(Stochastic) Normalizing Flows for lattice field theory

Alessandro Nada

Università degli Studi di Torino

29th February 2024

1st COMETA General Meeting, Izmir, 28th February–1st March 2024

Based on:
M. Caselle, E. Cellini, A. N., M. Panero, JHEP 07 (2022) 015, [arXiv:2201.08862]
M. Caselle, E. Cellini, A. N., JHEP 02 (2024) 048, [arXiv:2307.01107]

Lattice field theory simulations - a (very) quick primer

Simple case: scalar field theory on a lattice:

- discretize space-time into a square lattice of spacing a
- scalar field variables placed on sites, action discretized in a consistent way
- compute v.e.v. as in statistical mechanics

$$\langle \mathcal{O} \rangle = \frac{1}{Z} \int \prod_i \mathrm{d}\phi_i \underbrace{\mathcal{O}(\phi)}_{\text{measure}} \underbrace{\exp(-S(\phi))}_{\text{sample}}$$

  with the very complicated probability distribution $p(\phi) = \exp(-S(\phi))/Z$
- perform continuum extrapolation $a \to 0$

Lattice field theories need an efficient way to generate configurations ϕ according to $p(\phi)$

Lattice field theory simulations - a (very) quick primer

Simple case: scalar field theory on a lattice:

- discretize space-time into a square lattice of spacing a
- scalar field variables placed on sites, action discretized in a consistent way
- compute v.e.v. as in statistical mechanics

$$\langle \mathcal{O} \rangle = \frac{1}{Z} \int \prod_i \mathrm{d}\phi_i \underbrace{\mathcal{O}(\phi)}_{\text{measure}} \underbrace{\exp(-S(\phi))}_{\text{sample}}$$

  with the very complicated probability distribution $p(\phi) = \exp(-S(\phi))/Z$
- perform continuum extrapolation $a \to 0$

Lattice field theories need an efficient way to generate configurations ϕ according to $p(\phi)$

Elegant numerical solution: generate a (thermalized) Markov chain

$$\underbrace{\phi^{(0)} \xrightarrow{P_p} \phi^{(1)} \xrightarrow{P_p} \dots \xrightarrow{P_p}}_{\text{thermalization}} \underbrace{\phi^{(t)} \xrightarrow{P_p} \phi^{(t+1)} \xrightarrow{P_p} \dots \to \phi^{(t+N_{\text{conf}})}}_{\text{equilibrium}}$$

Compute $\hat{\mathcal{O}} = \frac{1}{N_{\text{conf}}} \sum_n \mathcal{O}(\phi^{(n)})$

Critical slowing down

The configurations sampled sequentially in a Markov Chain are **autocorrelated**

$$\cdots \to \phi^{(t)} \to \phi^{(t+1)} \to \cdots \to \phi^{(t+n)}$$

The measure of this autocorrelation is given by τ_{int}

$\to$ # effectively independent configurations $= n/2\tau_{\text{int}}$

Critical slowing down

The configurations sampled sequentially in a Markov Chain are **autocorrelated**

$$\dots \to \phi^{(t)} \to \phi^{(t+1)} \to \dots \to \phi^{(t+n)}$$

The measure of this autocorrelation is given by τ_{int}
$\to$ # effectively independent configurations $= n/2\tau_{\text{int}}$

When a critical point is approached τ_{int} diverges
$\to$ **critical slowing down**

The continuum limit $a \to 0$ is a critical point, so

$$\tau_{\text{int}}(\mathcal{O}) \sim a^{-z} \qquad \text{or} \qquad \tau_{\text{int}}(\mathcal{O}) \sim \exp(\alpha/a)$$

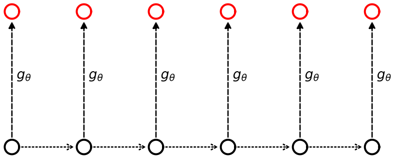
Configurations become more and more autocorrelated as the lattice spacing gets finer

Particularly severe for **topological** observables (see e.g. **[Schaefer; 1009.5228]**)

What if every new configuration is sampled independently from the previous one?

What if every new configuration is sampled independently from the previous one?

Try to model the target $p(\phi)$ by a mapping to a tractable distribution $q_0(z)$



Normalizing Flows might be a deep generative architecture efficient enough to provide this mapping

Deeply related to the idea of **trivializing maps [Lüscher; 0907.5491]**

Normalizing flow for lattice field theory

(Discrete) Normalizing Flows successfully applied in 2D:

- ϕ^4 scalar field theory: **[Albergo et al.; 1904.12072]**, **[Kanwar et al.; 2003.06413]**, **[Nicoli et al.; 2007.07115]**, **[Del Debbio et al.; 2105.12481]**
- gauge theories: $\mathrm{SU}(3)$ **[Boyda et al.; 2008.05456]** and $\mathrm{U}(1)$ **[Singha et al.; 2306.00581]**
- including fermions **[Albergo et al.; 2106.05934]**: Schwinger model **[Finkenrath; 2201.02216]** **[Albergo et al.; 2202.11712]** and $\mathrm{SU}(3)$ **[Abbott et al.; 2207.08945]**

First proof-of-concept for QCD **[Abbott et al.; 2208.03832]** and $\mathrm{SU}(3)$ in 4D **[Abbott et al.; 2305.02402]**; further applications already within reach **[Abbott et al.; 2401.10874]**

Alternative architectures:

- Continuous Normalizing Flows for ϕ^4 scalar theory **[Gerdes et al.; 2207.00283]**, Nambu-Goto string model **[Caselle et al.; 2307.01107]**
- Trivializing maps for $\mathrm{SU}(3)$ theory in 2D **[Bacchio et al.; 2212.08469]**
- Generalized with the use stochastic methods: SNFs **[Caselle et al.; 2201.08862]**, CRAFT **[Matthews et al.; 2201.13117]**

For a review check out plenary talk by Tej Kanwar at Lattice2023

Normalizing flows: structure

Normalizing Flows are a deterministic mapping

$$g_\theta(\phi_0) = (g_N \circ \cdots \circ g_1)(\phi_0) \qquad \phi_0 \sim q_0$$

composed of N invertible transformations $\rightarrow$ **coupling layers** g_i

Normalizing flows: structure

Normalizing Flows are a deterministic mapping

$$g_\theta(\phi_0) = (g_N \circ \cdots \circ g_1)(\phi_0) \qquad \phi_0 \sim q_0$$

composed of N invertible transformations $\rightarrow$ **coupling layers** g_i

In each layer the field variables ϕ are transformed

$$\phi_{n+1} = g_n(\phi_n)$$

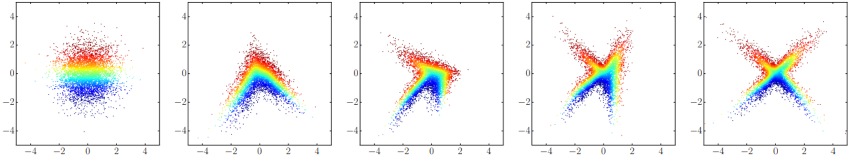


figure from **[Papamakarios; 1912.02762]**

Normalizing flows: structure

Normalizing Flows are a deterministic mapping

$$g_\theta(\phi_0) = (g_N \circ \cdots \circ g_1)(\phi_0) \qquad\qquad \phi_0 \sim q_0$$

composed of N invertible transformations $\rightarrow$ **coupling layers** g_i

In each layer the field variables ϕ are transformed

$$\phi_{n+1} = g_n(\phi_n)$$

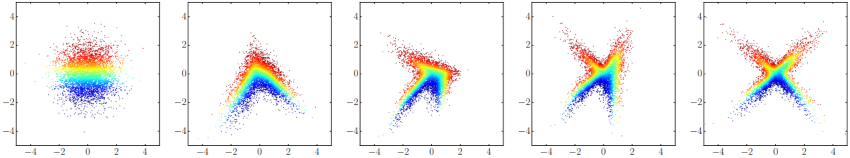


figure from **[Papamakarios; 1912.02762]**

The generated distribution for the output ϕ is

$$q(\phi) = q_0(g_\theta^{-1}(\phi)) \prod_n |\det J_n(\phi_n)|^{-1}$$

and depends on the **prior** distribution q_0 and on the Jacobian of the transformation

Discrete Normalizing flows: affine layers

Transformations g_n must be invertible + the Jacobian has to be efficiently computable

Affine layers meet this criteria (**RealNVP** architecture **[Dinh et al.; 1605.08803]**)

- Divide variables ϕ into two partitions A and B
- One is kept "frozen" while the other is transformed following

$$g_n : \begin{cases} \phi_{\mathrm{A}}^{n+1} = \phi_{\mathrm{A}}^{n} \\ \phi_{\mathrm{B}}^{n+1} = e^{-s(\phi_{\mathrm{A}}^{n})} \phi_{\mathrm{B}}^{n} + t(\phi_{\mathrm{A}}^{n}) \end{cases}$$

- s and t are the neural networks where the trainable parameters θ are

Natural choice for lattice variables: checkerboard (even-odd) partitioning

Normalizing flows: training

Training: iterative procedure to minimize the **loss**

It must assure q to be as close as possible to the target p

Typical choice is the (reverse) **Kullback-Leibler divergence**

$$\tilde{D}_{\mathrm{KL}}(q\|p) = \int \mathrm{d}\phi\, q(\phi) \log \frac{q(\phi)}{p(\phi)} = -\langle \log \tilde{w}(\phi) \rangle_{\phi\sim q} + \log Z \geq 0$$

Measure of the "similarity" between two distributions

Define the weight

$$\tilde{w}(\phi) = p(\phi)/q(\phi)$$

Normalizing flows and the free energy

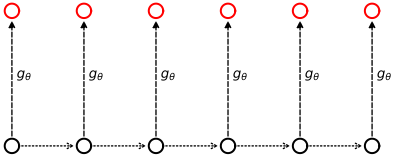
How do we use a trained flow g_θ and the distribution q?

How do we use a trained flow g_θ and the distribution q?

- **Reweighting**

$$\langle \mathcal{O} \rangle = \frac{1}{Z} \int \mathrm{d}\phi \, \mathcal{O}(\phi) q(\phi) \frac{p(\phi)}{q(\phi)} = \frac{1}{Z} \int \mathrm{d}\phi \, \underbrace{q(\phi)}_{\text{sample}} \underbrace{\mathcal{O}(\phi)\tilde{w}(\phi)}_{\text{measure}} = \frac{\langle \mathcal{O}(\phi)\tilde{w}(\phi) \rangle_{\phi \sim q}}{\langle \tilde{w}(\phi) \rangle_{\phi \sim q}}$$

- **Independent Metropolis-Hastings** → build a new Markov Chain from the output of the flow



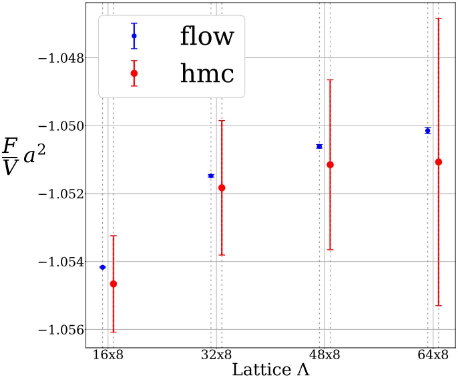
Normalizing flows provide an exact sampling procedure of p!

From the literature: the partition function

Get Z directly **[Nicoli et al.; 2007.07115]**

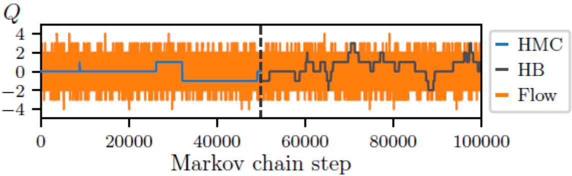
$$Z = \int \mathrm{d}\phi \, \exp(-S[\phi]) = \int \mathrm{d}\phi \, q(\phi)\tilde{w}(\phi) = \langle \tilde{w}(\phi) \rangle_{\phi \sim q}$$

$\rightarrow$ free-energy calculation in the 2D ϕ^4 scalar field theory



From the literature: topological unfreezing

History of the topological charge in $\mathrm{U}(1)$ gauge theory in 2D from **[Kanwar et al.; 2003.06413]**



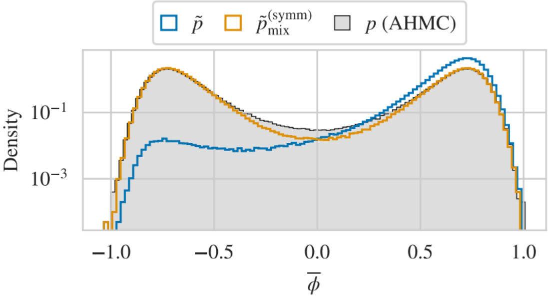
Topological freezing effectively disappears!

Theory is effectively trivialized

Some possible issues with NFs: multi-modal distributions

in the presence of multiple vacua the training procedure "picks" only one

"mode-collapse": only one mode of the distribution is sampled by the flow

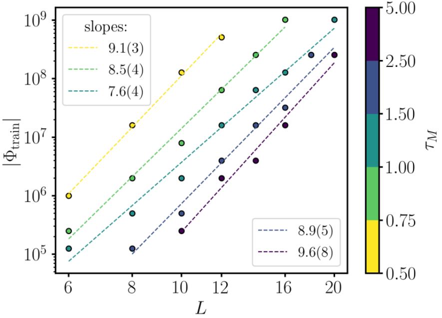


several solutions proposed in **[Hackett et al.; 2107.00734]** (see plot), **[Nicoli et al.; 2302.14082]**

Some possible issues with NFs: scalability

measurements of v.e.v. are statistically independent (no autocorrelation)

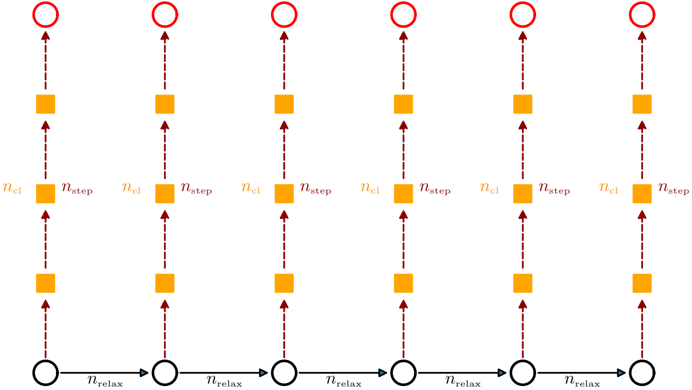
not clear however how the training times scale when approaching the continuum limit



comprehensive discussion in **[Del Debbio et al.; 2105.12481]** (see plot) and **[Abbott et al.; 2211.07541]**

Stochastic Normalizing Flows and Jarzynski's equality

Adding stochastic updates in the middle?



In between coupling layers we apply regular Monte Carlo updates with transition probabilities P_{η_n}

η_n is a **protocol** that interpolates the parameters of the theory between q_0 and p

We get SNFs $\rightarrow$ **[Wu et al.; 2002.06707] [Caselle et al.; 2201.08862]**

Jarzynski's equality

Free-energy differences (at equilibrium) directly calculated with an average over **non-equilibrium processes [Jarzynski; 1997]**:

$$\frac{Z}{Z_0} = \langle \exp\left(-W\right) \rangle_f$$

Along the process we compute the **work**

$$W = \sum_{n=0}^{N-1} \left\{ S_{\eta_{n+1}} \left[\phi_n\right] - S_{\eta_n} \left[\phi_n\right] \right\}$$

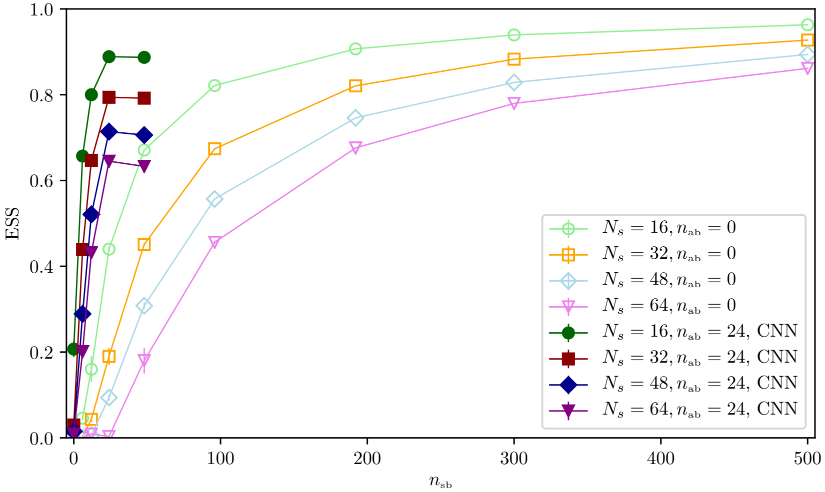
The proper KL divergence is a measure of reversibility

$$\tilde{D}_{\mathrm{KL}}(q_0 P_{\mathrm{f}} \| p P_{\mathrm{r}}) = \int \mathrm{d}\phi_0 \ldots \, q_0(\phi_0) P_{\mathrm{f}}[\phi_0 \to \phi] \ln \frac{q_0(\phi_0) P_{\mathrm{f}}[\phi_0 \to \phi]}{p(\phi) P_{\mathrm{r}}[\phi \to \phi_0]} = \underbrace{\langle W \rangle_{\mathrm{f}} - \Delta F \geq 0}_{\text{Second Law of thermodynamics!}}$$

JE is purely stochastic, but trainable coupling layers are easily accounted for including the Jacobian in the work and in the $\tilde{D}_{\mathrm{KL}}$

SNFs are a powerful common framework!

Training length: 10^4 epochs for all volumes. ESS $= \langle \tilde{w} \rangle_f^2 / \langle \tilde{w}^2 \rangle_f$ saturates fast



Conclusions

- Normalizing Flows are an extremely promising approach to mitigate critical slowing down in Lattice QCD
- Already capable of defeating or mitigating critical slowing down in low-dimensional theories
- Still, the scaling of training costs with the volume or for more complicated theories is challenging
- New ideas might be needed to actually build an efficient mapping to fine lattice spacings
- The stochastic nature of SNFs have the chance to improve the scaling of the training and provide insights on interpretability

Thank you for your attention!

DeepMind-MIT group NF notebook for ϕ^4 theory

Torino group SNF notebook for ϕ^4 theory

Continuous Normalizing Flows

Continuous NFs are built on Neural Ordinary Differential Equations (NODE) **[Chen et al.; 1806.07366]**

In CNFs g_θ is the solution of an ODE parameterized by a neural network V_θ:

$$\frac{d\phi(t)}{dt} = V_\theta(\phi(t), t)$$

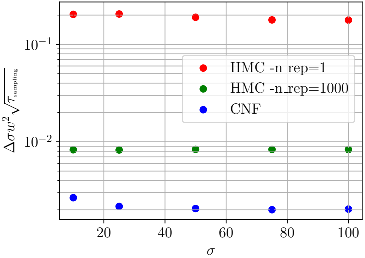
and solving it numerically gives the desired output

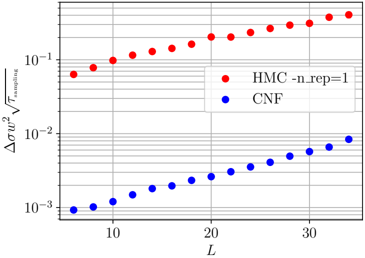
$$\phi(T) = \text{ODESOLVER}(V_\theta, \phi(0), [0, T])$$

The density of the generated samples can be computed through the ODE as well

$$\frac{d \log q_\theta(\phi(t))}{dt} = -(\nabla \cdot V_\theta)(\phi(t), t)$$

Impressive improvement over HMC in estimating the free energy





Out-of-equilibrium stochastic evolutions

Closer look at the average on the processes in the equality:

$$\frac{Z}{Z_0} = \langle \exp(-W) \rangle_f = \int \mathrm{d}\phi_0 \, \mathrm{d}\phi_1 \ldots \mathrm{d}\phi_N \, q_0(\phi_0) \, P_{\mathrm{f}}[\phi_0, \phi_1, \ldots, \phi_N] \, \exp(-W)$$

with

$$P_{\mathrm{f}}[\phi_0, \phi_1, \ldots, \phi_N] = \prod_{n=0}^{N-1} P_{\eta_n}(\phi_n \to \phi_{n+1})$$

- the *actual* probability distribution at each step is NOT the equilibrium distribution $\sim \exp(-S_{\eta_n})$: it's a non-equilibrium process!
- the $\langle \ldots \rangle_f$ average is taken over as many evolutions as possible (all independent from each other!)

for expectation values $\rightarrow$ reweighting-like formula

$$\langle \mathcal{O} \rangle = \frac{\langle \mathcal{O}(\phi_N) \exp(-W(\phi_0 \to \phi_N)) \rangle_{\mathrm{f}}}{\langle \exp(-W(\phi_0 \to \phi_N)) \rangle_{\mathrm{f}}}$$

A common framework: Stochastic Normalizing Flows

Jarzynski's relation is the same formula used to extract Z in NFs:

$$\frac{Z}{Z_0} = \langle \tilde{w}(\phi) \rangle_{\phi \sim q} = \langle \exp(-W) \rangle_{\mathrm{f}}$$

The "work" is simply

$$W(\phi_0, \dots, \phi_N) = S(\phi_N) - S_0(\phi_0) - Q(\phi_1, \dots, \phi_N) = -\ln \tilde{w}(\phi)$$

normalizing flows

$$\phi_0 \to \phi_1 = g_1(\phi_0) \to \cdots \to \phi$$

$$Q = \sum_{n=0}^{N-1} \ln |\det J_n(\phi_n)|$$

stochastic non-equilibrium evolutions

$$\phi_0 \overset{P_{\eta_1}}{\to} \phi_1 \overset{P_{\eta_2}}{\to} \dots \overset{P_{\eta_N}}{\to} \phi$$

$$Q = \sum_{n=0}^{N-1} S_{\eta_{n+1}}(\phi_{n+1}) - S_{\eta_{n+1}}(\phi_n)$$

Stochastic Normalizing Flows (introduced in **[Wu et al.; 2002.06707]**)

$$\phi_0 \to g_1(\phi_0) \overset{P_{\eta_1}}{\to} \phi_1 \to g_2(\phi_1) \overset{P_{\eta_2}}{\to} \dots \overset{P_{\eta_N}}{\to} \phi_N$$

$$Q = \sum_{n=0}^{N-1} S_{\eta_{n+1}}(\phi_{n+1}) - S_{\eta_{n+1}}(g_n(\phi_n)) + \ln |\det J_n(\phi_n)|$$

	normalizing flows	stochastic evolutions	SNFs
preparation	training	setting the protocol η_n	both
forward prob. P_f	$P_\mathrm{f} = \prod_n P_n(\phi_n \to \phi_{n+1})$		
transition prob. P_n	$\delta(\phi_{n+1} - g_n(\phi_n))$	$P_{\eta_n}(\phi_n \to \phi_{n+1})$	uses both
KL divergence	$\tilde{D}_\mathrm{KL}(q\|\|p)$	$\tilde{D}_\mathrm{KL}(q_0 P_\mathrm{f}\|\|p P_\mathrm{r})$	
"work"	$W = S - S_0 - Q = -\ln \tilde{w}$		
"heat" Q	$\sum_{n=0}^{N-1} \ln \lvert\det J_n(\phi_n)\rvert$	$\sum_{n=0}^{N-1} S_{\eta_{n+1}}(\phi_{n+1}) - S_{\eta_{n+1}}(\phi_n)$	both
e.v. $\langle \mathcal{O} \rangle$	$\frac{\langle \mathcal{O}(\phi_N)\tilde{w}(\phi_N)\rangle_{\phi_N \sim q}}{\langle \tilde{w}(\phi_N)\rangle_{\phi_N \sim q}}$	$\frac{\langle \mathcal{O}(\phi_N)\exp(-W(\phi_0 \to \phi_N))\rangle_\mathrm{f}}{\langle \exp(-W(\phi_0 \to \phi_N))\rangle_\mathrm{f}}$	

Testing SNFs

Goals

- can we train SNFs efficiently?
- can we improve both on NFs and on stochastic evolutions?
- how do the SNFs behave for a given neural network architecture?
- previous experience with stochastic evolutions with JE: the $\mathrm{SU}(3)$ equation of state in $(3+1)\mathrm{D}$ **[Caselle et al.; 2018]**. Can we learn something from it?

Using the Effective Sample Size as metric to evaluate architectures

$$\mathrm{ESS} = \frac{\langle \tilde{w} \rangle_{\mathrm{f}}^2}{\langle \tilde{w}^2 \rangle_{\mathrm{f}}}$$

$\mathrm{ESS} = 1 \rightarrow$ perfect training

Typical toy model for tests: ϕ^4 field theory in 2 dimensions

$$S(\phi) = \sum_{x \in \Lambda} -2\kappa \sum_{\mu=0,1} \phi(x)\phi(x+\hat{\mu}) + (1-2\lambda)\phi(x)^2 + \lambda\phi(x)^4$$

target parameters $\kappa = 0.2$ and $\lambda = 0.022$ (as in **[Nicoli et al.; 2020]**): unbroken symmetry phase

Protocol

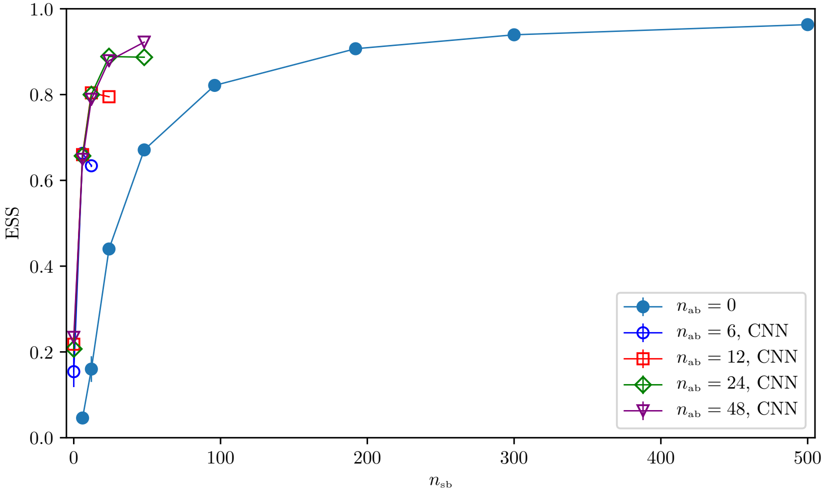
η_n interpolates between the prior (normal distribution is recovered with $\kappa = \lambda = 0$) and target parameters

- linear protocol η_n
- heatbath algorithm for the stochastic updates
- $n_{sb} = \#$ of stochastic updates

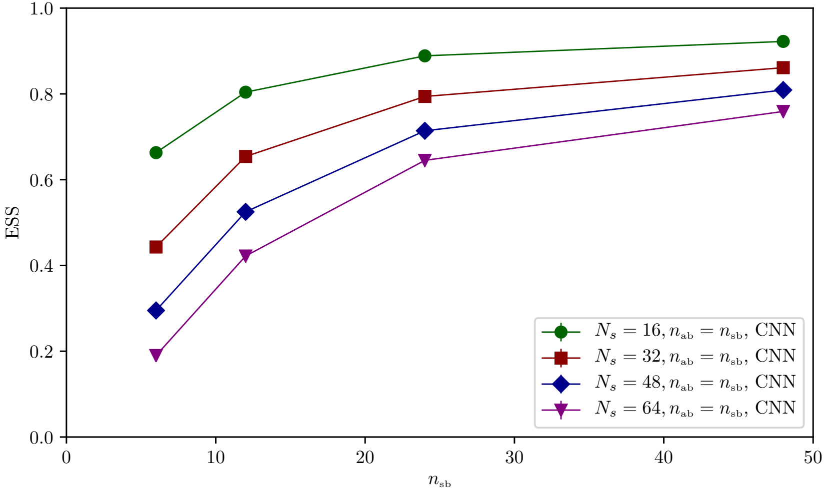
Coupling layers and NN

- $n_{ab} = \#$ of affine blocks
- inside each affine layer neural networks are CNNs with 1 hidden layer, 3×3 kernel and 1 feature map

Comparing stochastic evolutions with (S)NFs on a $N_s \times N_t = 16 \times 8$ lattice,



SNFs with $n_{sb} = n_{ab}$ as a possible recipe for efficient scaling



Some consideration on SNFs

The common framework between Jarzynski's equality and NFs is now explicit

General idea: use knowledge from non-equilibrium SM to create efficient SNFs

SNFs vs. stochastic evolutions

- Jarzynski's equality provides a way to compute Z and $\langle O \rangle$ (which works well also in LGTs, see $\mathrm{SU}(3)$ e.o.s. **[Caselle et al.; 2018]**)
- SNFs might be an even better method!
- trade-off: training for less MCMC updates
- very interesting for thermodynamic applications (or similar)

SNFs vs. normalizing flows

- improve scalability and interpretability?
- SNFs with CNNs and $n_{sb} = n_{ab}$ have a promising volume scaling at fixed training length
- training could be qualitatively "guided" towards the target by the protocol, but ultimately might also be limited by it

The Second Law of Thermodynamics

We start from Clausius inequality

$$\int_A^B \frac{\mathrm{d}Q}{T} \leq \Delta S$$

that for isothermal transformations becomes

$$\frac{Q}{T} \leq \Delta S$$

If we use

$$\begin{cases} Q = & \Delta E - W \quad \text{(First Law)} \\ F \stackrel{\text{def}}{=} & E - ST \end{cases}$$

the Second Law becomes

$$W \geq \Delta F$$

where the equality holds for reversible processes.

Moving from thermodynamics to statistical mechanics we know that the former relation (valid for a *macroscopic* system) becomes

$$\langle W \rangle_f \geq \Delta F$$

JE and the Second Law

Starting from Jarzynski's equality

$$\left\langle \exp\left(-\frac{W}{T}\right)\right\rangle_f = \exp\left(-\frac{\Delta F}{T}\right)$$

and using *Jensen's inequality*

$$\langle \exp x \rangle \geq \exp\langle x \rangle$$

(valid for averages on real x) we get

$$\exp\left(-\frac{\Delta F}{T}\right) = \left\langle \exp\left(-\frac{W}{T}\right)\right\rangle_f \geq \exp\left(-\frac{\langle W\rangle_f}{T}\right)$$

from which we have

$$\langle W\rangle_f \geq \Delta F$$

In this sense Jarzynski's relation can be seen as a **generalization** of the Second Law.