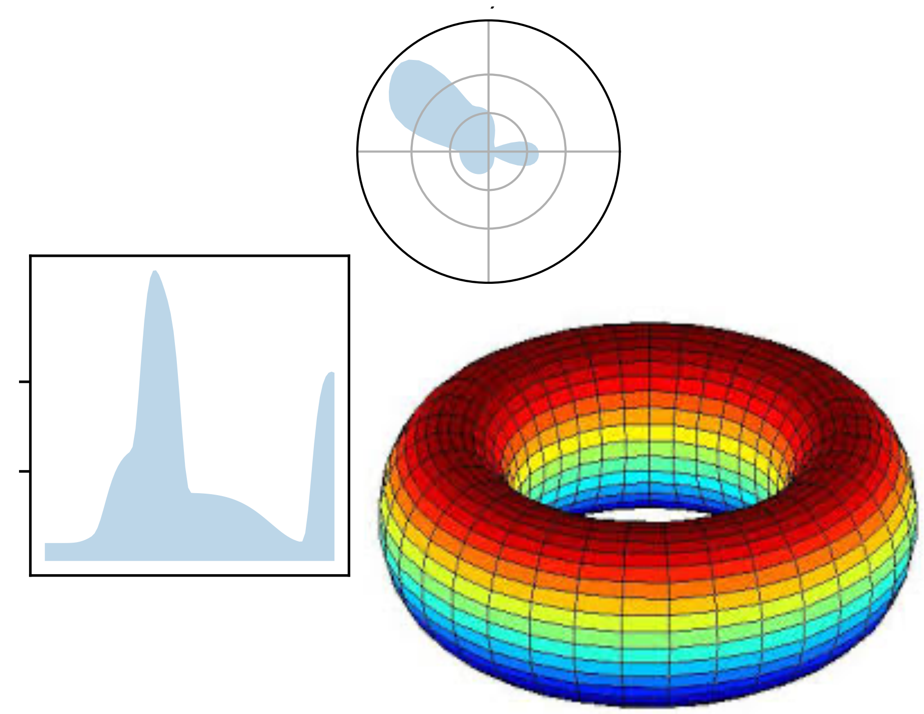


Jonas Köhler* Andreas Krämer* Frank Noé

Smooth Normalizing Flows

35th Conference on Neural Information Processing Systems (NeurIPS 2021)

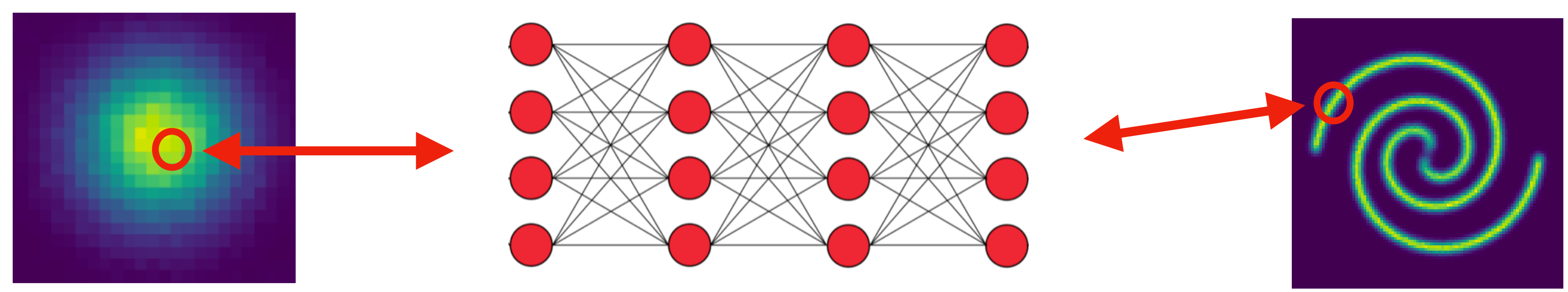
Outline



- A new class of flow transforms
  - smooth
  - expressive
  - defined on complex topologies ($\mathbb{T}^n \times [0,1]^m$)
- Efficient inversion and bidirectional training
- Utilizing smoothness (force matching; MM potentials)

Normalizing Flows

Deep Probabilistic Models



Simple Prior Distribution
$z \sim p_0$

Network
$f : \mathbb{R}^n \to \mathbb{R}^n$

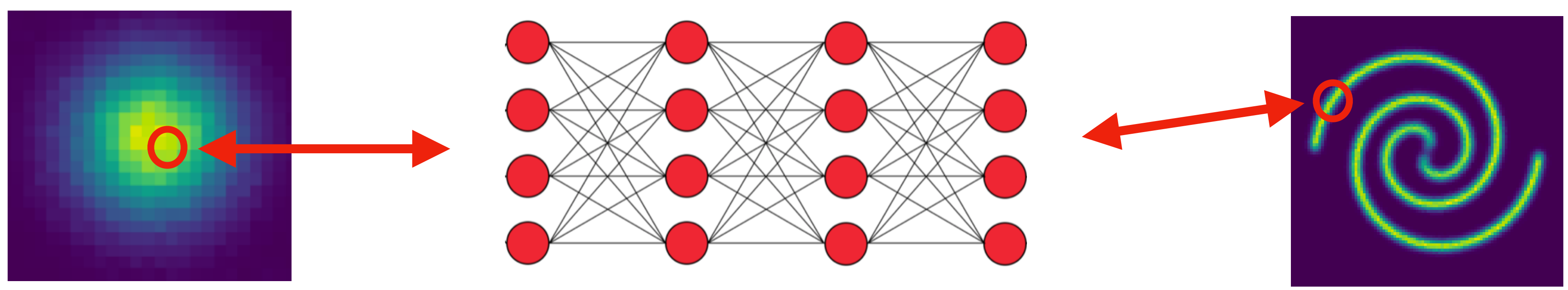
Complicated Distribution
$x = f(z) \sim p_f$

Density Estimation ←

Sampling →

Normalizing Flows

Deep Probabilistic Models



Simple Prior Distribution

$z \sim p_0$

Network

$f : \mathbb{R}^n \to \mathbb{R}^n$

Complicated Distribution

$x = f(z) \sim p_f$

$$p_f(x) = p_0\left(f^{-1}(x)\right) |\det J_{f^{-1}}(x)|$$

Normalizing Flows in Physics Applications

Generative Modeling

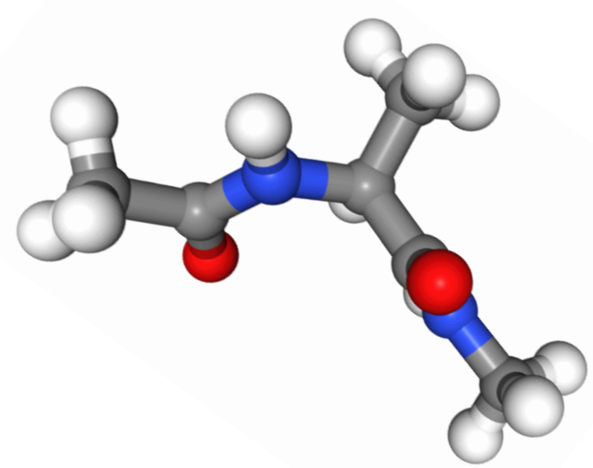
- A replacement or add-on for iterative samplers (e.g., MC, MD)

Density Estimation

- Processing of observed data (relative free energy/entropy/stability of metastable states)

Energy $u = -\log p$

Force $F = \frac{\partial}{\partial x} \log p$



Normalizing Flows in Physics Applications

Generative Modeling

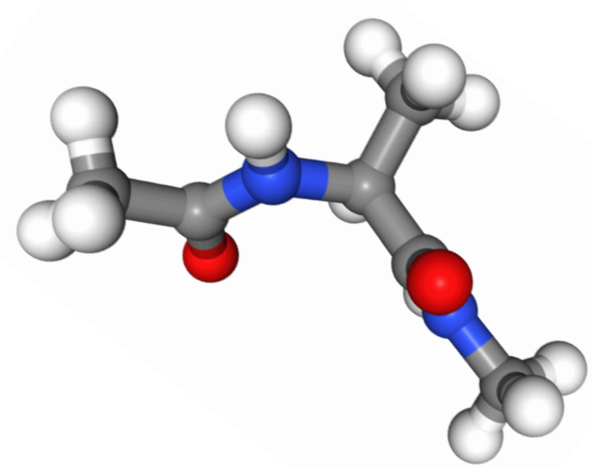
- A replacement or add-on for iterative samplers (e.g., MC, MD)

Density Estimation

- Processing of observed data (relative free energy/entropy/stability of metastable states)

Energy $u = -\log p$

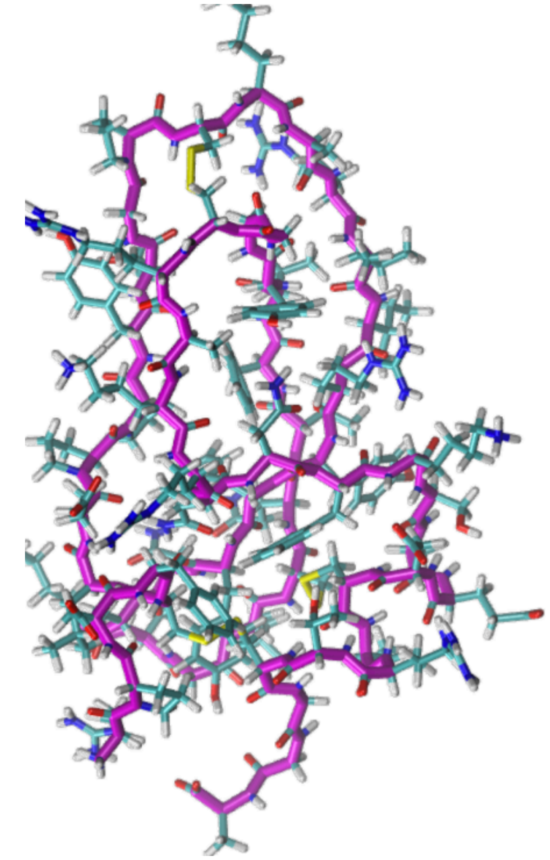
Force $F = \frac{\partial}{\partial x} \log p$



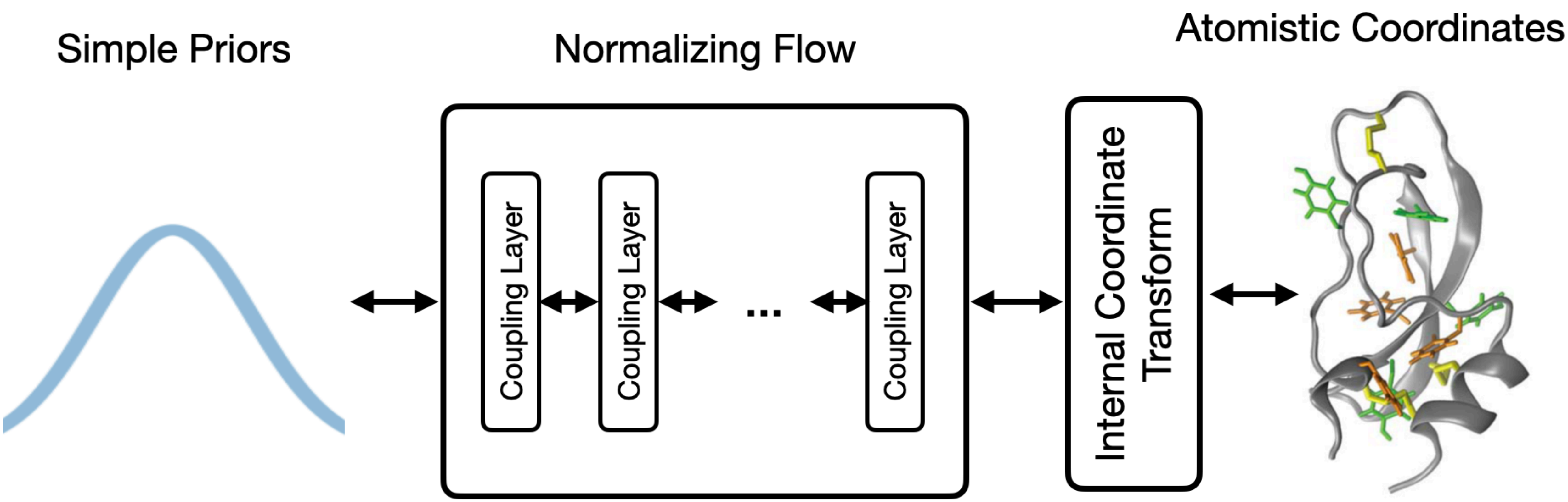
requires smooth transformations

Boltzmann Generators

Noé, Olsson, Köhler, Wu, Science (2019)

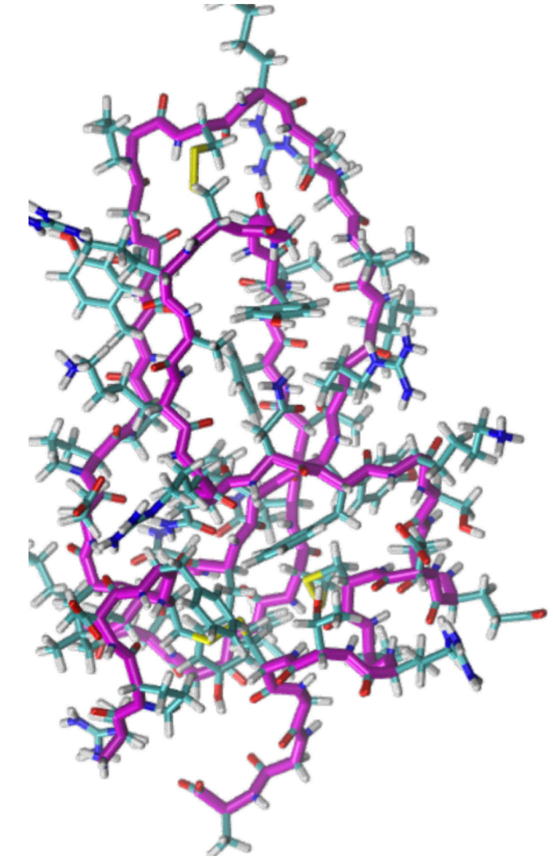


- Given: potential energy $u(x)$, MD data
- Match the Boltzmann distribution $p(x) \sim e^{-u(x)}$
- Reweight samples to the target distribution

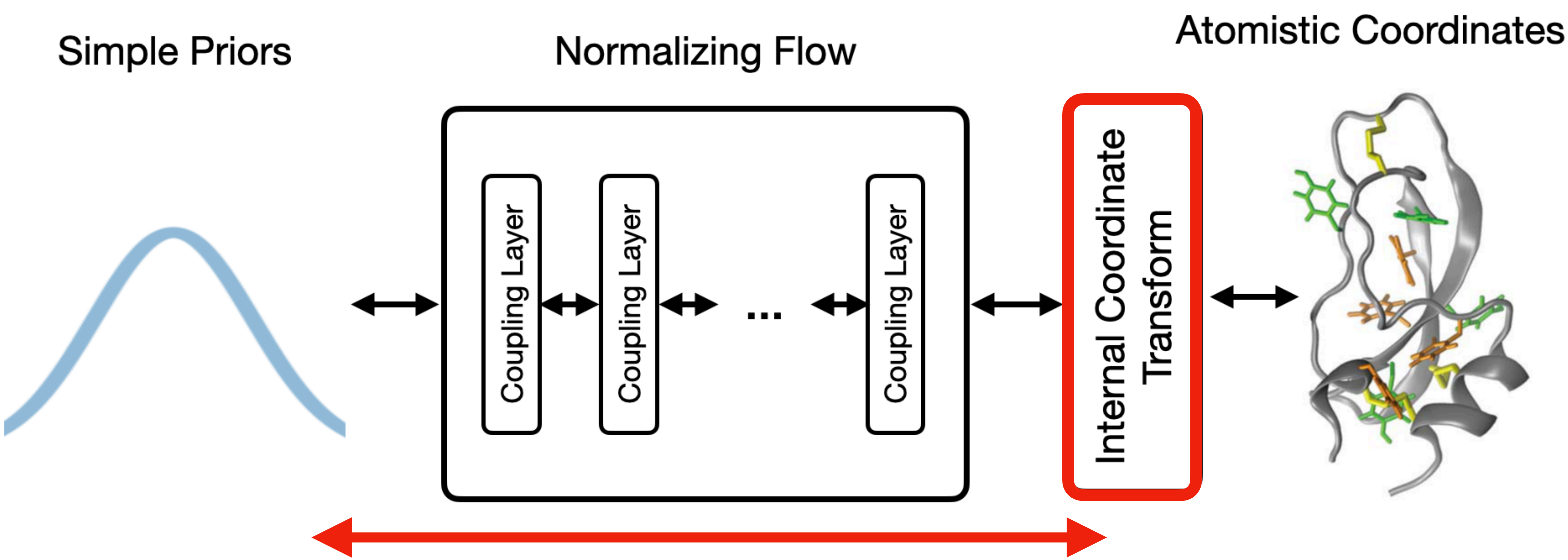


Boltzmann Generators

Noé, Olsson, Köhler, Wu, Science (2019)



- Given: potential energy $u(x)$, MD data
- Match the Boltzmann distribution $p(x) \sim e^{-u(x)}$
- Reweight samples to the target distribution

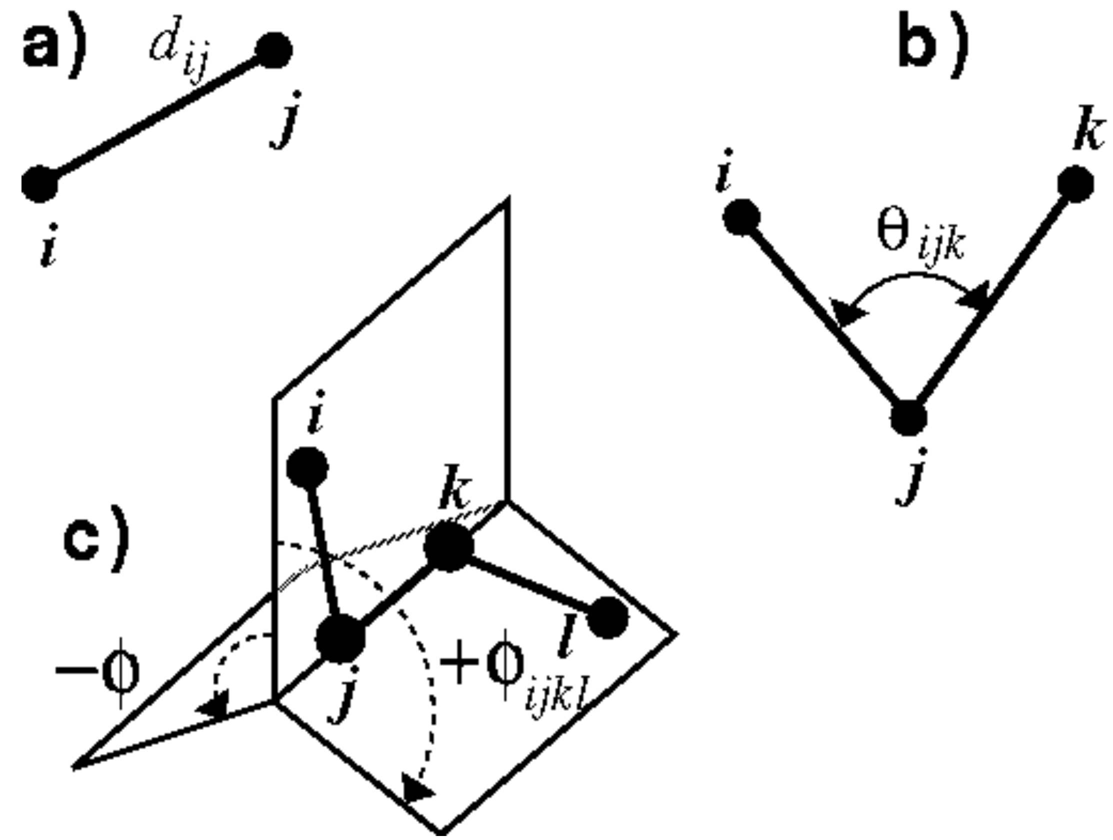


Bidirectional training requires efficient inversion.

Internal Coordinates

Topological Constraints

- Bond Lengths $d_{ij} \in (0,\infty)$
- Angles $\theta_{ijk} \in [0,\pi]$
- Torsions $\phi_{ijkl} \in S^1$



Flows operate on product spaces of tori and compact intervals.

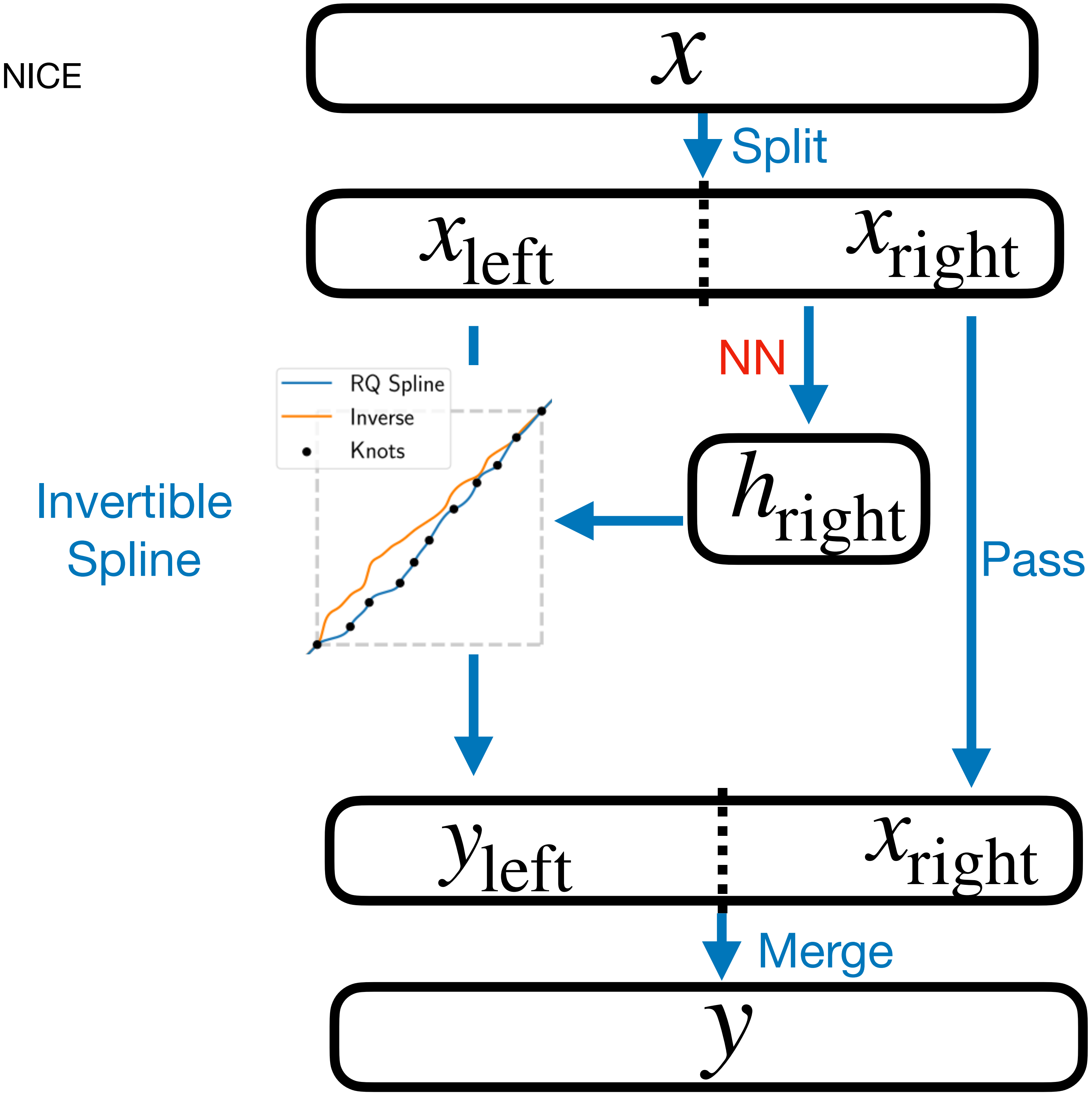
Desiderata

- We need an expressive flow architecture that …
  - … is smooth
  - … is efficient in the forward and inverse direction
  - … works on nontrivial topologies (circular and compact intervals)

Neural Spline Flows

Durkan et al. (2019): arXiv:1906.04032

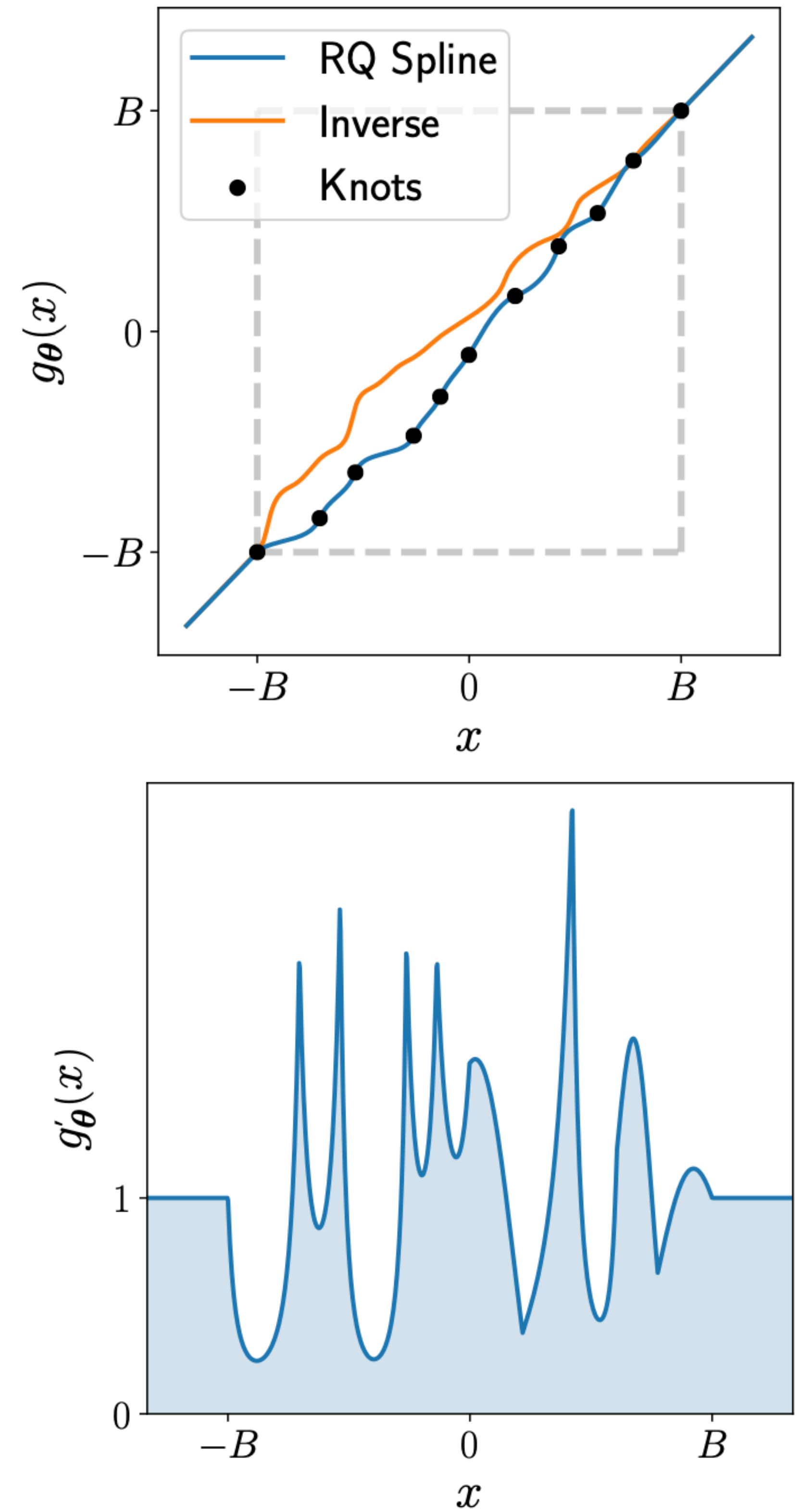
- Coupling layer Dinh et al. (2014): NICE



Neural Spline Flows

Durkan et al. (2019): arXiv:1906.04032, Rezende et al. (2020): arXiv: 2002.02428

- Coupling layer Dinh et al. (2014): NICE
- Monotonic rational quadratic splines
  - Multimodal transforms
  - Analytic inverse
- Applicable to compact intervals and circular domains Rezende et al. (2020)

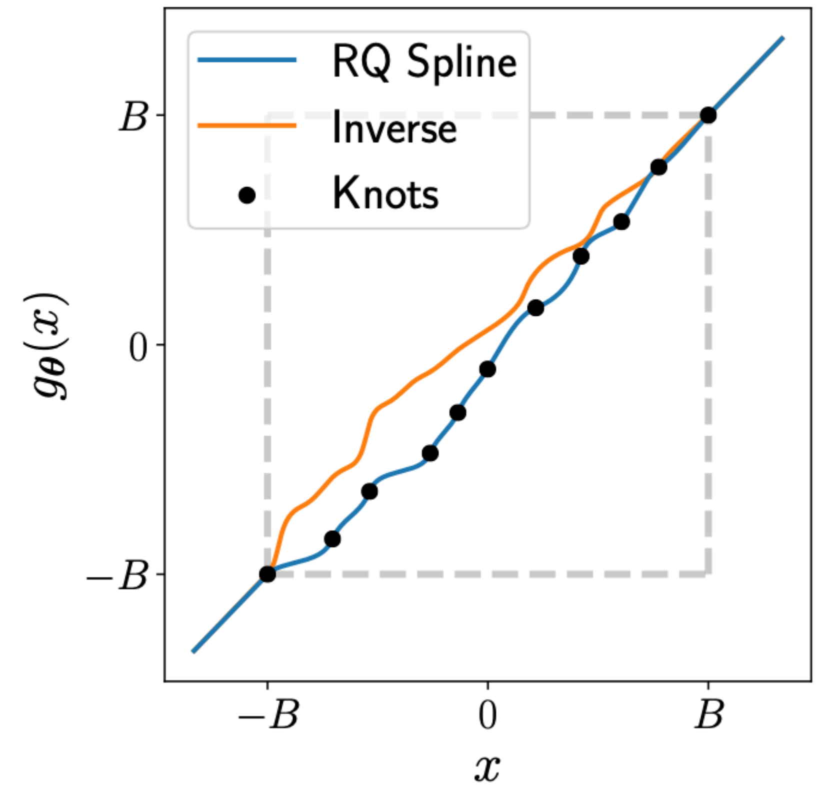


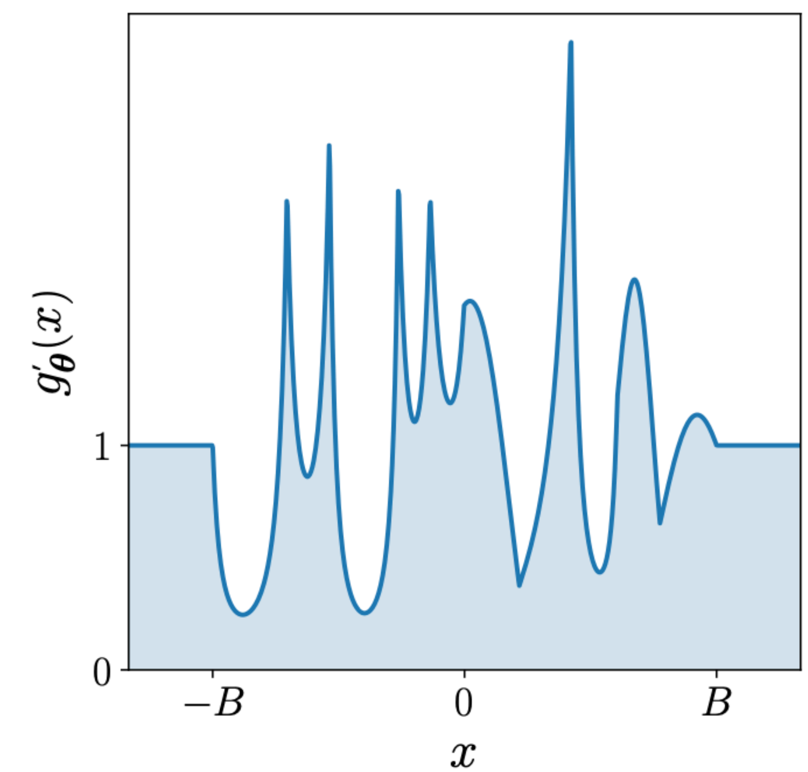
Neural Spline Flows

Durkan et al. (2019): arXiv:1906.04032, Rezende et al. (2020): arXiv: 2002.02428

- Coupling layer Dinh et al. (2014): NICE
- Monotonic rational quadratic splines
  - Multimodal transforms
  - Analytic inverse
- Applicable to compact intervals and circular domains Rezende et al. (2020)
- Discontinuous forces

Is there a smooth alternative?

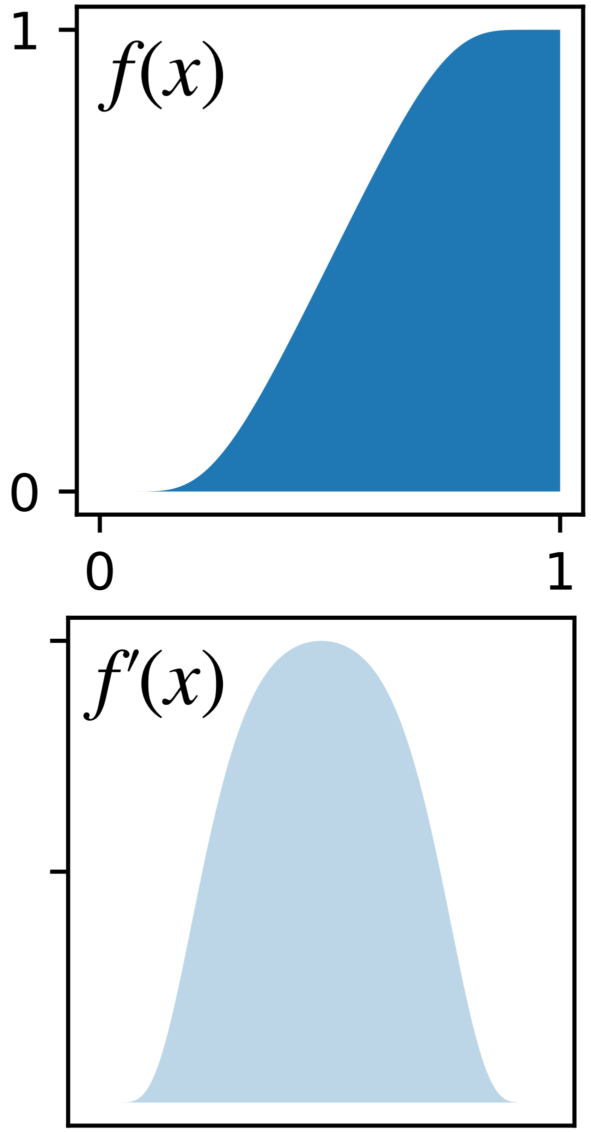




Construction of Smooth Transforms

Freie Universität Berlin

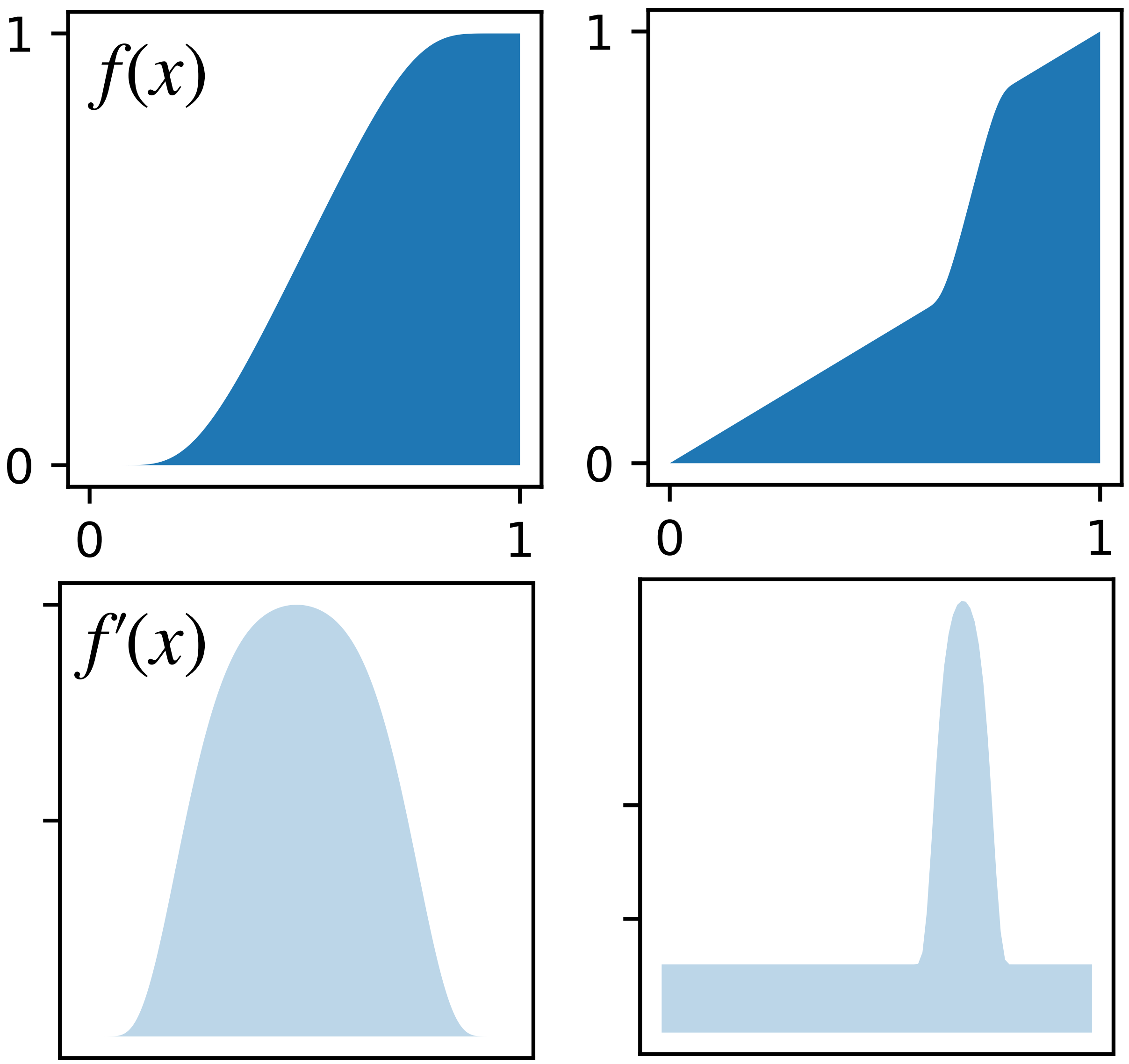
Bump Function



- Compact bump functions, e.g.
  - $\sigma(x) = \dfrac{\rho(x)}{\rho(x) - \rho(1-x)}$ with a smooth ramp $\rho(x)$
- Derivatives vanish at 0 and 1

Construction of Smooth Transforms

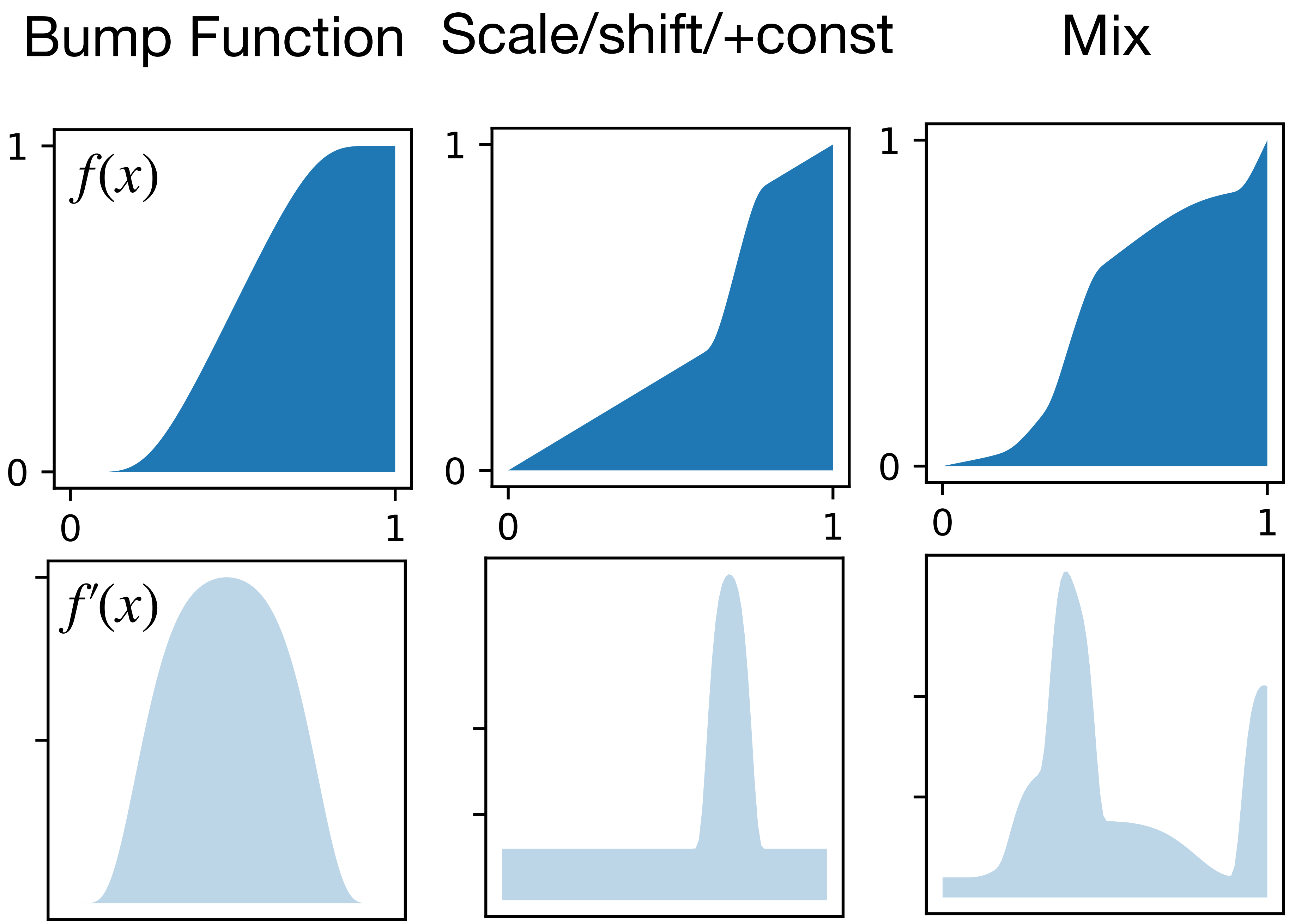
Bump Function | Scale/shift/+const



$$g(x) = \sigma(a(x-b)+0.5)$$

$$f(x) := (1-c)\cdot\left(\frac{g(x)-g(0)}{g(1)-g(0)}\right) + c\cdot x$$

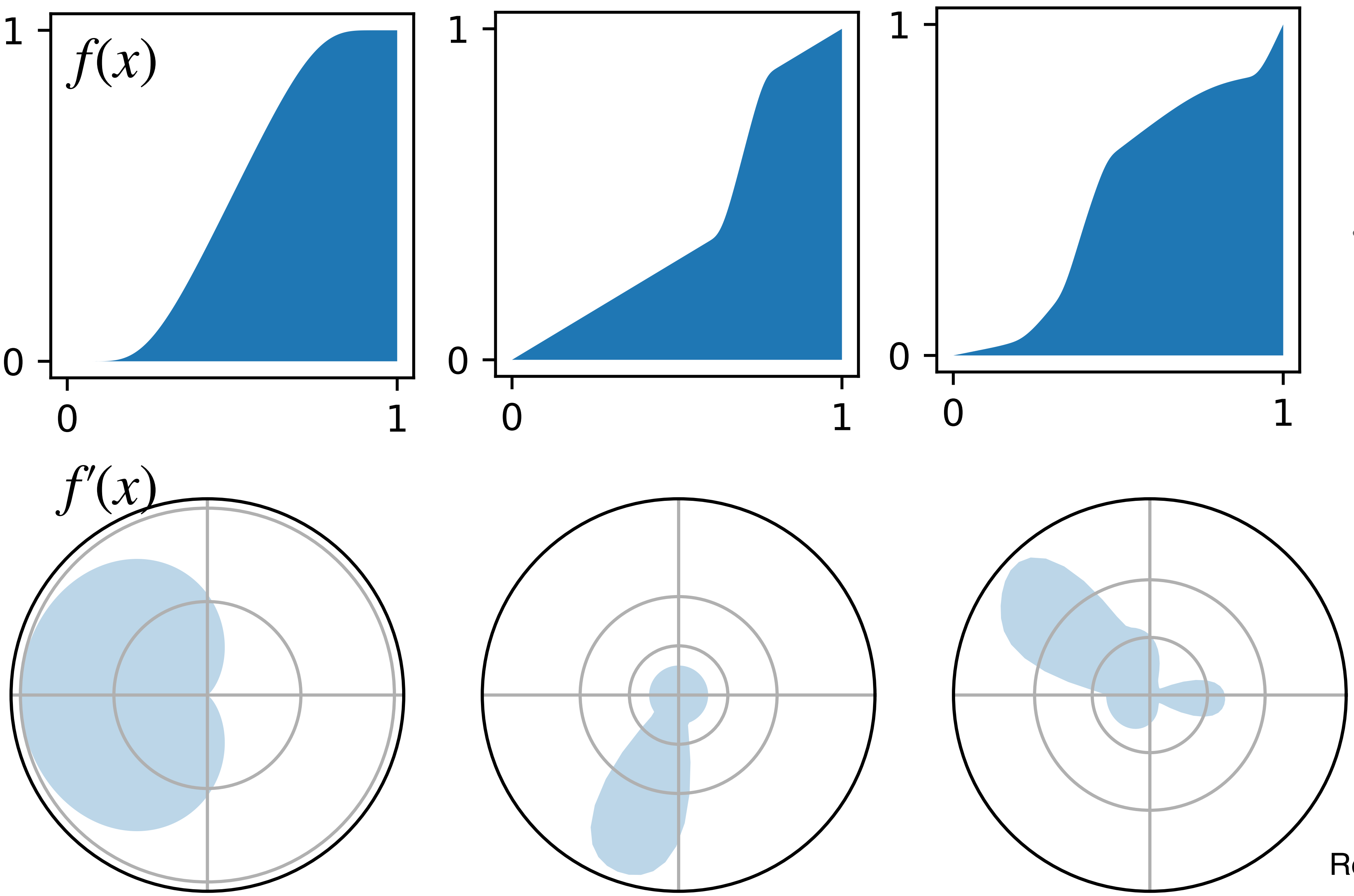
Construction of Smooth Transforms



$$f(x) = \sum w_i f_i(x)$$

Construction of Smooth Transforms

Bump Function | Scale/shift/+const | Mix



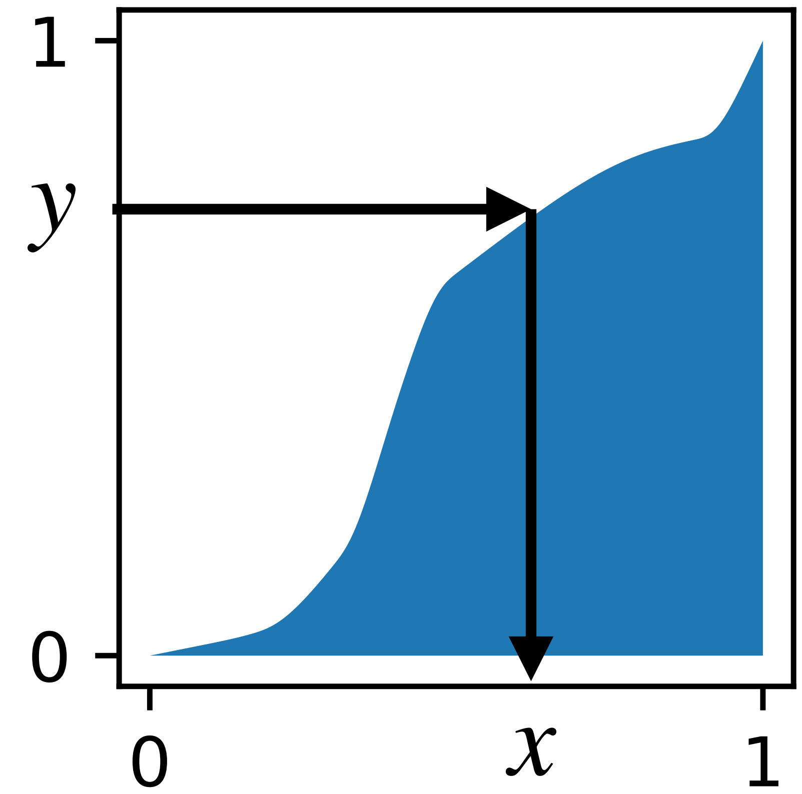
$$f(x) = \sum w_i f_i(x)$$

$$\sum_{k=1}^{n} f'(x+k)$$

Rezende et al. (2020): arXiv: 2002.02428

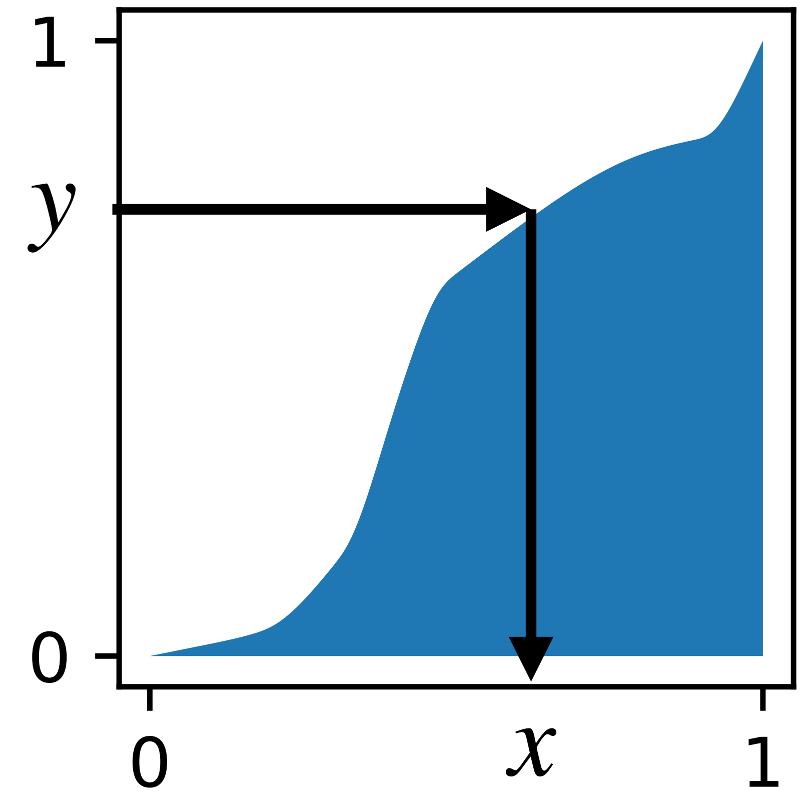
Inversion

- Non-analytic inverse
- Need to solve a 1D root-finding problem for each transform
- Bisection: one order of magnitude slower than neural spline flows



Inversion

- Non-analytic inverse
- Need to solve a 1D root-finding problem for each transform
- Bisection: one order of magnitude slower than neural spline flows

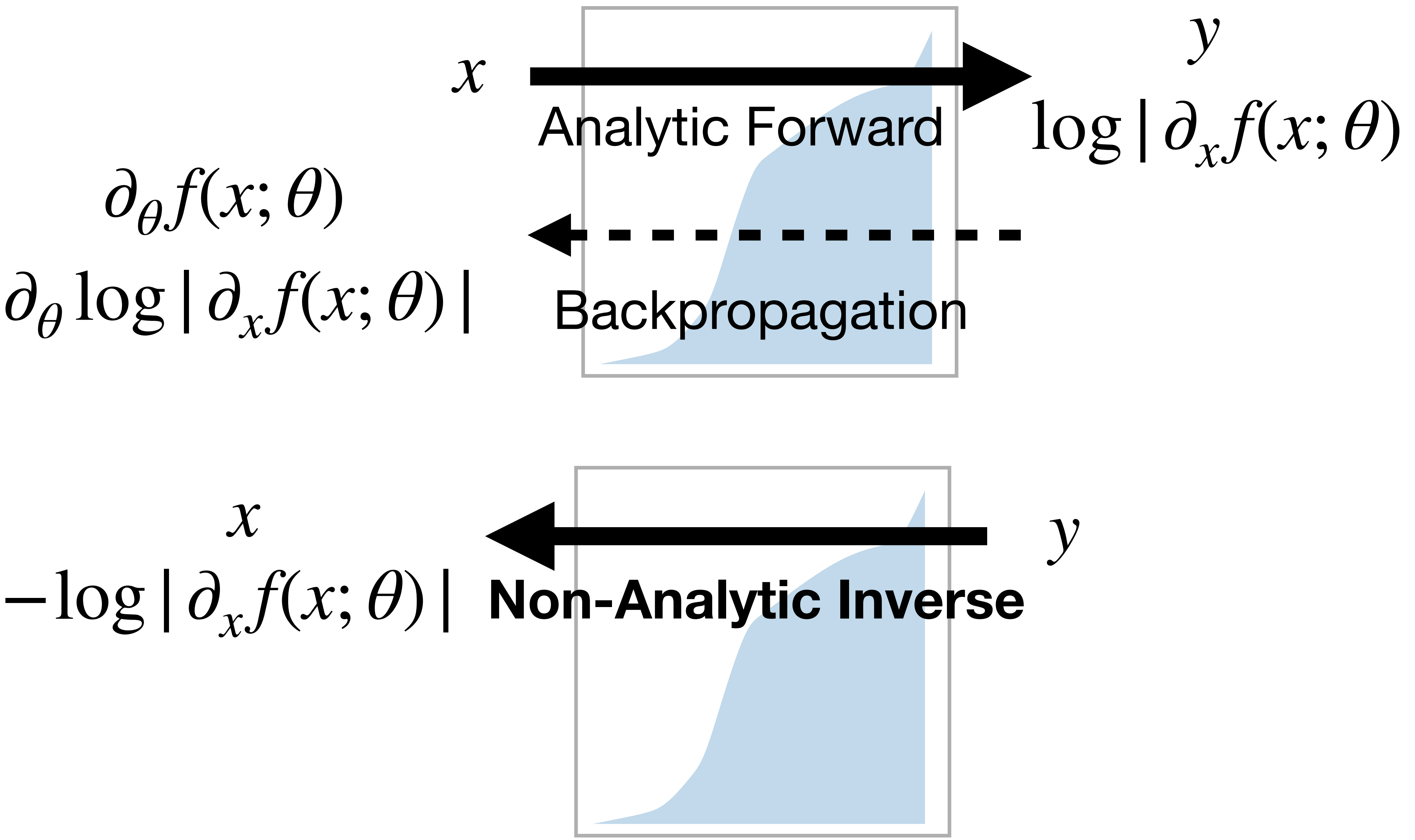


- Multi-bin bisection
    - Naive parallelism in low-dimensional (<1000-dim) applications

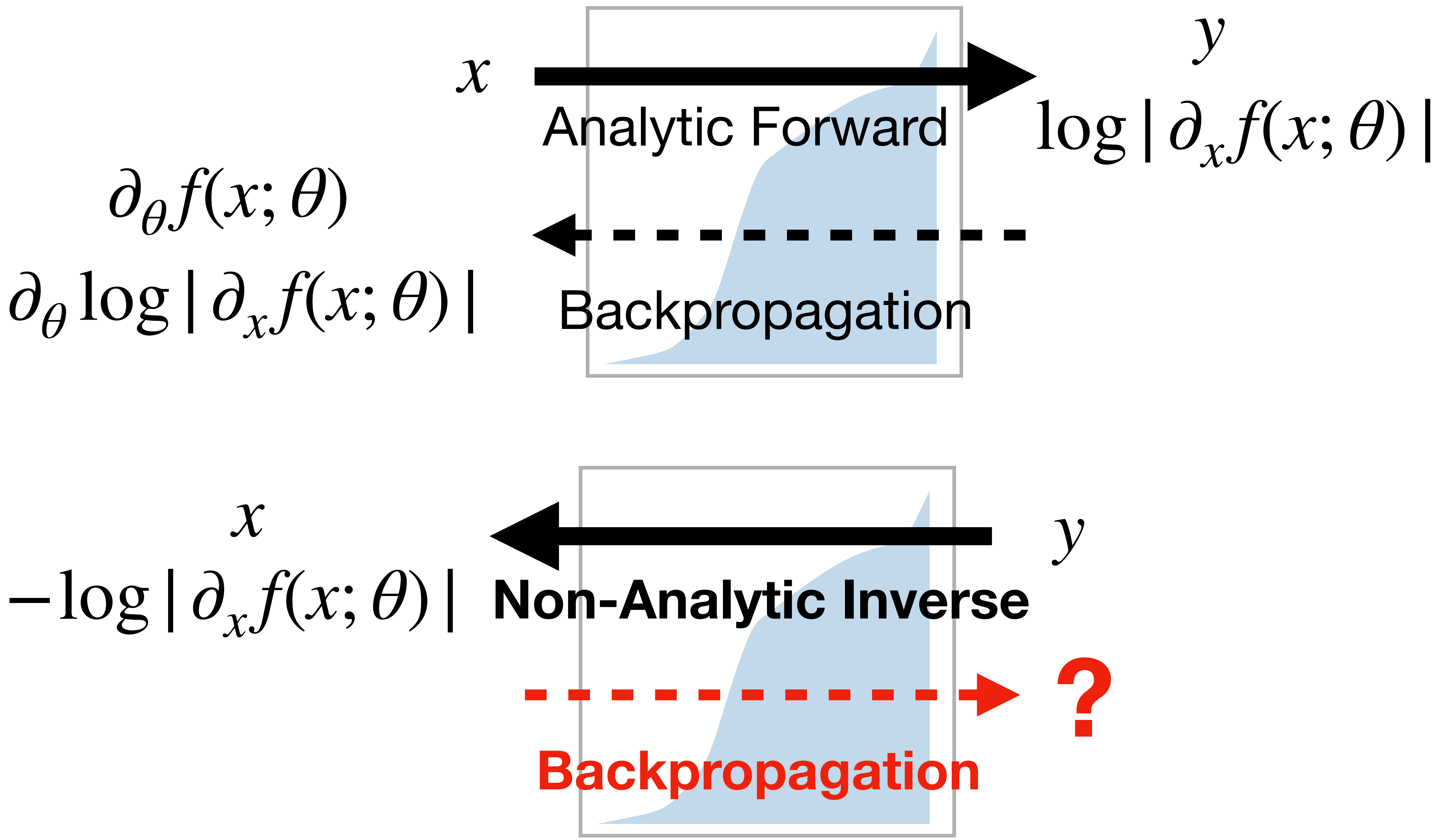
Performance vs. analytic inverse

dim	#bins	slowdown
2	128	2.1
32	32	2.7
512	4-8	6.5

Blackbox-Inversion



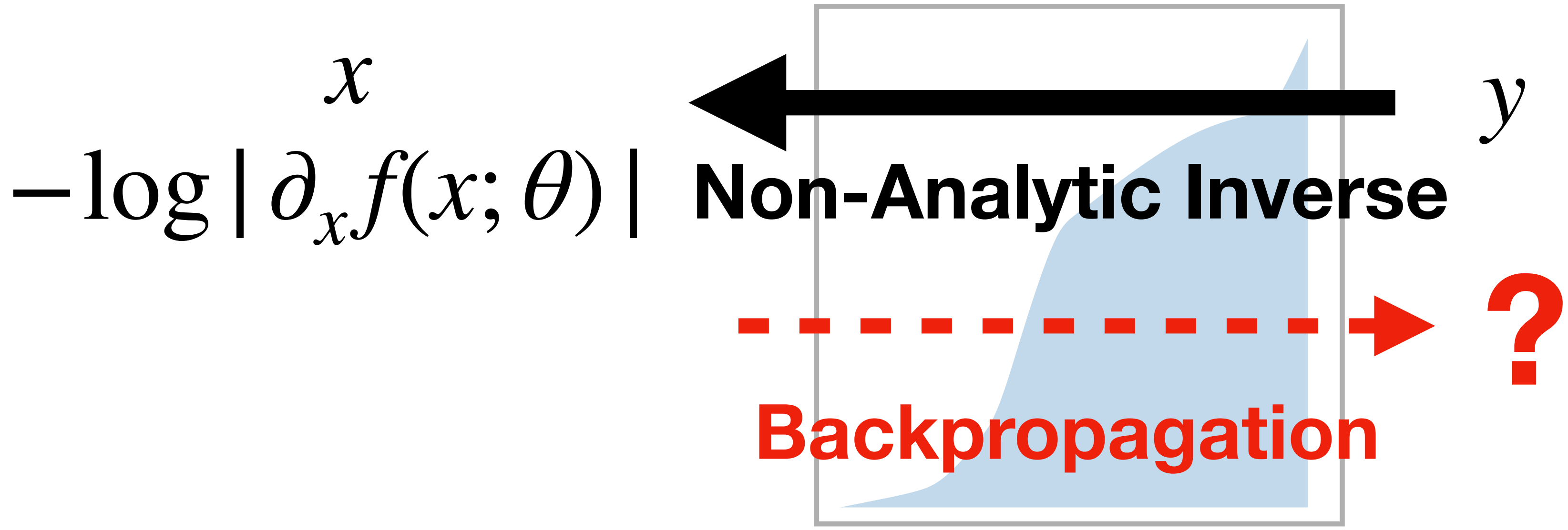
Blackbox-Inversion



Freie Universität Berlin

Blackbox-Inversion

$$
\begin{aligned}
\partial_y x(y;\boldsymbol{\theta}) &= \left(\partial_x f(x;\boldsymbol{\theta})\right)^{-1} \\
\partial_{\boldsymbol{\theta}} x(y;\boldsymbol{\theta}) &= -\left(\partial_x f(x;\boldsymbol{\theta})\right)^{-1} \partial_{\boldsymbol{\theta}} f(x;\boldsymbol{\theta}) \\
\partial_y \log|\partial_y x(y;\boldsymbol{\theta})| &= -\left(\partial_x f(x;\boldsymbol{\theta})\right)^{-1} \log|\partial_x f(x;\boldsymbol{\theta})| \\
\partial_{\boldsymbol{\theta}} \log|\partial_y x(y;\boldsymbol{\theta})| &= -\left(\partial_x f(x;\boldsymbol{\theta})\right)^{-1} \left(\log|\partial_x f(x;\boldsymbol{\theta})|\, \partial_{\boldsymbol{\theta}} f(x;\boldsymbol{\theta}) - \partial_{\boldsymbol{\theta}}\partial_x f(x;\boldsymbol{\theta})\right)
\end{aligned}
$$



Express inverse gradients through forward gradients

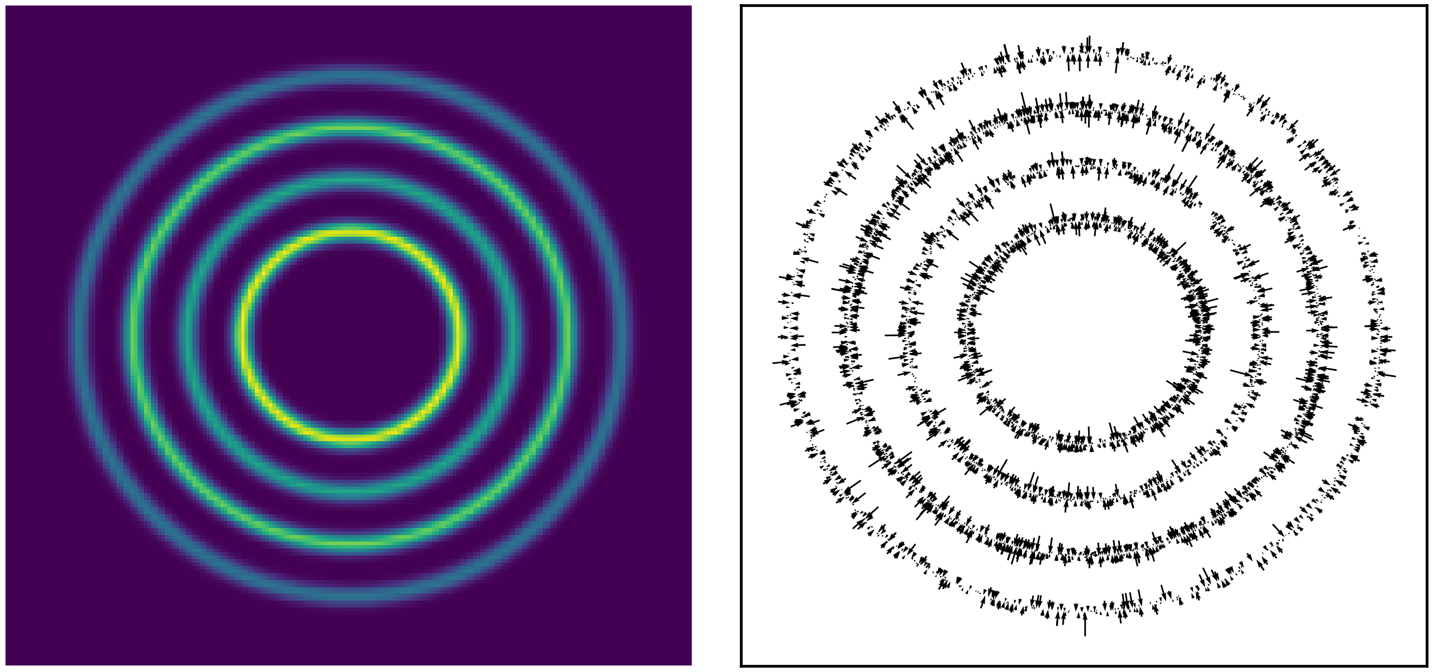
Desiderata

- We need an expressive flow architecture that …
  - … is smooth ✓
  - … is efficient in the forward and inverse direction ✓
  - … works on nontrivial topologies (circular and compact intervals) ✓

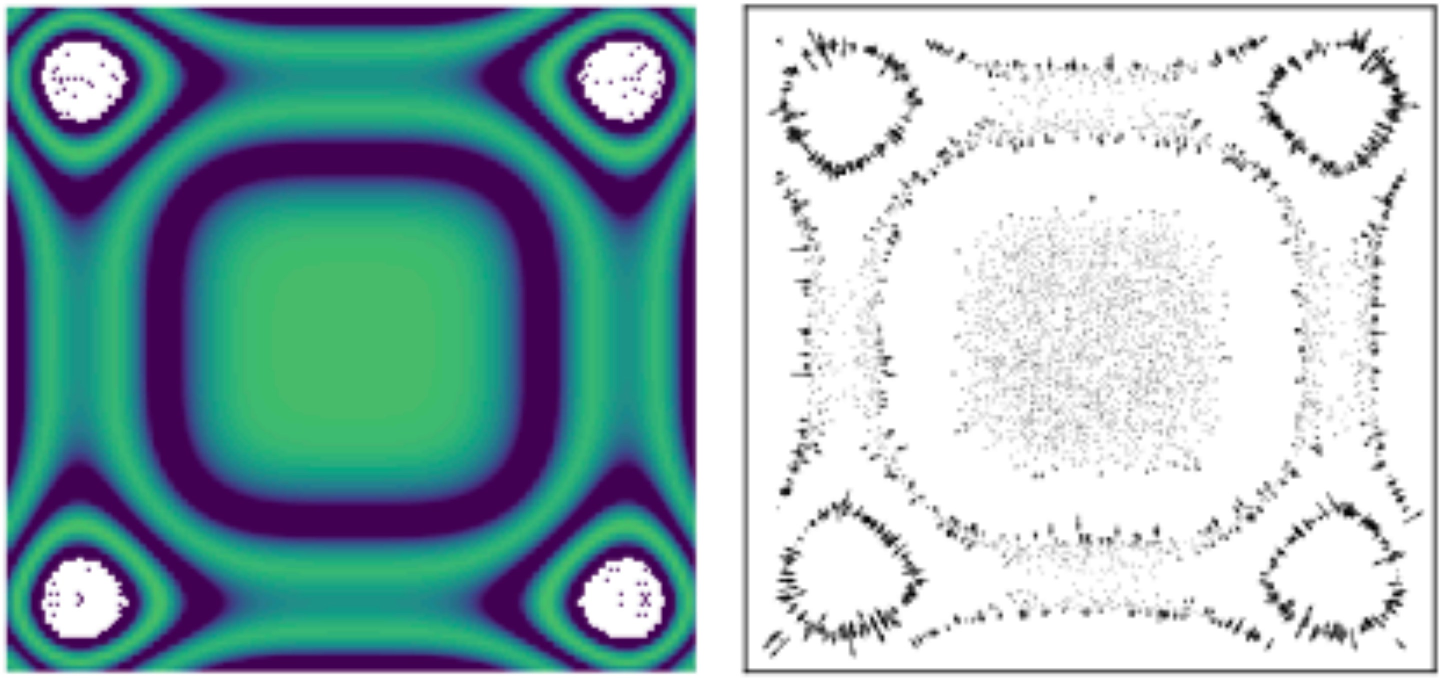
Toy Examples

- Maximum likelihood training on samples

Compact Domains



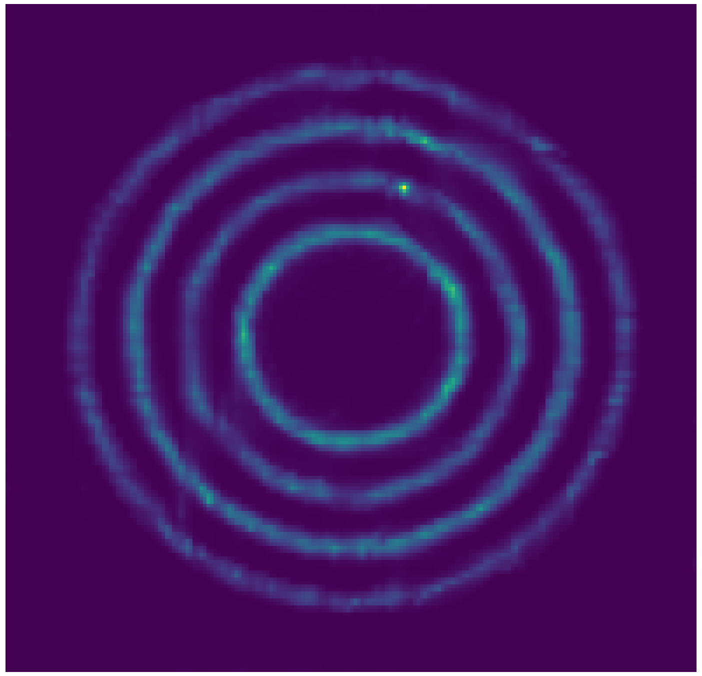
Periodic Domains

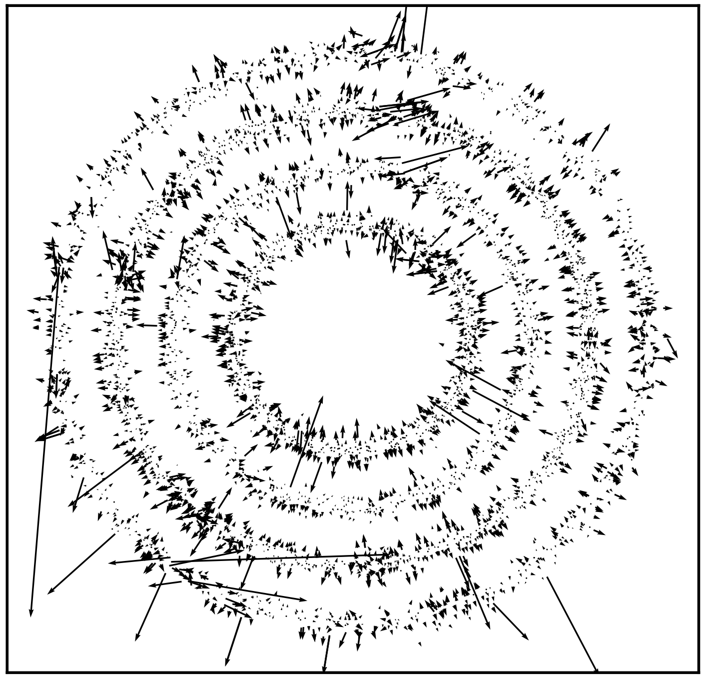


Toy Examples

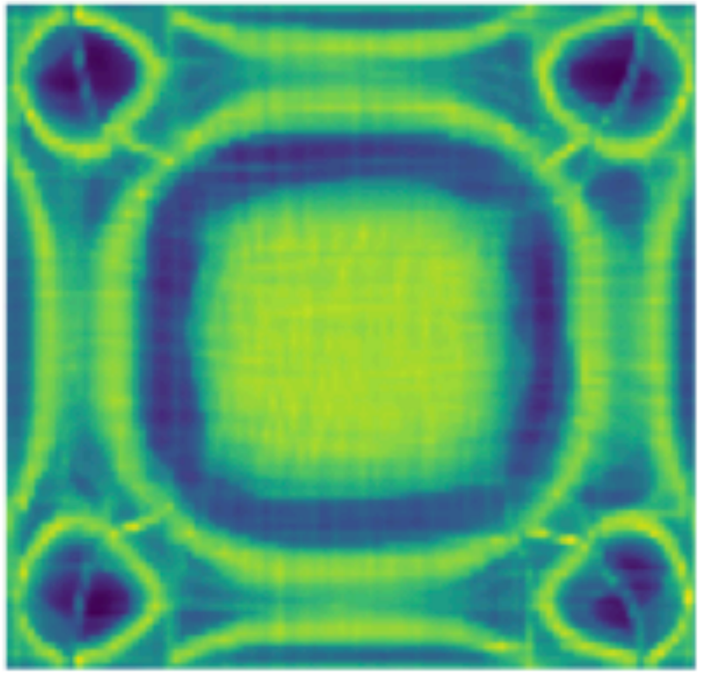
- Maximum likelihood training on samples
- Neural spline flows reproduce the density but **have discontinuous forces and extreme outliers**

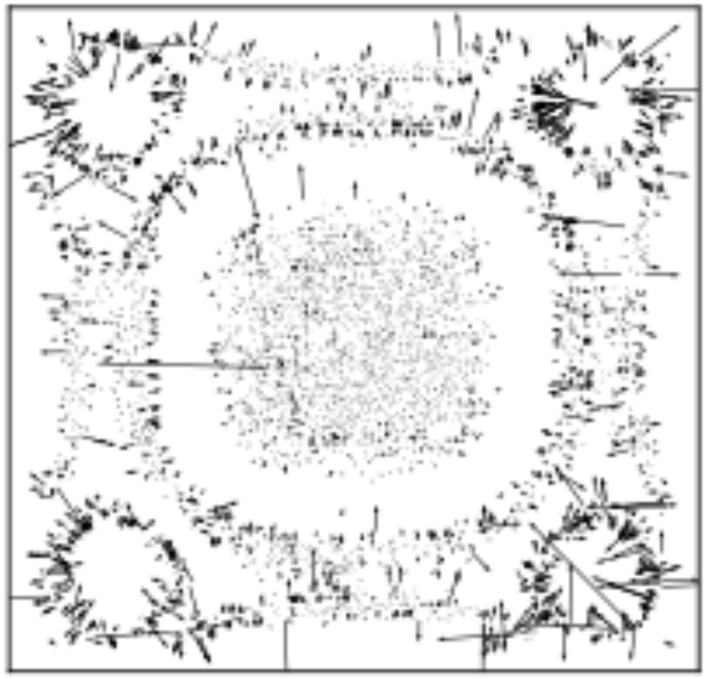
Compact Domains





Periodic Domains

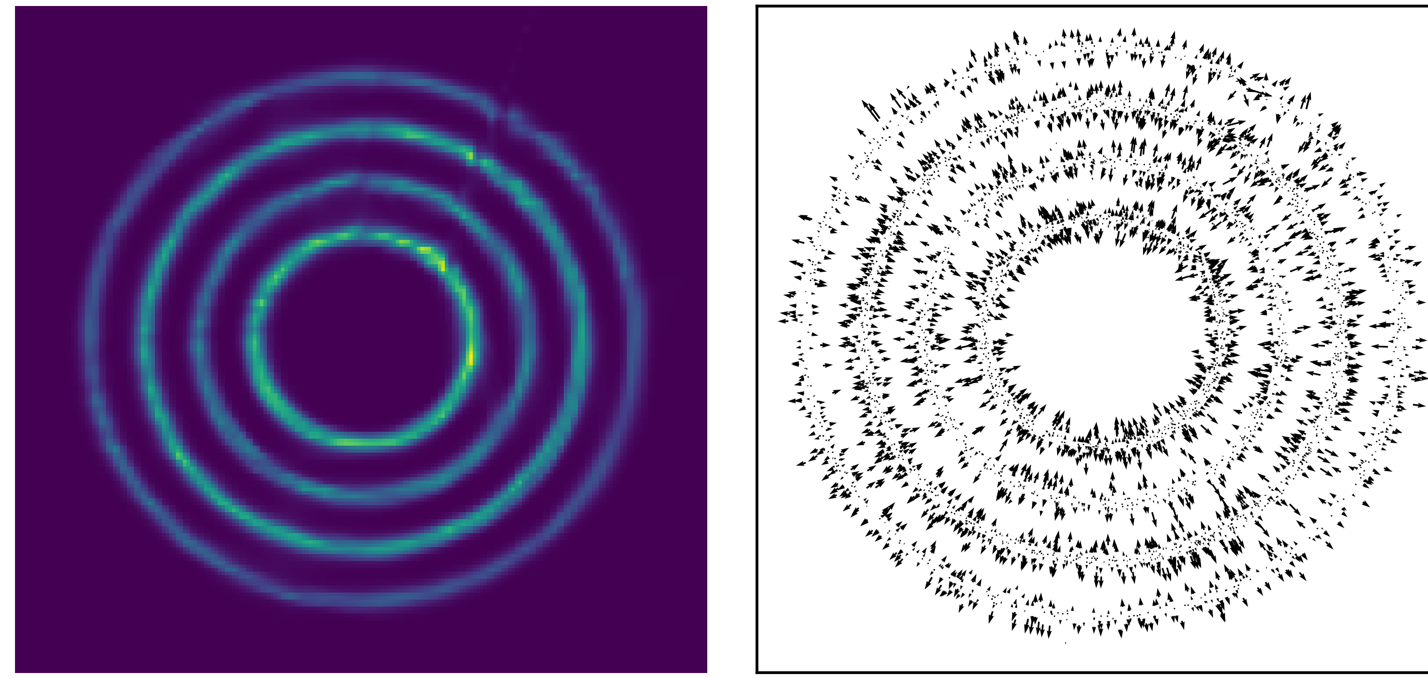




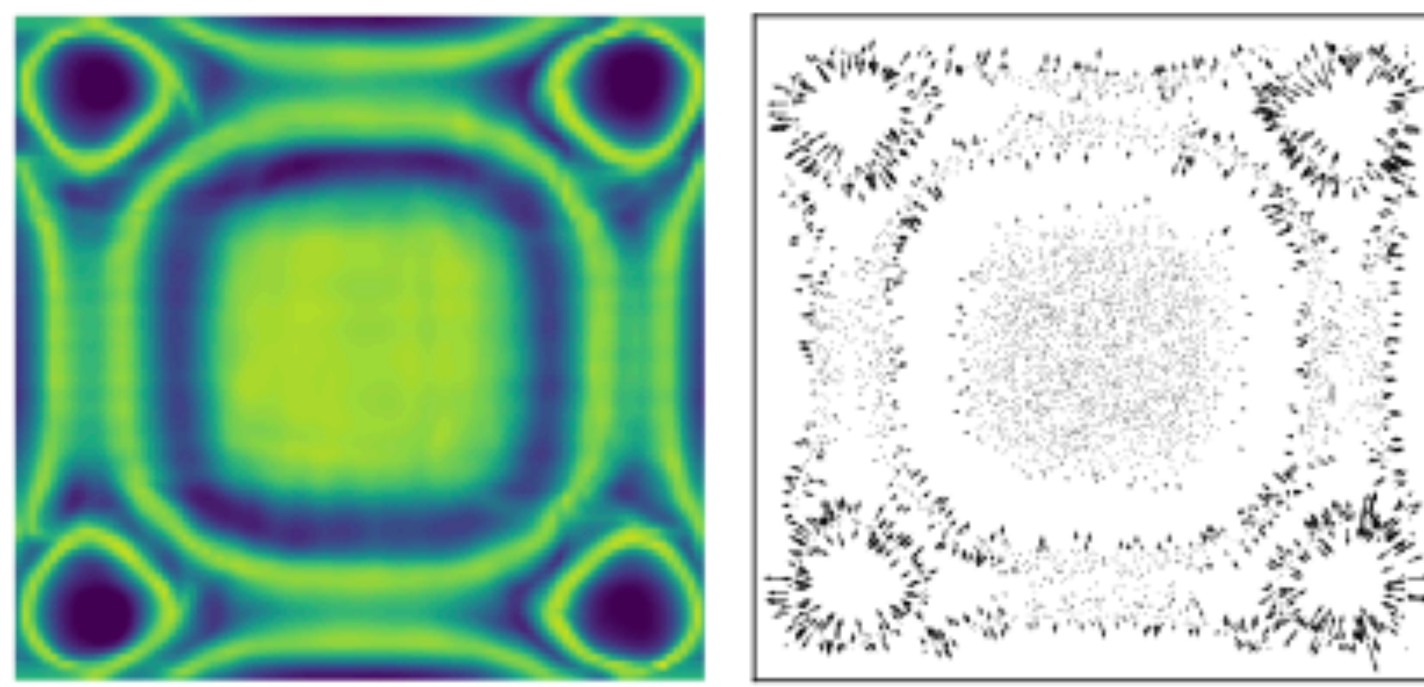
Toy Examples

- Maximum likelihood training on samples
- Neural spline flows reproduce the density but **have discontinuous forces and extreme outliers**
- Mixtures of bump functions **reproduce density and forces**

Compact Domains



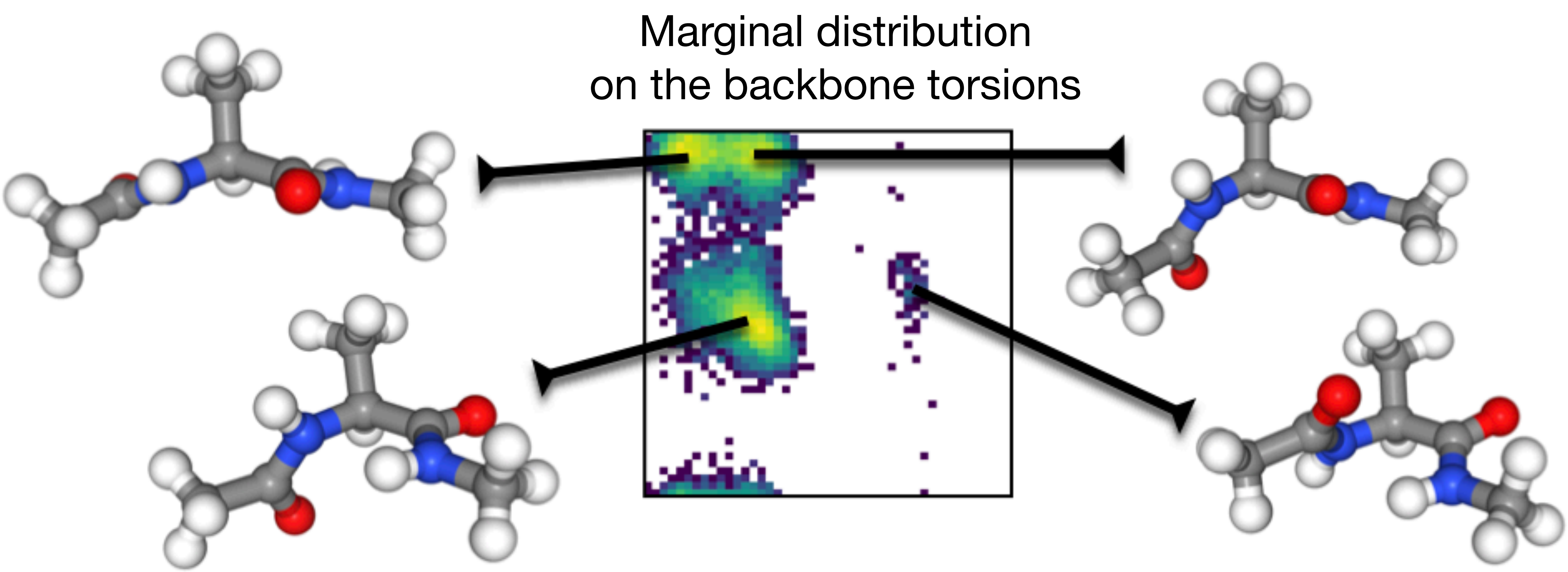
Periodic Domains



Alanine Dipeptide

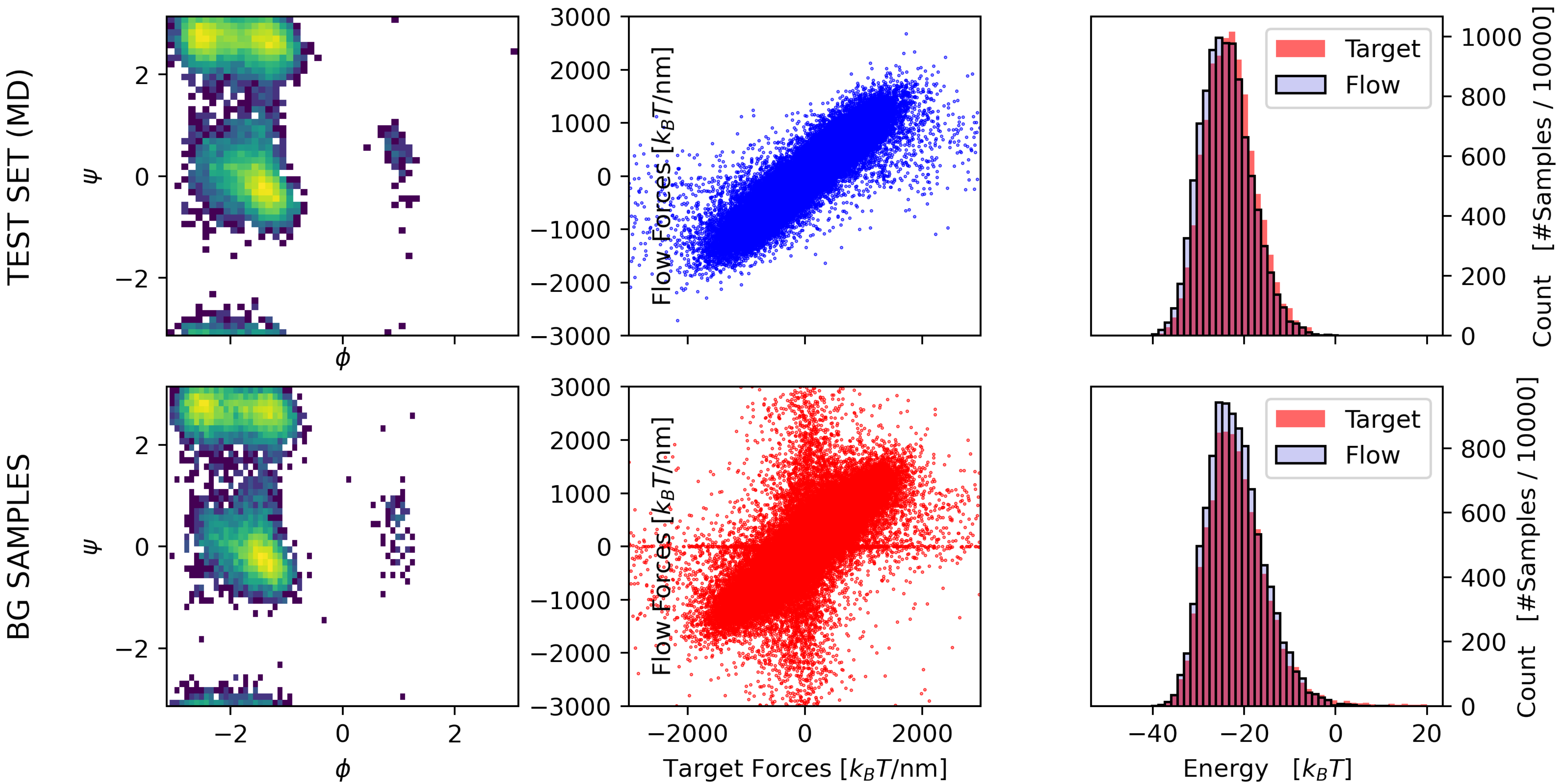
- Flow operates on 60 internal coordinates (circular and non-circular)

- Multimodal target distribution $\mu(x) = \exp(-u(x))/Z$



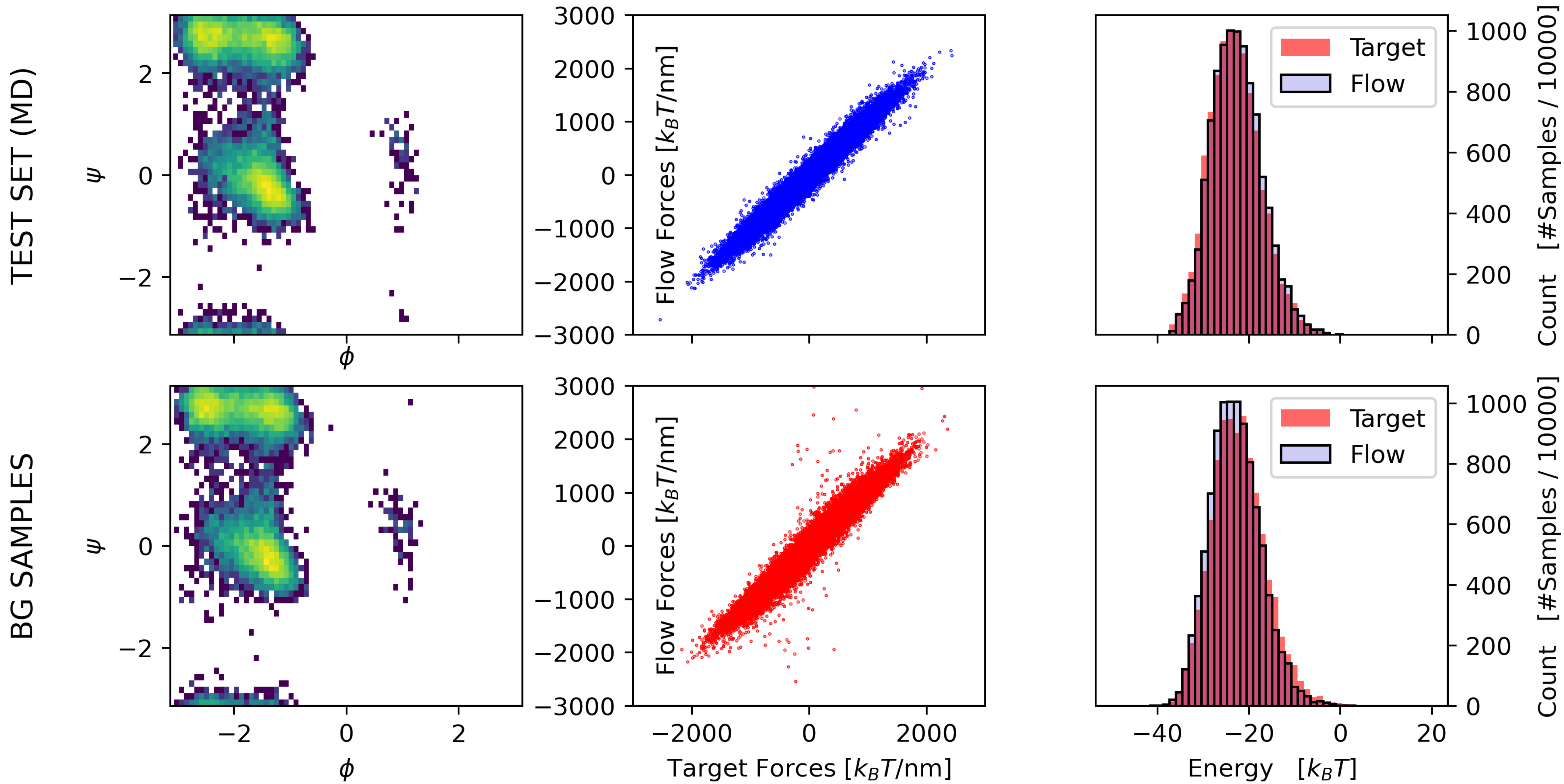
Spline Flows

Freie Universität Berlin



Sampling efficiency: 25%

Smooth Flows



Sampling efficiency 38%

Training Flows by Force Matching

Freie Universität Berlin

- Force residual with respect to ground truth forces

$$\mathcal{L}_{\mathrm{FM}}(\boldsymbol{\theta}) := \mathbb{E}_{\boldsymbol{x}\sim\mu(\boldsymbol{x})}\left[\left\|\mathbf{f}(\boldsymbol{x}) - \partial_{\boldsymbol{x}} \log p_f(\boldsymbol{x};\boldsymbol{\theta})\right\|_2^2\right]$$

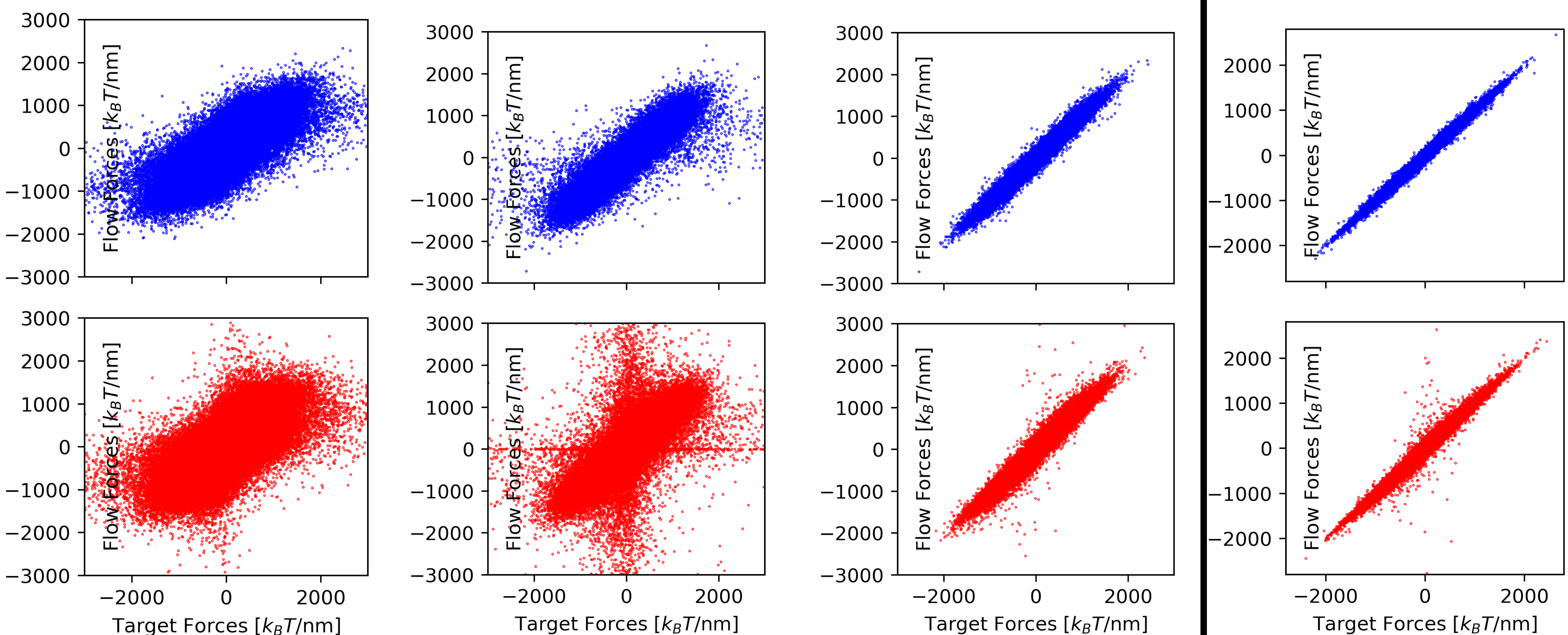
- Combine it with maximum-likelihood estimation

$$\mathcal{L}(\boldsymbol{\theta}) = \omega_n \mathcal{L}_{\mathrm{NLL}}(\boldsymbol{\theta}) + \omega_k \mathcal{L}_{\mathrm{KLD}}(\boldsymbol{\theta}) + \underline{\omega_{\mathrm{f}} \mathcal{L}_{\mathrm{FM}}(\boldsymbol{\theta})}$$

Freie Universität Berlin

Smooth Flows

RealNVP	Spline	Smooth	**Smooth+FM**



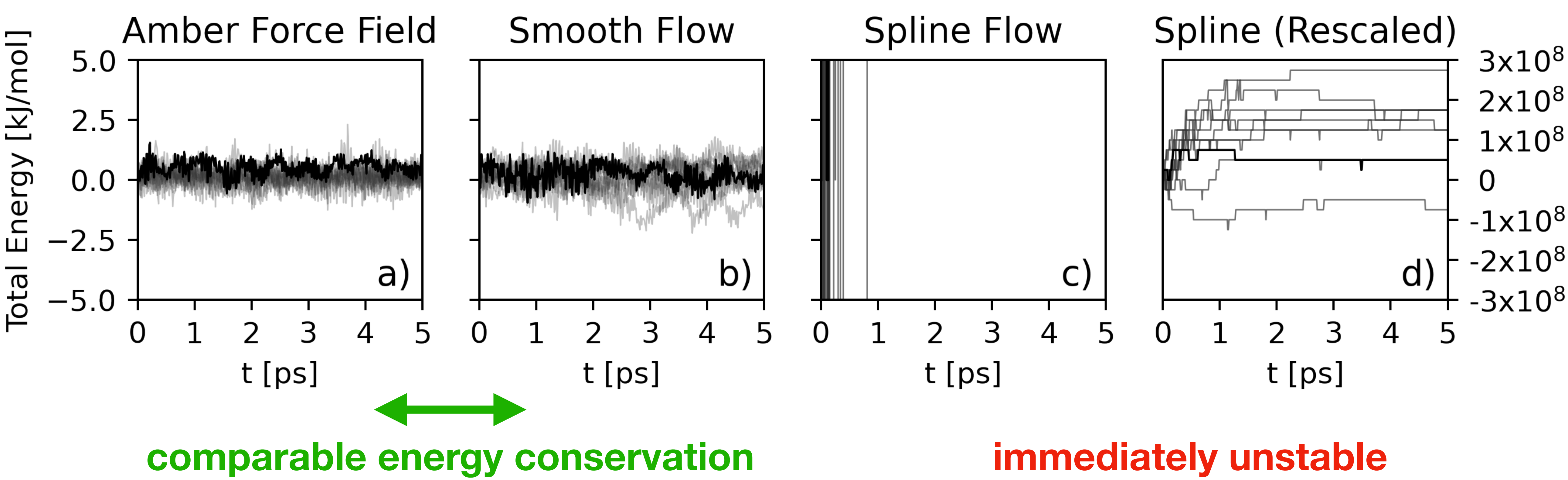
Sampling efficiency:			
<1%	25%	38%	**42%**

Köhler, Krämer, Noé: Smooth Normalizing Flows

arXiv: 2110.00351

Using Flows as Molecular Potentials

- Molecular dynamics simulations (NVE)
- Energy fluctuations are solely due to numerical integration errors
- Discontinuous forces -> energy not conserved



Conclusions

- Smooth flow architecture on compact intervals and tori
- Efficient backpropagation through black-box inversion
- Smoothness
  - improves the inductive bias for physical applications
  - enables training normalizing flows with force matching
  - opens new ways of applying normalizing flows (e.g., simulations)