Gradient estimators without action derivative in Schwinger model.

Piotr Białas

Institute of Applied Computer Science
Faculty of Physics, Astronomy and Applied Computer Science
Jagiellonian University, Kraków, Poland

ECT* Workshop Machine Learning for lattice field theory and beyond

27 June 2023

with P. Korcyl and T. Stebel "Gradient estimators for normalising flows" arXiv:2202.01314

Outline

- Neural Markov Chain Monte-Carlo
- Gradient estimators
  - Reparametrization trick
  - REINFORCE
- Results
  - 2D Schwinger model

Independent Metropolised Sampler

$$p(\phi) = Z^{-1} e^{-\beta S(\phi)}, \qquad P(\phi) = Z \cdot p(\phi)$$

Jun S. Liu. "Metropolized independent sampling with comparisons to rejection sampling and importance sampling". Statistics and Computing 6.2 (1996), pp. 113–119.

Independent Metropolised Sampler

$$p(\phi) = Z^{-1} e^{-\beta S(\phi)}, \qquad P(\phi) = Z \cdot p(\phi)$$

$$\phi_{trial} \sim q(\cdot)$$

Jun S. Liu. "Metropolized independent sampling with comparisons to rejection sampling and importance sampling". Statistics and Computing 6.2 (1996), pp. 113–119.

Independent Metropolised Sampler

$$p(\phi) = Z^{-1} e^{-\beta S(\phi)}, \qquad P(\phi) = Z \cdot p(\phi)$$

$$\phi_{trial} \sim q(\cdot)$$

$$p_a(\phi_{trial}|\phi_i) = \min\left\{1, \frac{p(\phi_{trial})}{q(\phi_{trial})}\frac{q(\phi_i)}{p(\phi_i)}\right\}$$

Jun S. Liu. "Metropolized independent sampling with comparisons to rejection sampling and importance sampling". Statistics and Computing 6.2 (1996), pp. 113–119.

Learning $q(\phi)$

$$q(\phi) = q(\phi|\boldsymbol{\theta})$$

Learning $q(\phi)$

$$q(\phi) = q(\phi|\boldsymbol{\theta})$$

$$\underset{\theta}{\operatorname{argmin}} \operatorname{dist}(q(\cdot|\boldsymbol{\theta})|p)$$

Kullback-Leibler divergence

$$D_{KL}(q(\cdot|\boldsymbol{\theta})|p) = \int d\phi \, q(\phi|\boldsymbol{\theta}) \log \frac{q(\phi|\boldsymbol{\theta})}{p(\phi)}$$

Kullback-Leibler divergence

$$D_{KL}(q(\cdot|\boldsymbol{\theta})|p) = \int d\phi \, q(\phi|\boldsymbol{\theta}) \log \frac{q(\phi|\boldsymbol{\theta})}{p(\phi)}$$

$$D_{KL}(q(\cdot|\boldsymbol{\theta})|p) \geq 0, \qquad D_{KL}(q_{\boldsymbol{\theta}}|p) = 0 \iff p = q$$

Kullback-Leibler divergence

$$D_{KL}(q(\cdot|\boldsymbol{\theta})|p) = \int d\phi \, q(\phi|\boldsymbol{\theta}) \log \frac{q(\phi|\boldsymbol{\theta})}{p(\phi)}$$

$$D_{KL}(q(\cdot|\boldsymbol{\theta})|p) \geq 0, \qquad D_{KL}(q_{\boldsymbol{\theta}}|p) = 0 \iff p = q$$

$$D_{KL}(q(\cdot|\boldsymbol{\theta})|p) \neq D_{KL}(p|q(\cdot|\boldsymbol{\theta}))$$

Normalising flows

$$\mathbb{R}^D \ni \boldsymbol{z} \longrightarrow (q_{pr}(\boldsymbol{z}),\ \underset{\substack{\uparrow \\ \text{bijection}}}{\boldsymbol{\varphi}(\boldsymbol{z}|\boldsymbol{\theta})}\ ) \in (\mathbb{R}, \mathbb{R}^D)$$

$$\boldsymbol{\phi} = \boldsymbol{\varphi}(\boldsymbol{z}|\boldsymbol{\theta}), \qquad q(\boldsymbol{\phi}|\boldsymbol{\theta}) \equiv q_z(\boldsymbol{z}|\boldsymbol{\theta}) = q_{pr}(z) J(\boldsymbol{z}|\boldsymbol{\theta})^{-1}$$

$$J(\boldsymbol{z}|\boldsymbol{\theta}) = \det\left(\frac{\partial \boldsymbol{\varphi}(\boldsymbol{z}|\boldsymbol{\theta})}{\partial \boldsymbol{z}}\right)$$

Ivan Kobyzev, Simon Prince, and Marcus Brubaker. “Normalizing Flows: An Introduction and Review of Current Methods”. IEEE Transactions on Pattern Analysis and Machine Intelligence(2020), pp. 1.
M. S. Albergo, G. Kanwar, and P. E. Shanahan. “Flow-based generative models for Markov chain Monte Carlo in lattice field theory”. Phys. Rev. D100 (2019), p. 034515.
Michael S. Albergo et al. ”Introduction to Normalizing Flows for Lattice Field Theory” (2021) arXiv:2101.08176

Reparametrization trick

$$D_{KL}(q_{\boldsymbol{\theta}}|p) = \int d\phi\, q(\phi|\boldsymbol{\theta}) \left(\log q(\phi|\boldsymbol{\theta}) - \log(p(\phi))\right)$$

Diederik P. Kingma and Max Welling, "Auto-Encoding Variational Bayes" (2013) arXiv:1312.6114v11

Reparametrization trick

$$D_{KL}(q_{\boldsymbol{\theta}}|p) = \int d\phi\, q(\phi|\boldsymbol{\theta})\left(\log q(\phi|\boldsymbol{\theta}) - \log(p(\phi))\right)$$

$$D_{KL}(q|p) = \int d\boldsymbol{z}\, q_{pr}(\boldsymbol{z})\left(\log q_z(\boldsymbol{z}|\boldsymbol{\theta}) - \log p(\varphi(\boldsymbol{z}|\boldsymbol{\theta}))\right)$$

Diederik P. Kingma and Max Welling, "Auto-Encoding Variational Bayes" (2013) arXiv:1312.6114v11

Reparametrization trick - gradient

$$\frac{dD_{KL}(q|p)}{d\boldsymbol{\theta}} = \int d\boldsymbol{z}\, q_{pr}(\boldsymbol{z}) \frac{d}{d\boldsymbol{\theta}} \left(\log q(\boldsymbol{z}|\boldsymbol{\theta}) - \log p(\varphi(\boldsymbol{z}|\boldsymbol{\theta}))\right)$$

Reparametrization trick - gradient

$$\frac{dD_{KL}(q|p)}{d\boldsymbol{\theta}} = \int d\boldsymbol{z}\, q_{pr}(\boldsymbol{z}) \frac{d}{d\boldsymbol{\theta}} \left(\log q(\boldsymbol{z}|\boldsymbol{\theta}) - \log p(\varphi(\boldsymbol{z}|\boldsymbol{\theta}))\right)$$

$$\begin{aligned}\frac{dD_{KL}(q|p)}{d\boldsymbol{\theta}} &\approx \mathbf{g}_{rt}[\{\phi\}] \\ &\equiv \frac{d}{d\boldsymbol{\theta}} \frac{1}{N} \sum_{i=1}^{N} \Big(\log q_z(\boldsymbol{z}_i|\boldsymbol{\theta}) - \boxed{\log p(\varphi(\boldsymbol{z}_i|\boldsymbol{\theta}))}\Big)\end{aligned}$$

$$\boldsymbol{z}_i \sim q_{pr}(\cdot)$$

Action derivative

$$\frac{d}{d\boldsymbol{\theta}} \log p(\varphi(\boldsymbol{z}_i|\boldsymbol{\theta})) = \boxed{\left.\frac{d}{d\phi} \log p(\phi)\right|_{\phi=\varphi(\boldsymbol{z}_i|\boldsymbol{\theta})}} \frac{d}{d\boldsymbol{\theta}} \varphi(\boldsymbol{z}_i|\boldsymbol{\theta})$$

$$\frac{d}{d\phi} \log p(\phi) = -\frac{d}{d\phi} S(\phi)$$

D_{KL} gradient

$$\frac{dD_{KL}(q|p)}{d\boldsymbol{\theta}} = \int d\phi \, \frac{\partial q(\boldsymbol{\phi}|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \left(\log q(\phi|\theta) - \log p(\phi)\right) + \int d\phi \, q(\phi|\boldsymbol{\theta}) \frac{\partial}{\partial \boldsymbol{\theta}} \log q(\phi|\theta)$$

A. Mnih, D. J. Rezende, "Variational Inference for Monte Carlo Objectives" arXiv:1502.06725
Dian Wu, Lei Wang, and Pan Zhang. "Solving Statistical Mechanics Using Variational Autoregressive Networks". Phys. Rev. Lett.122 (2019), p. 080602.

D_{KL} gradient

$$\frac{dD_{KL}(q|p)}{d\boldsymbol{\theta}} = \int d\phi \, \frac{\partial q(\phi|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \left(\log q(\phi|\theta) - \log p(\phi)\right) + \int d\phi \, q(\phi|\boldsymbol{\theta}) \frac{\partial}{\partial \boldsymbol{\theta}} \log q(\phi|\theta)$$

$$\int d\phi \, \frac{\partial q(\phi|\theta)}{\partial \boldsymbol{\theta}} = \frac{\partial}{\partial \boldsymbol{\theta}} \underbrace{\int d\phi \, q(\phi|\theta)}_{1} = 0$$

A. Mnih, D. J. Rezende, "Variational Inference for Monte Carlo Objectives" arXiv:1502.06725
Dian Wu, Lei Wang, and Pan Zhang. "Solving Statistical Mechanics Using Variational Autoregressive Networks". Phys. Rev. Lett.122 (2019), p. 080602.

REINFORCE

$$\frac{dD_{KL}(q|p)}{d\boldsymbol{\theta}} = \int d\phi\, q(\phi|\boldsymbol{\theta}) \frac{\partial \log q(\phi|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \left(\log q(\phi|\theta) - \log p(\phi)\right)$$

A. Mnih, D. J. Rezende, "Variational Inference for Monte Carlo Objectives" arXiv:1502.06725
Dian Wu, Lei Wang, and Pan Zhang. "Solving Statistical Mechanics Using Variational Autoregressive Networks". Phys. Rev. Lett.122 (2019), p. 080602.

REINFORCE

$$\frac{dD_{KL}(q|p)}{d\boldsymbol{\theta}} = \int d\phi\, q(\phi|\boldsymbol{\theta})\frac{\partial \log q(\phi|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\left(\log q(\phi|\theta) - \log p(\phi)\right)$$

$$\frac{dD_{KL}(q|p)}{d\boldsymbol{\theta}} \approx$$

$$\mathbf{g}_{re}[\{\phi\}] \equiv \frac{1}{N}\sum_{i=1}^{N} \frac{\partial \log q(\phi_i|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\left(\log q(\phi_i|\boldsymbol{\theta}) - \log p(\phi_i)\right)$$

$$\phi \sim q(\phi|\boldsymbol{\theta})$$

A. Mnih, D. J. Rezende, "Variational Inference for Monte Carlo Objectives" arXiv:1502.06725
Dian Wu, Lei Wang, and Pan Zhang. "Solving Statistical Mechanics Using Variational Autoregressive Networks". Phys. Rev. Lett.122 (2019), p. 080602.

REINFORCE

$$\mathbf{g}_{re}[\{\phi\}] = \frac{1}{N} \sum_{i=1}^{N} \frac{\partial \log q(\phi_i|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \underbrace{(\log q(\phi_i|\boldsymbol{\theta}) - \log p(\phi_i))}_{s_i}$$

$$\phi \sim q(\phi|\boldsymbol{\theta})$$

A. Mnih, D. J. Rezende, "Variational Inference for Monte Carlo Objectives" arXiv:1502.06725
Dian Wu, Lei Wang, and Pan Zhang. "Solving Statistical Mechanics Using Variational Autoregressive Networks". Phys. Rev. Lett.122 (2019), p. 080602.

REINFORCE

$$\mathbf{g}_{re}[\{\phi\}] = \frac{1}{N}\sum_{i=1}^{N} \frac{\partial \log q(\phi_i|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \underbrace{\left(\log q(\phi_i|\boldsymbol{\theta}) - \log p(\phi_i)\right)}_{s_i}$$

$$\phi \sim q(\phi|\boldsymbol{\theta})$$

$$\mathbf{g}_{\bar{re}}[\{\phi\}] = \frac{1}{N}\sum_{i=1}^{N} \frac{\partial \log q(\phi_i|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \left(s_i - \bar{s}\right)$$

$$\bar{s} = \frac{1}{N}\sum_{i=1}^{N} s_i$$

A. Mnih, D. J. Rezende, "Variational Inference for Monte Carlo Objectives" arXiv:1502.06725
Dian Wu, Lei Wang, and Pan Zhang. "Solving Statistical Mechanics Using Variational Autoregressive Networks". Phys. Rev. Lett.122 (2019), p. 080602.

REINFORCE

$$E\left[\mathbf{g}_{\bar{re}}[\{\phi\}]\right] = \frac{N-1}{N} E\left[\mathbf{g}_{re}[\{\phi\}]\right].$$

P. Białas, P. Korcyl, T. Stebel, "Gradient estimators for normalising flows", arXiv:2202.01314.

Implementing REINFORCE

$$\frac{\partial \log q(\phi_i|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$$

P. Białas, P. Korcyl, T. Stebel, "Gradient estimators for normalising flows", arXiv:2202.01314.
L. Vaitl, K. A. Nicoli, S. Nakajima, P. Kessel "Gradients should stay on Path: Better Estimators of the Reverse- and Forward KL Divergence for Normalizing Flows" Machine Learning: Science and Technology **3** (2022) 045006.

Implementing REINFORCE

$$\frac{\partial \log q(\phi_i|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$$

$$q(\phi|\boldsymbol{\theta}) = q_{pr}(\boldsymbol{\varphi}^{-1}(\phi|\boldsymbol{\theta}))\bar{J}(\phi|\boldsymbol{\theta})$$

$$\bar{J}(\phi|\boldsymbol{\theta}) \equiv \det\left(\frac{\partial \boldsymbol{\varphi}^{-1}(\phi|\boldsymbol{\theta})}{\partial \phi}\right)$$

P. Białas, P. Korcyl, T. Stebel, "Gradient estimators for normalising flows", arXiv:2202.01314.
L. Vaitl, K. A. Nicoli, S. Nakajima, P. Kessel "Gradients should stay on Path: Better Estimators of the Reverse- and Forward KL Divergence for Normalizing Flows" Machine Learning: Science and Technology **3** (2022) 045006.

Implementing REINFORCE

No gradient calculations

$$z_i \sim q_{pr}$$

Implementing REINFORCE

No gradient calculations

$$z_i \sim q_{pr}$$

$$\phi = q(z|\theta)$$

Implementing REINFORCE

No gradient calculations

$$z_i \sim q_{pr}$$

$$\phi = q(z|\theta)$$

Gradient calculations

$$\boldsymbol{z}_i' = \boldsymbol{\varphi}^{-1}(\phi_i|\boldsymbol{\theta})$$

$$\bar{J}(\phi|\boldsymbol{\theta}) \equiv \det\left(\frac{\partial\boldsymbol{\varphi}^{-1}(\boldsymbol{\phi}|\boldsymbol{\theta})}{\partial\boldsymbol{\phi}}\right)$$

Implementing REINFORCE

No gradient calculations

$$z_i \sim q_{pr}$$

$$\phi = q(z|\theta)$$

Gradient calculations

$$\boldsymbol{z}_i' = \boldsymbol{\varphi}^{-1}(\phi_i|\boldsymbol{\theta})$$

$$\bar{J}(\phi|\boldsymbol{\theta}) \equiv \det\left(\frac{\partial \boldsymbol{\varphi}^{-1}(\phi|\boldsymbol{\theta})}{\partial \phi}\right)$$

$$q(\phi|\boldsymbol{\theta}) = q_{pr}(\boldsymbol{z}')\bar{J}(\phi|\boldsymbol{\theta})$$

Implementing REINFORCE

No gradient calculations

$$z_i \sim q_{pr}$$

$$\phi = q(z|\theta)$$

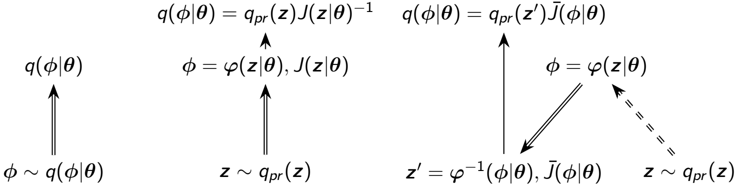
Gradient calculations

$$\boldsymbol{z}'_i = \boldsymbol{\varphi}^{-1}(\phi_i|\boldsymbol{\theta})$$

$$\bar{J}(\phi|\boldsymbol{\theta}) \equiv \det\left(\frac{\partial \boldsymbol{\varphi}^{-1}(\boldsymbol{\phi}|\boldsymbol{\theta})}{\partial \boldsymbol{\phi}}\right)$$

$$q(\phi|\boldsymbol{\theta}) = q_{pr}(\boldsymbol{z}')\bar{J}(\phi|\boldsymbol{\theta})$$

Implementing REINFORCE



Implementation - reparameterization

```python
def grt_loss(z, log_prob_z, *, model, action, use_amp):
    layers = model['layers']

    with autocast(enabled=use_amp):
        x, logq = nf.apply_flow(layers, z, log_prob_z)

        logp = -action(x)
        loss = nf.calc_dkl(logp, logq)

        return loss, logq.detach(), logp.detach()
```

Implementation - REINFORCE

```python
def gri_loss(z_a, log_prob_z_a, *,
             model, action, use_amp):
    layers, prior = model['layers'], model['prior']
    with torch.no_grad():
        with autocast(enabled=use_amp):
            phi, logq = nf.apply_flow(layers, z_a, log_prob_z_a)
            logp = -action(phi)
            signal = logq - logp

    with autocast(enabled=use_amp):
        z, log_q_phi = nf.reverse_apply_flow(layers, phi)
        prob_z = prior.log_prob(z)
        log_q_phi = prob_z - log_q_phi
        loss = torch.mean(log_q_phi * (signal - signal.mean()))
```

Schwinger model

$$S(U) = -\beta \sum_x \operatorname{Re} P(x) - \log\det D[U]^\dagger D[U].$$

$$\begin{aligned} D[U](y,x)^{\alpha\beta} =& \delta(y-x)\delta^{\alpha\beta} \\ & - \kappa \sum_{\mu=0,1} \Big\{ [1-\sigma^\mu]^{\beta\alpha}\delta(y-x+\hat{\mu}) \\ & \qquad + [1+\sigma^\mu]^{\beta\alpha}\delta(y-x-\hat{\mu}) \Big\} \end{aligned}$$

Schwinger model - Implementation

- Gauge equivariant layers with circular neural spline flows (8-knots)
- Fermionic determinant calculated explicitely
- $\beta = 2$ $\kappa = 0.276$

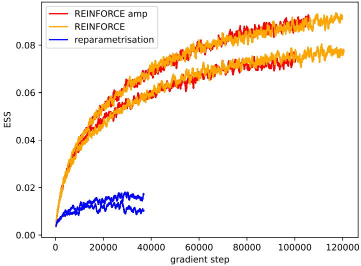
M. S. Albergo, D. Boyda, K. Cranmer, D. C. Hackett, G. Kanwar, S. Racaniere, D. J. Rezende, F. Romero-Lopez, P. E. Shanahan, J. M. Urban, ´ Flow-based sampling in the lattice schwinger model at criticality, Phys. Rev. D 106 (2022) 014514.
G. Kanwar, M. S. Albergo, D. Boyda, K. Cranmer, D. C. Hackett, S. Racaniere, D. J. Rezende, P. E. Shanahan, Equivariant flow-based ' sampling for lattice gauge theory, Phys. Rev. Lett. 125 (2020) 121601.
D. J. Rezende, G. Papamakarios, S. Racanière, M. S. Albergo, G. Kanwar, P. E. Shanahan, K. Cranmer, "Normalizing Flows on Tori and Spheres" arXiv:2002.02428
M.'S. Albergo, D. Boyda and D. C. Hackett, G. Kanwar, K. Cranmer,S. Racanière and D. J. Rezende, P. E. Shanahan, "Introduction to Normalizing Flows for Lattice Field Theory" arXiv:2101.08176.

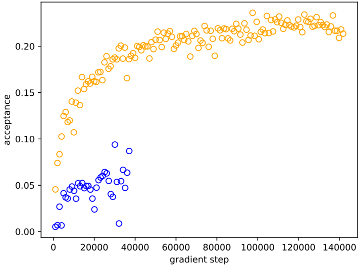
Results – Effective Sample Size

$$w(\phi) = \frac{p(\phi)}{q(\phi|\boldsymbol{\theta})}$$

$$ESS = \frac{\langle w \rangle_q^2}{\langle w^2 \rangle_q}$$

Results - 16 × 16 - 512 × 512 Dirac operator



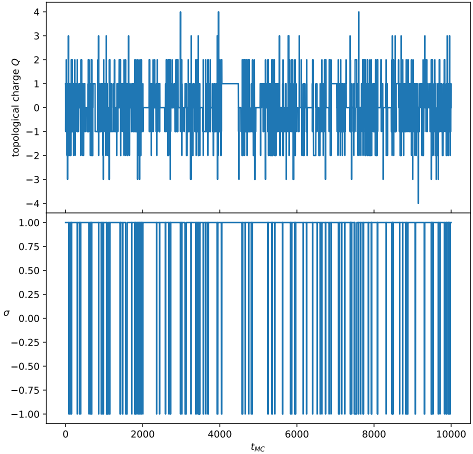


Results - Topological charge

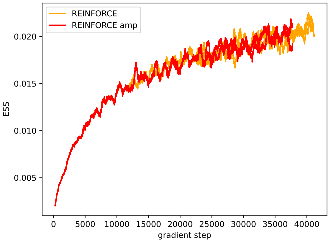
$$Q = \frac{1}{2\pi} \sum_x \operatorname{Im} \log P(x)$$

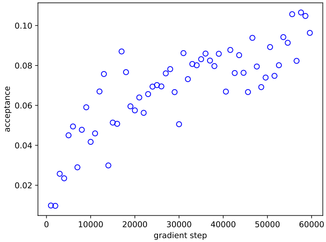
$$\sigma = \operatorname{sign}(\operatorname{Re} \det D)$$

Results - Topological charge - 16×16

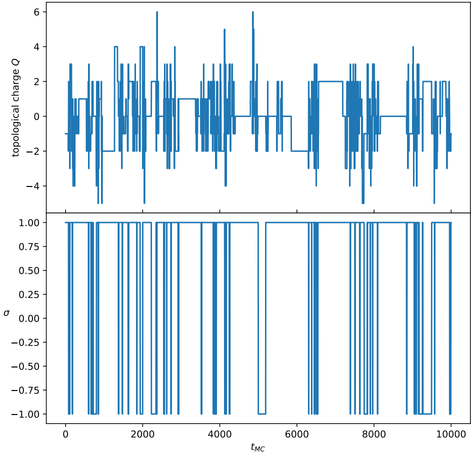


Results - 24 $\times$ 24 - 1152 $\times$ 1152 Dirac operator

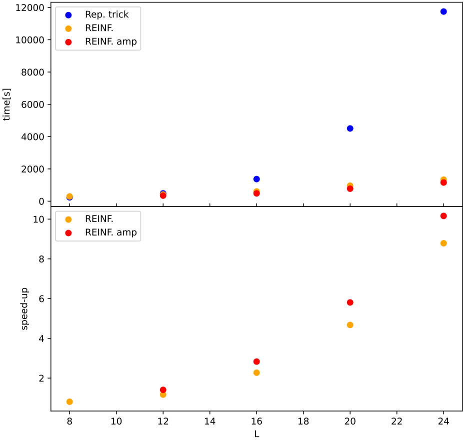




Results - Topological charge 24×24



Timings - Tesla V100-SXM2-32GB



Time per 100 gradient update steps (batch size = 1536)

Memory used for gradient calculations

L	batch	r.t.		REINF.	
		M1 [GB]	L1	M1 [GB]	L1
16	128	3.09	5296	2.6	3401
16	256	6.16	5292	5.19	3401
24	128	7.88	7847	5.37	3401
24	256	15.72	7827	10.74	3401

M1 - memory used for storing buffers in flow graphs for gradient calculations
L1 - number of different buffers allocated in flow graphs for gradient calculations

Conclusions

- REINFORCE gradient estimator avoids action derivative.
- It can be easily implemented for reversible normalising flows.
- For complicated actions like *e.g.* Schwinger model it offers
  - substantial time and memory savings.
  - better numerical accuracy.

Thank you :)

Thank you :)

Computer time allocation grant plgng on the Ares and plgnglft on Athena supercomputers hosted by AGH Cyfronet in Kraków, Poland was used through the polish PLGRID consortium.
T.S. kindly acknowledges support of the Polish National Science Center (NCN) Grant No. 2019/32/C/ST2/00202 and support of the Faculty of Physics, Astronomy and Applied Computer Science, Jagiellonian University Grant No. 2021-N17/MNS/000062.
This research was partially funded by the Priority Research Area Digiworld under the program Excellence Initiative – Research University at the Jagiellonian University in Kraków.