Title: Reducing the Sign Problem with Complex Neural Networks

Speakers: Johann Ostmeyer

Collection: Quantum Criticality: Gauge Fields and Matter

Date: May 16, 2022 - 3:00 PM

URL: https://pirsa.org/22050030

Abstract: The sign problem is arguably the greatest weakness of the otherwise highly efficient, non-perturbative Monte Carlo simulations. Recently, considerable progress has been made in alleviating the sign problem by deforming the integration contour of the path integral into the complex plane and applying machine learning to find near-optimal alternative contours. This deformation however requires a Jacobian determinant calculation which has a generic computational cost scaling as volume cubed. In this talk I am going to present a new architecture with linear runtime, based on complex-valued affine coupling layers.

Reducing the Sign Problem with Complex Neural Networks

based on arXiv:2203.00390 by
Marcel Rodekamp, Evan Berkowitz, Christoph Gäntgen, Stefan Krieg, Tom Luu, JO

Johann Ostmeyer

Department of Mathematical Sciences, University of Liverpool, United Kingdom

May 16, 2022

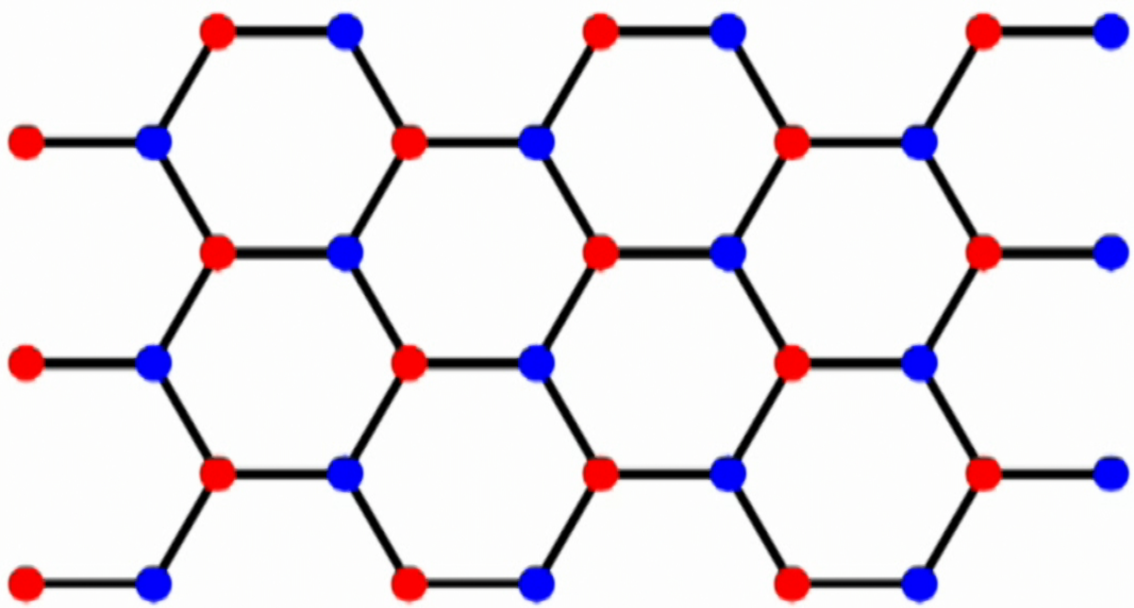
UNIVERSITY OF LIVERPOOL

LIVERPOOL THEORETICAL PHYSICS

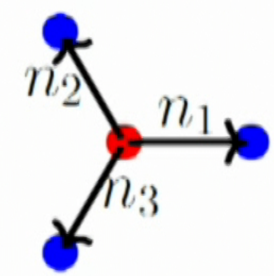
HMC simulations of the Honeycomb Hubbard model at Half Filling

[Ostmeyer *et al.* 2020, 2021]

$$H = -\sum_{\langle x,y\rangle ,s} c^\dagger_{x,s} c_{y,s} + \frac{1}{2} U \sum_x q_x^2$$



Sub-lattices **A** and **B**



Johann Ostmeyer

HMC simulations of the Honeycomb Hubbard model at Half Filling

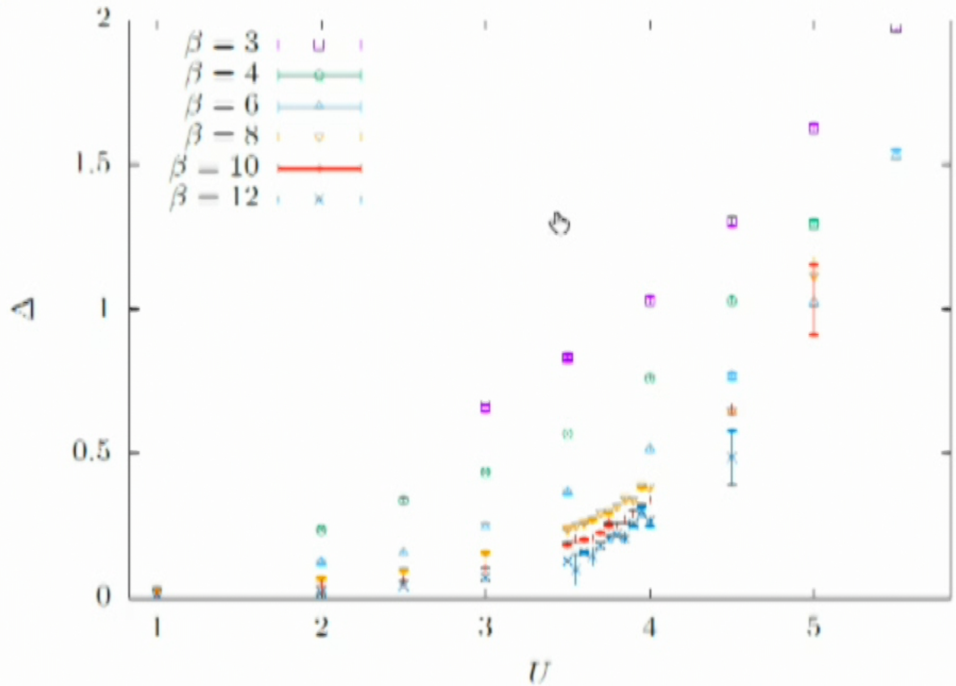
[Ostmeyer *et al.* 2020, 2021]

$$H = -\sum_{\langle x,y\rangle,s} c^\dagger_{x,s} c_{y,s} + \frac{1}{2} U \sum_x q_x^2$$

$\Delta \sim \beta^{-1}$

$\Delta \sim (U - U_c)^\nu$

$\Delta = \beta^{-1} F(\beta^{1/\nu}(U - U_c))$



HMC simulations of the Honeycomb Hubbard model at Half Filling

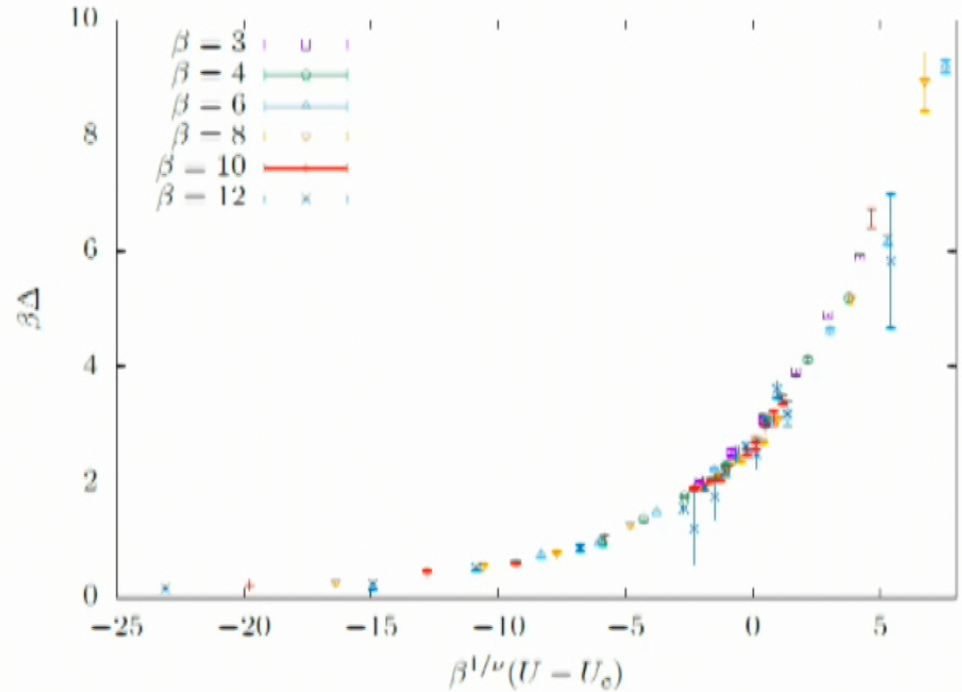
[Ostmeyer *et al.* 2020, 2021]

$$H = -\sum_{\langle x,y\rangle,s} c^\dagger_{x,s} c_{y,s} + \frac{1}{2} U \sum_x q_x^2$$

$\Delta \sim \beta^{-1}$

$\Delta \sim (U - U_c)^\nu$

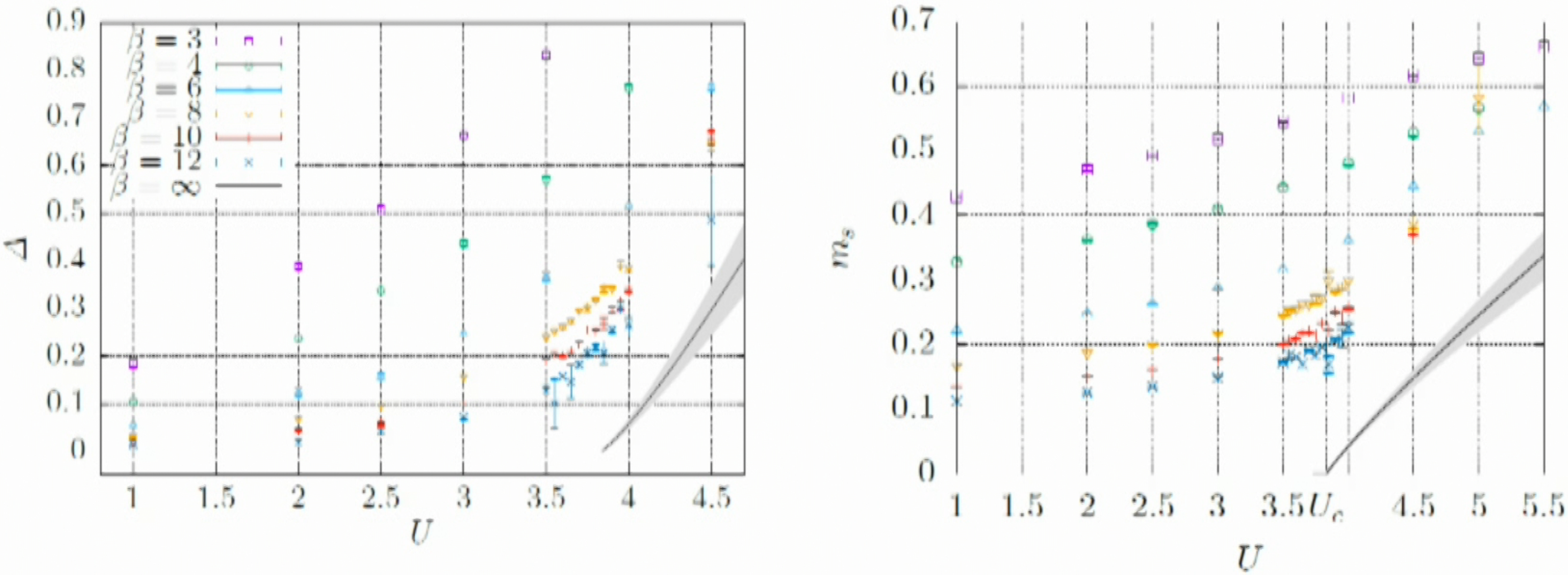
$\Delta = \beta^{-1} F(\beta^{1/\nu}(U - U_c))$



HMC simulations of the Honeycomb Hubbard model at Half Filling

[Ostmeyer *et al.* 2020, 2021]

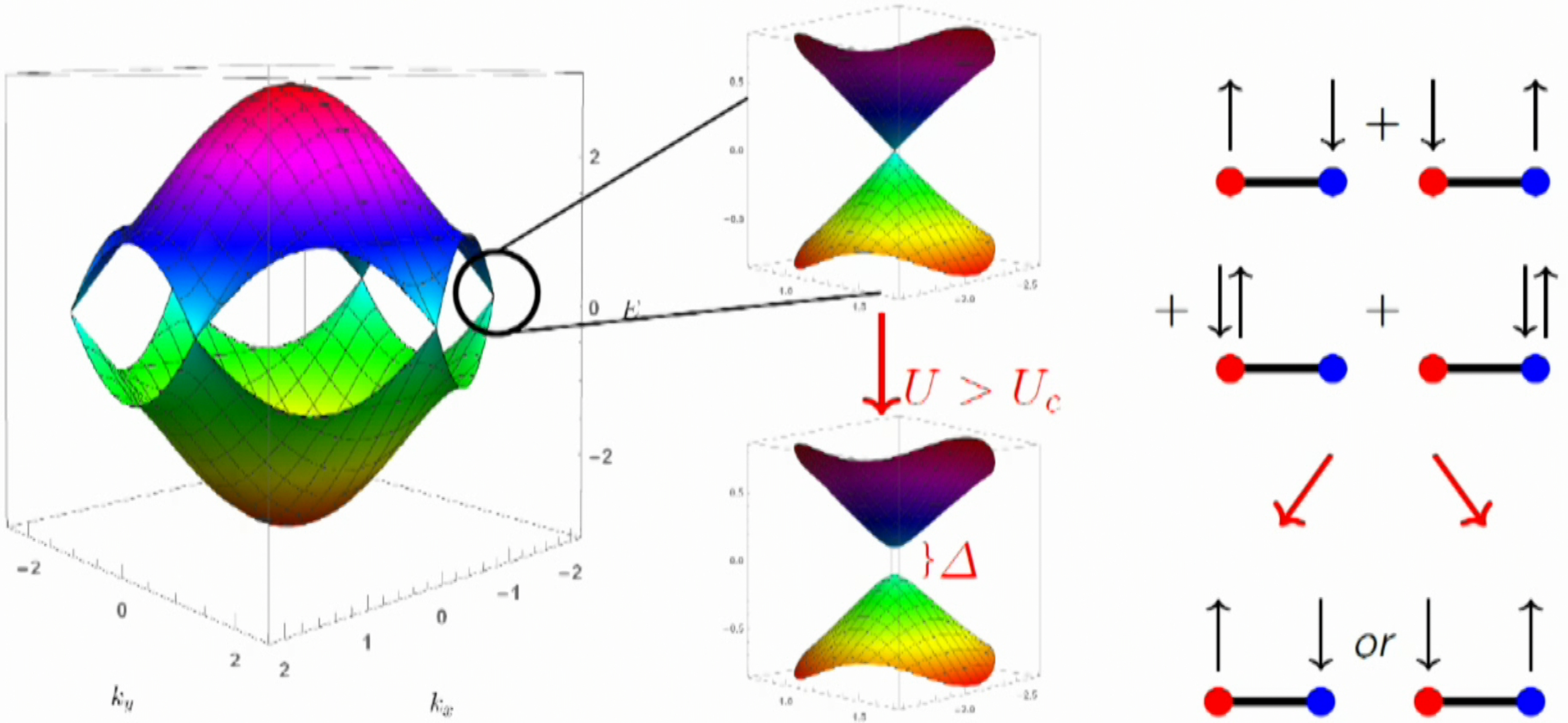
$$H = -\sum_{\langle x,y\rangle,s} c^\dagger_{x,s} c_{y,s} + \frac{1}{2} U \sum_x q_x^2$$



Johann Ostmeyer 2/20

HMC simulations of the Honeycomb Hubbard model at Half Filling

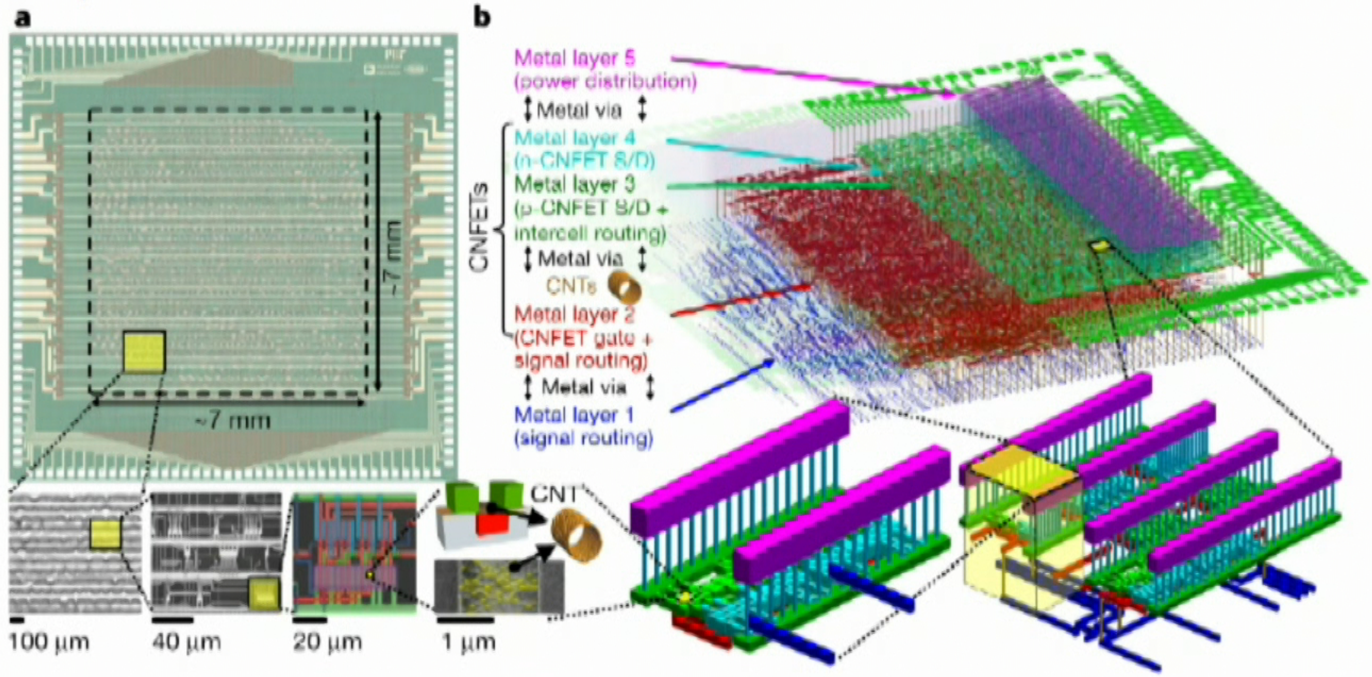
[Ostmeyer *et al.* 2020, 2021]



Johann Ostmeyer

2/20

Graphene computer

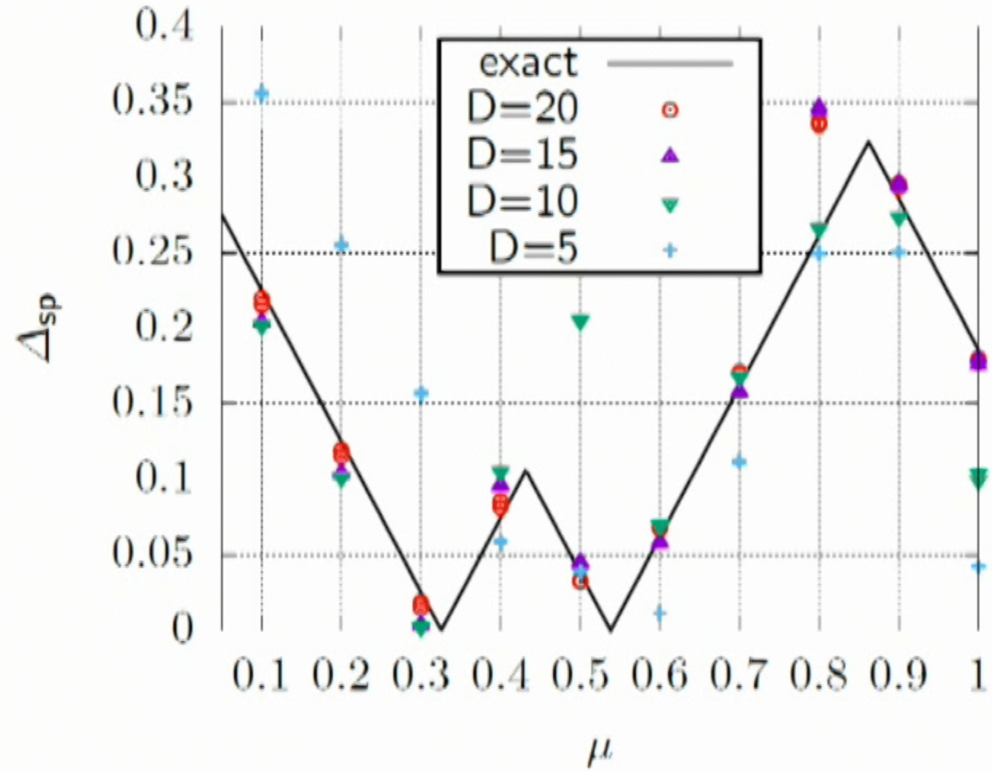


"Hello, world!
I am RV16XNano."

[Hills *et al.* 2019]

Beyond half filling?
Sign problem!

- Reweighting: $\langle \mathcal{O} \rangle = \frac{\langle \mathcal{O} e^{i\phi} \rangle}{\langle e^{i\phi} \rangle}$
- Tensor Networks
  [Corboz 2016; Schneider *et al.* 2021]



Johann Ostmeyer

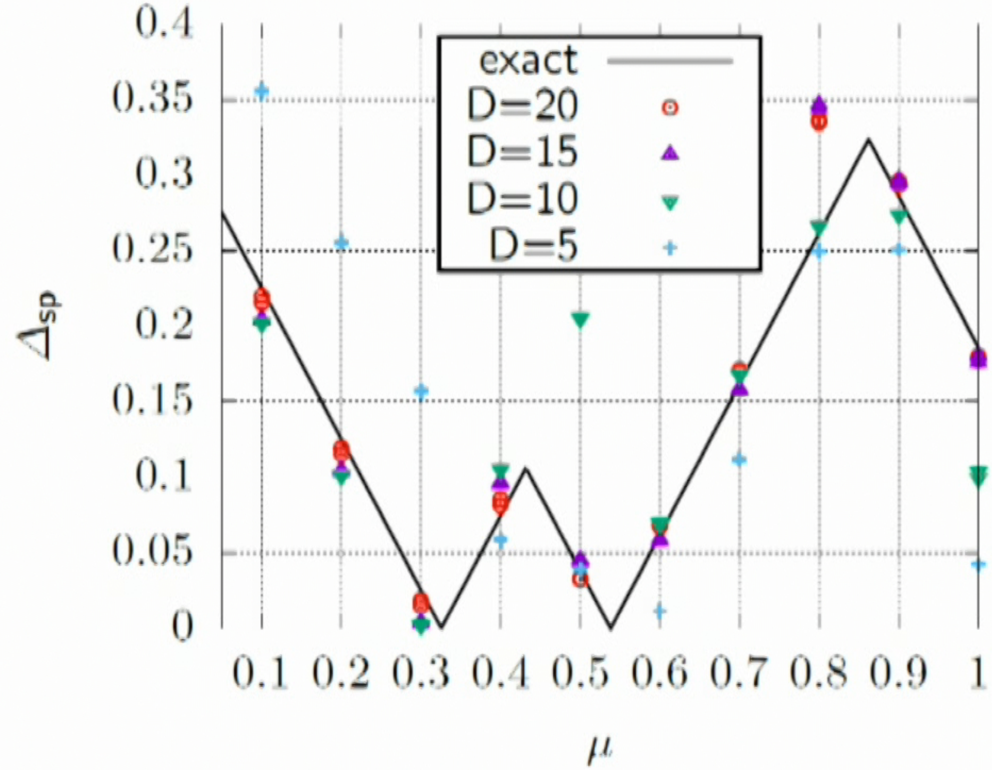
Beyond half filling?
Sign problem!

- Reweighting: $\langle \mathcal{O} \rangle = \frac{\langle \mathcal{O} e^{i\phi} \rangle}{\langle e^{i\phi} \rangle}$
- Tensor Networks
  [Corboz 2016; Schneider *et al.* 2021]
- Lefschetz thimbles & holomorphic flow
  [Alexandru *et al.* 2016; Cristoforetti *et al.* 2013; Ulybyshev *et al.* 2020; Wynen *et al.* 2021]

  [Rodekamp *et al.* 2022]

Beyond half filling?
Sign problem!

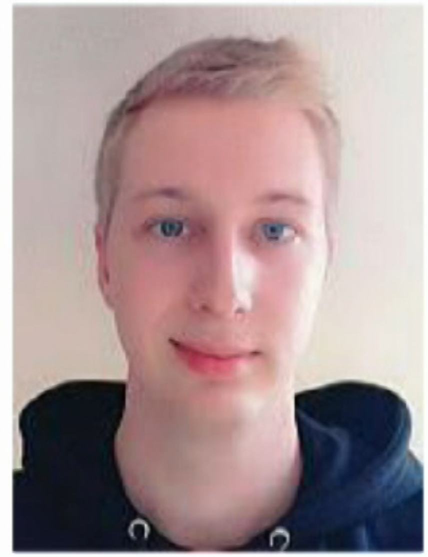
- Reweighting: $\langle \mathcal{O} \rangle = \frac{\langle \mathcal{O} e^{i\phi} \rangle}{\langle e^{i\phi} \rangle}$
- Tensor Networks
  [Corboz 2016; Schneider *et al.* 2021]



Johann Ostmeyer 4/20

Credits

"Mitigating the Hubbard Sign Problem with Complex-Valued Neural Networks"
Marcel Rodekamp, Evan Berkowitz, Christoph Gäntgen, Stefan Krieg, Thomas Luu, Johann Ostmeyer



Hubbard model [Hubbard 1963]

$$H = -\sum_{\langle x,y\rangle ,s} c^\dagger_{x,s} c_{y,s} + \frac{1}{2} U \sum_x q_x^2 + \mu \sum_x q_x$$

$$q_x = n_{x,\uparrow} + n_{x,\downarrow} - 1$$

$$n_{x,s} = c^\dagger_{x,s} c_{x,s}$$

- U is a repulsive coupling
- μ is chemical potential $\rightarrow$ sign problem for $\mu \neq 0$

Path integral observables

- Observable: $\langle \mathcal{O} \rangle = \frac{1}{\mathcal{Z}} \int \mathcal{D}\Phi \; e^{-S[\Phi]} \mathcal{O}[\Phi] \equiv \int \mathcal{D}\Phi \; p_S[\Phi] \, \mathcal{O}[\Phi]$
- Probability density: $p_S[\Phi] \overset{!}{\geq} 0$
- Sign Problem: $S[\Phi] \notin \mathbb{R} \Rightarrow p_S[\Phi] \ngeq 0$

Path integral observables

- Observable: $\langle \mathcal{O} \rangle = \frac{1}{Z} \int \mathcal{D}\Phi \, e^{-S[\Phi]} \mathcal{O}[\Phi] \equiv \int \mathcal{D}\Phi \, p_S[\Phi] \, \mathcal{O}[\Phi]$
- Probability density: $p_S[\Phi] \overset{!}{\geq} 0$
- Sign Problem: $S[\Phi] \notin \mathbb{R} \Rightarrow p_S[\Phi] \ngeq 0$

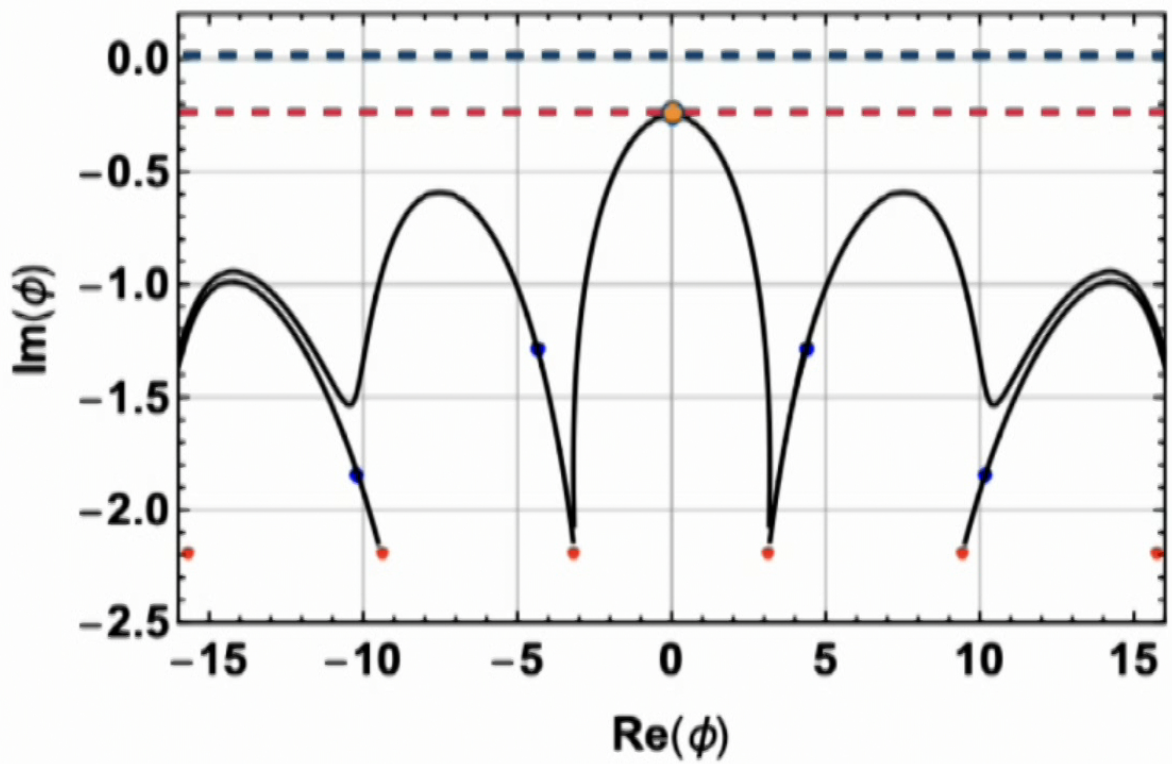
Path integral observables

- Observable: $\langle\mathcal{O}\rangle = \frac{1}{Z}\int\mathcal{D}\Phi\, e^{-S[\Phi]}\mathcal{O}[\Phi] \equiv \int\mathcal{D}\Phi\, p_S[\Phi]\,\mathcal{O}[\Phi]$
- Probability density: $p_S[\Phi] \overset{!}{\geq} 0$
- Sign Problem: $S[\Phi] \notin \mathbb{R} \Rightarrow p_S[\Phi] \ngeq 0$
- Reweighting:

$$\langle\mathcal{O}\rangle = \frac{1}{Z}\int\mathcal{D}\Phi\, e^{-\Re S}e^{-i\Im S}\mathcal{O}[\Phi] = \frac{\int\mathcal{D}\Phi\, e^{-\Re S}e^{-i\Im S}\mathcal{O}[\Phi]}{\int\mathcal{D}\Phi\, e^{-\Re S}} \cdot \frac{\int\mathcal{D}\Phi\, e^{-\Re S}}{\int\mathcal{D}\Phi\, e^{-\Re S}e^{-i\Im S}}$$

Lefschetz Thimbles [Alexandru *et al.* 2016; Lefschetz 1921; Tanizaki *et al.* 2016]

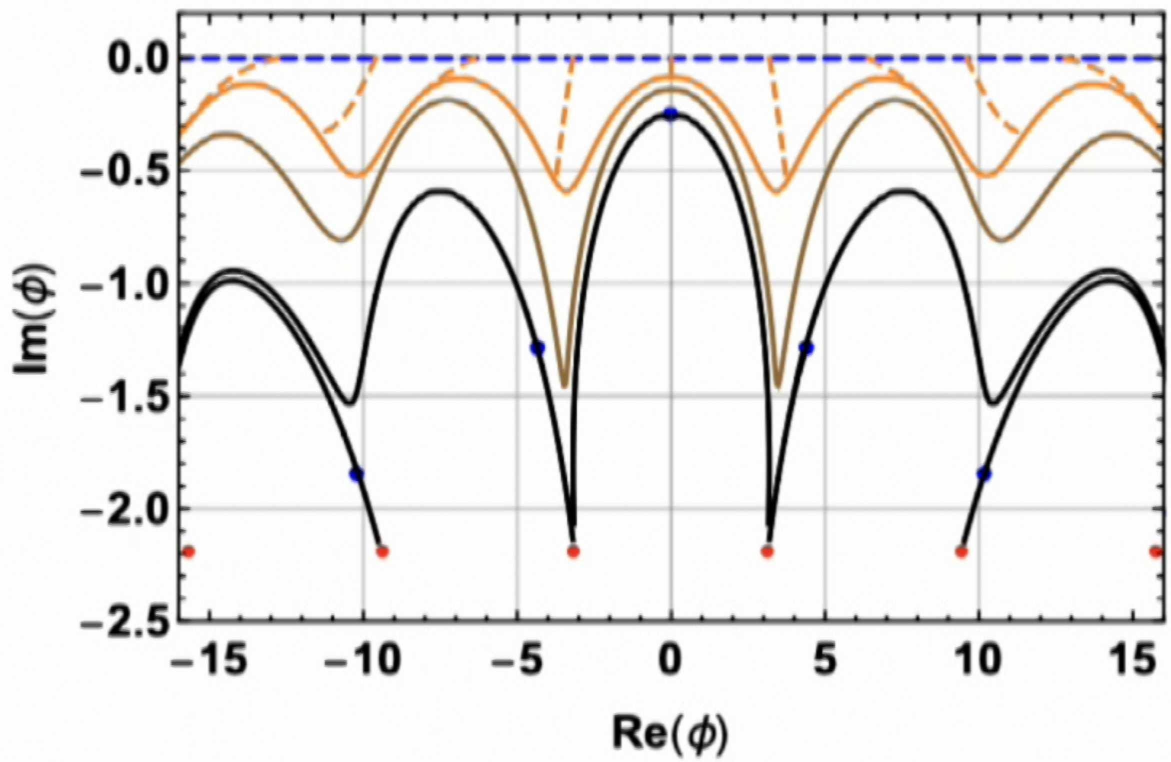
- Manifolds of constant $\Im S$



Johann Ostmeyer

Holomorphic flow [Cristoforetti *et al.* 2012]

$$\frac{d\Phi(\tau)}{d\tau} = \left(\frac{\partial S\left[\Phi(\tau)\right]}{\partial\Phi(\tau)}\right)^*$$



Use Machine Learning [Alexandru *et al.* 2017; Wynen *et al.* 2021]

- Flow some random field configurations
- 'Learn' structure of Lefschetz Thimbles from flowed data
- SHIFT: $\mathbb{R}^n \to \mathbb{C}^n$, $\phi \mapsto \phi + i\mathcal{N}\mathcal{N}(\phi)$

Use Machine Learning [Alexandru *et al.* 2017; Wynen *et al.* 2021]

- Flow some random field configurations
- 'Learn' structure of Lefschetz Thimbles from flowed data
- SHIFT: $\mathbb{R}^n \to \mathbb{C}^n$, $\phi \mapsto \phi + i\mathcal{N}\mathcal{N}(\phi)$
- Apply reweighting $\Rightarrow$ SHIFT doesn't have to be perfect

Statistical Power

$$\langle \mathcal{O} \rangle = \frac{\left\langle e^{-i\Im S}\mathcal{O}\right\rangle_{\Re S}}{\left\langle e^{-i\Im S}\right\rangle_{\Re S}}$$

$$\Sigma := \left\langle e^{-i\Im S}\right\rangle_{\Re S} \equiv \frac{\int \mathcal{D}\Phi\; e^{-\Re S} e^{-i\Im S}}{\int \mathcal{D}\Phi\; e^{-\Re S}}$$

Statistical Power

$$\langle \mathcal{O} \rangle = \frac{\left\langle e^{-i\Im S}\mathcal{O}\right\rangle_{\Re S}}{\left\langle e^{-i\Im S}\right\rangle_{\Re S}} = \frac{1}{\Sigma}\left\langle e^{-i\Im S}\mathcal{O}\right\rangle_{\Re S}$$

$$\Sigma := \left\langle e^{-i\Im S}\right\rangle_{\Re S} \equiv \frac{\int \mathcal{D}\Phi\; e^{-\Re S} e^{-i\Im S}}{\int \mathcal{D}\Phi\; e^{-\Re S}}$$

Johann Ostmeyer

Statistical Power

$$\langle \mathcal{O} \rangle = \frac{\left\langle e^{-i\Im S}\mathcal{O} \right\rangle_{\Re S}}{\left\langle e^{-i\Im S} \right\rangle_{\Re S}} = \frac{1}{\Sigma} \left\langle e^{-i\Im S}\mathcal{O} \right\rangle_{\Re S}$$

$$\Sigma := \left\langle e^{-i\Im S} \right\rangle_{\Re S} \equiv \frac{\int \mathcal{D}\Phi \; e^{-\Re S} e^{-i\Im S}}{\int \mathcal{D}\Phi \; e^{-\Re S}}$$

Statistical Power

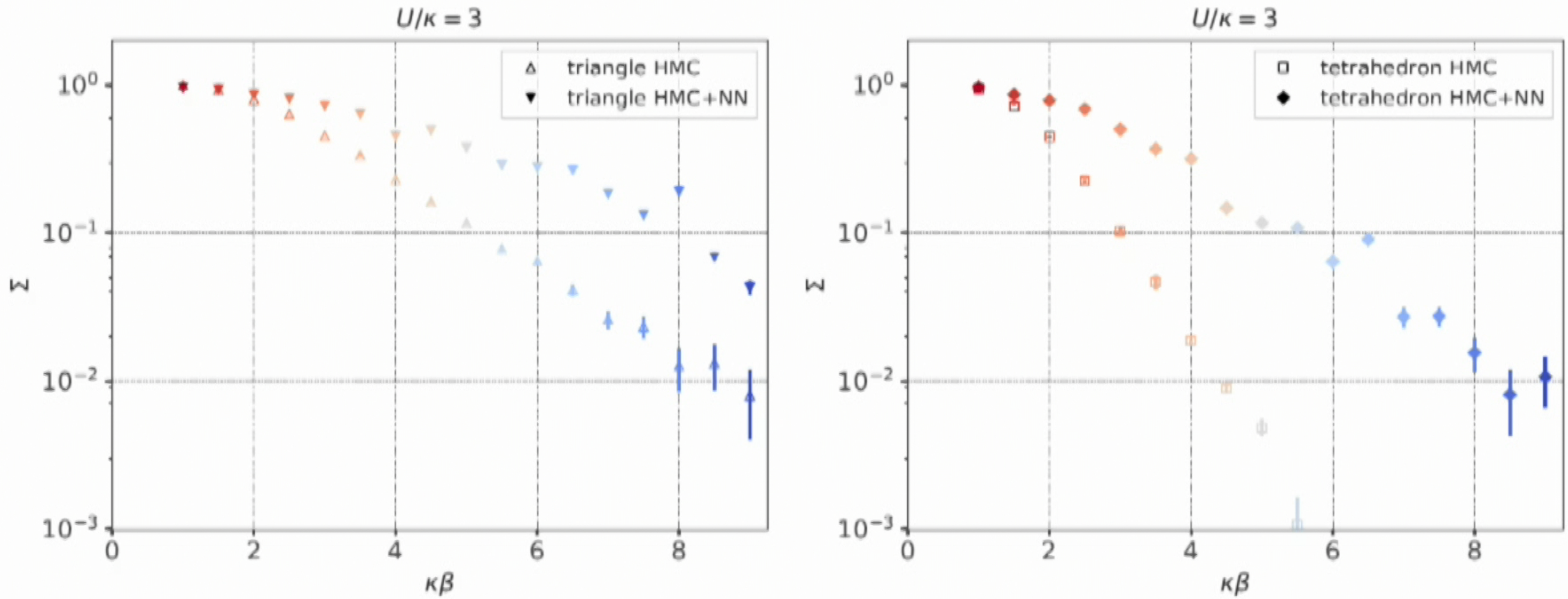
$$\langle \mathcal{O} \rangle = \frac{\left\langle e^{-i\Im S} \mathcal{O} \right\rangle_{\Re S}}{\left\langle e^{-i\Im S} \right\rangle_{\Re S}} = \frac{1}{\Sigma} \left\langle e^{-i\Im S} \mathcal{O} \right\rangle_{\Re S}$$

$$\Sigma := \left\langle e^{-i\Im S} \right\rangle_{\Re S} \equiv \frac{\int \mathcal{D}\Phi \; e^{-\Re S} e^{-i\Im S}}{\int \mathcal{D}\Phi \; e^{-\Re S}}$$

$$N^{\mathrm{eff}} = |\Sigma|^2 \cdot N$$

statistical error $\sim 1/\sqrt{N^{\mathrm{eff}}}$

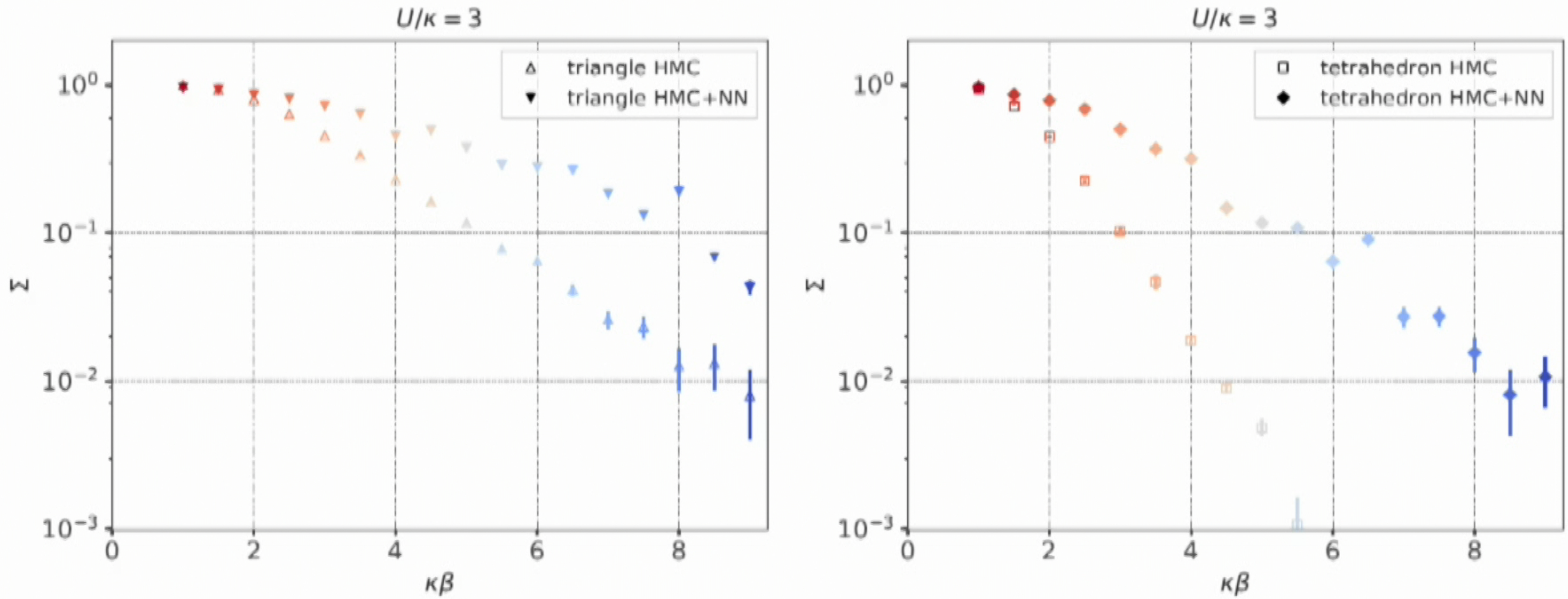
Statistical Power of different geometries [Wynen *et al.* 2021]



Johann Ostmeyer

12/20

Statistical Power of different geometries [Wynen *et al.* 2021]



- Sign problem exponentially bad in volume
- Weaker scaling with Neural Network

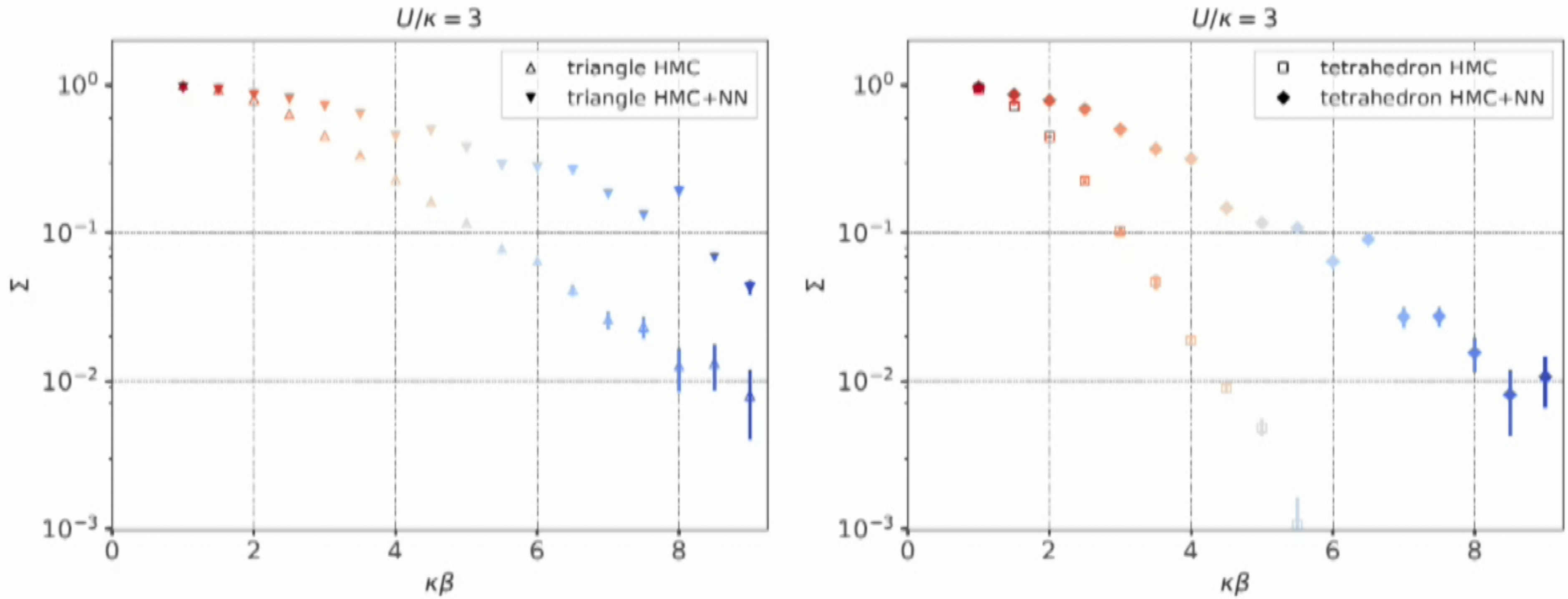
Transform path integral contour

$$\mathcal{Z} = \int_{\mathcal{M}} \mathcal{D}\Phi \; e^{-S[\Phi]}$$
$$= \int_{\widetilde{\mathcal{M}}} \mathcal{D}\phi \; \det J \left[\Phi\left(\phi\right)\right] e^{-S[\Phi(\phi)]} \,, \quad J_{ij} = \frac{\partial \Phi_i}{\partial \phi_j}$$

Transform path integral contour

$$
\begin{aligned}
\mathcal{Z} &= \int_{\mathcal{M}} \mathcal{D}\Phi \; e^{-S[\Phi]} \\
&= \int_{\widetilde{\mathcal{M}}} \mathcal{D}\phi \; \det J\left[\Phi\left(\phi\right)\right] e^{-S[\Phi(\phi)]}\,, \quad J_{ij} = \frac{\partial \Phi_i}{\partial \phi_j} \\
\Rightarrow S^{\text{eff}}\left[\phi\right] &= S\left[\Phi(\phi)\right] - \log\det J\left[\Phi(\phi)\right]
\end{aligned}
$$

Statistical Power of different geometries [Wynen *et al.* 2021]



- Sign problem exponentially bad in volume
- Weaker scaling with Neural Network

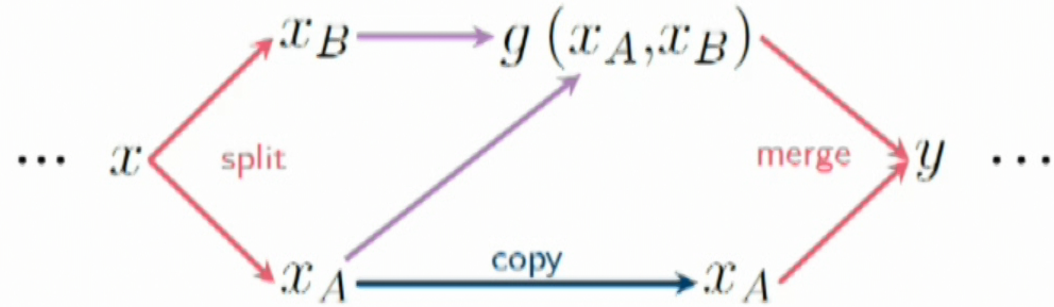
Transform path integral contour

$$\begin{aligned} \mathcal{Z} &= \int_{\mathcal{M}} \mathcal{D}\Phi \; e^{-S[\Phi]} \\ &= \int_{\widetilde{\mathcal{M}}} \mathcal{D}\phi \; \det J\left[\Phi\left(\phi\right)\right] e^{-S[\Phi(\phi)]} \,, \quad J_{ij} = \frac{\partial \Phi_i}{\partial \phi_j} \\ \Rightarrow S^{\text{eff}}\left[\phi\right] &= S\left[\Phi(\phi)\right] - \log\det J\left[\Phi(\phi)\right] \end{aligned}$$

Runtime of determinant as V^3!

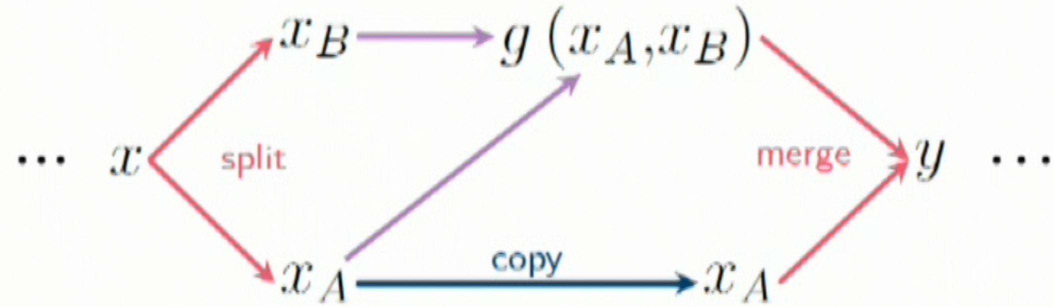
Affine coupling layers [Albergo *et al.* 2021; Dinh *et al.* 2014]

$$f(x) = \begin{cases} y_A = x_A \\ y_B = g(x_A, x_B) \end{cases}$$



Affine coupling layers [Albergo *et al.* 2021; Dinh *et al.* 2014]

$$f(x) = \begin{cases} y_A = x_A \\ y_B = g(x_A, x_B) \end{cases}$$

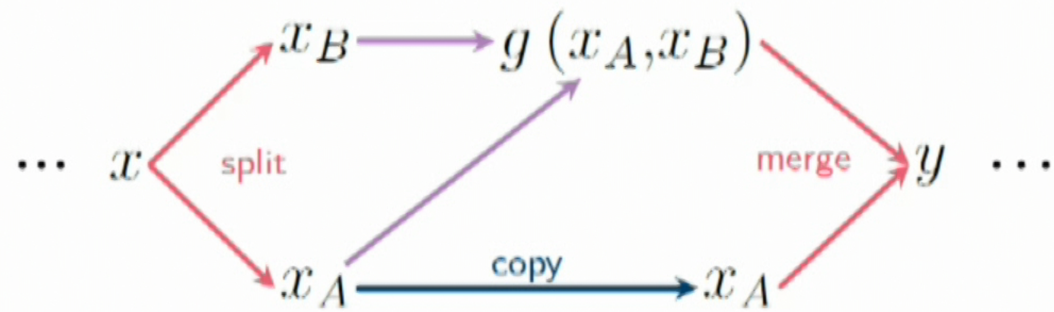


$$g(x_A, x_B) = x_B \odot s(x_A) + t(x_A)$$

$$\Rightarrow \det\left(\frac{\partial f}{\partial x}\right) = \det\begin{pmatrix} \mathbb{1} & 0 \\ \frac{\partial y_B}{\partial x_A} & s(x_A) \end{pmatrix} = \prod_j s(x_A)_j$$

Affine coupling layers [Albergo *et al.* 2021; Dinh *et al.* 2014]

$$f(x) = \begin{cases} y_A = x_A \\ y_B = g(x_A, x_B) \end{cases}$$



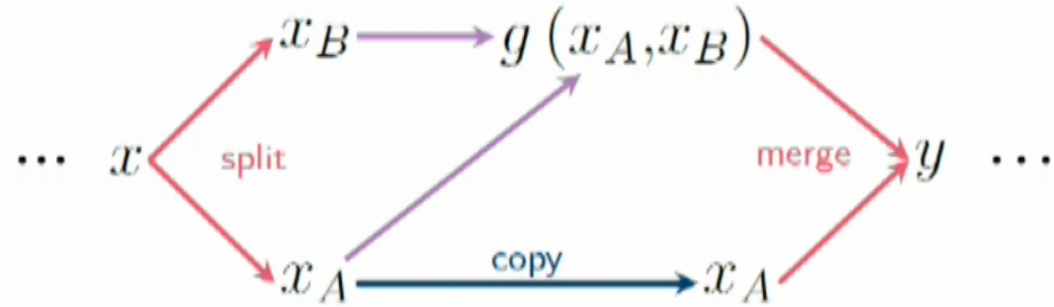
$$g(x_A, x_B) = x_B \odot s(x_A) + t(x_A)$$

$$\Rightarrow \det\left(\frac{\partial f}{\partial x}\right) = \det\begin{pmatrix} \mathbb{1} & 0 \\ \frac{\partial y_B}{\partial x_A} & s(x_A) \end{pmatrix} = \prod_j s(x_A)_j$$

$$\det \frac{\partial \mathcal{N}\mathcal{N}}{\partial x} = \det\left(\frac{\partial f^n(x)}{\partial x}\right) \det\left(\frac{\partial f^{n-1}(x)}{\partial x}\right) \cdots \det\left(\frac{\partial f^1(x)}{\partial x}\right)$$

Affine coupling layers [Albergo *et al.* 2021; Dinh *et al.* 2014]

$$f(x) = \begin{cases} y_A = x_A \\ y_B = g(x_A, x_B) \end{cases}$$



$$g(x_A, x_B) = x_B \odot s(x_A) + t(x_A)$$

$$\Rightarrow \det\left(\frac{\partial f}{\partial x}\right) = \det\begin{pmatrix} \mathbb{1} & 0 \\ \frac{\partial y_B}{\partial x_A} & s(x_A) \end{pmatrix} = \prod_j s(x_A)_j$$

$$\det \frac{\partial \mathcal{N}\mathcal{N}}{\partial x} = \det\left(\frac{\partial f^n(x)}{\partial x}\right)\det\left(\frac{\partial f^{n-1}(x)}{\partial x}\right)\cdots\det\left(\frac{\partial f^1(x)}{\partial x}\right)$$

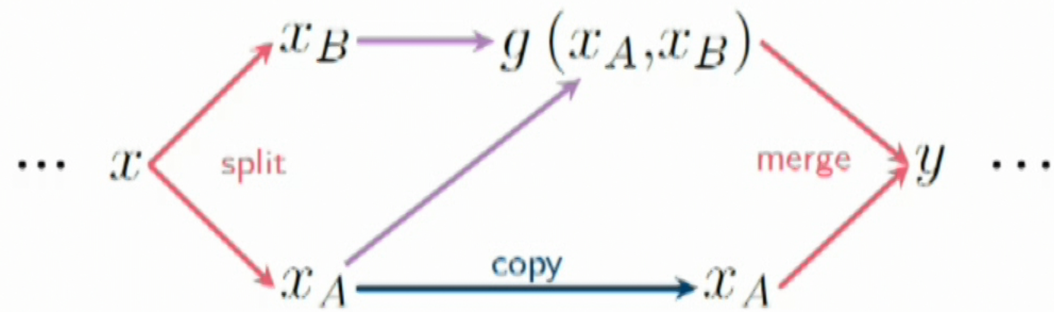
$$\det J = \det\left(\mathbb{1} + i\frac{\partial \mathcal{N}\mathcal{N}}{\partial x}\right)$$

Johann Ostmeyer 14/20

Affine coupling layers [Albergo *et al.* 2021; Dinh *et al.* 2014]

$$f(x) = \begin{cases} y_A = x_A \\ y_B = g(x_A, x_B) \end{cases}$$

$$g(x_A, x_B) = x_B \odot s(x_A) + t(x_A)$$



$$\Rightarrow \det\left(\frac{\partial f}{\partial x}\right) = \det\begin{pmatrix} \mathbb{1} & 0 \\ \frac{\partial y_B}{\partial x_A} & s(x_A) \end{pmatrix} = \prod_j s(x_A)_j$$

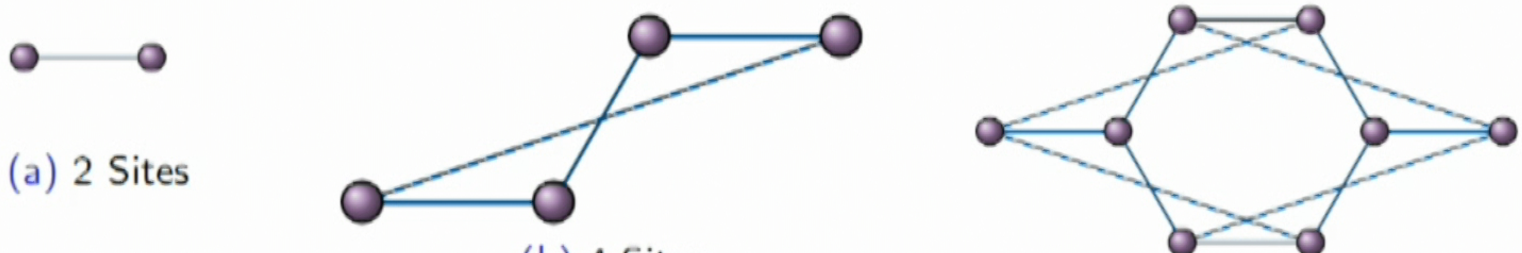
$$\det \frac{\partial \mathcal{N}\mathcal{N}}{\partial x} = \det\left(\frac{\partial f^n(x)}{\partial x}\right) \det\left(\frac{\partial f^{n-1}(x)}{\partial x}\right) \cdots \det\left(\frac{\partial f^1(x)}{\partial x}\right)$$

$$\det J = \det\left(\mathbb{1} + i\frac{\partial \mathcal{N}\mathcal{N}}{\partial x}\right) \neq \prod_k \det\left(\mathbb{1} + i\frac{\partial f^k(x)}{\partial x}\right)$$

Complex-valued Neural Networks [Bassey *et al.* 2021; Bouboulis 2010; Brandwood 1983]

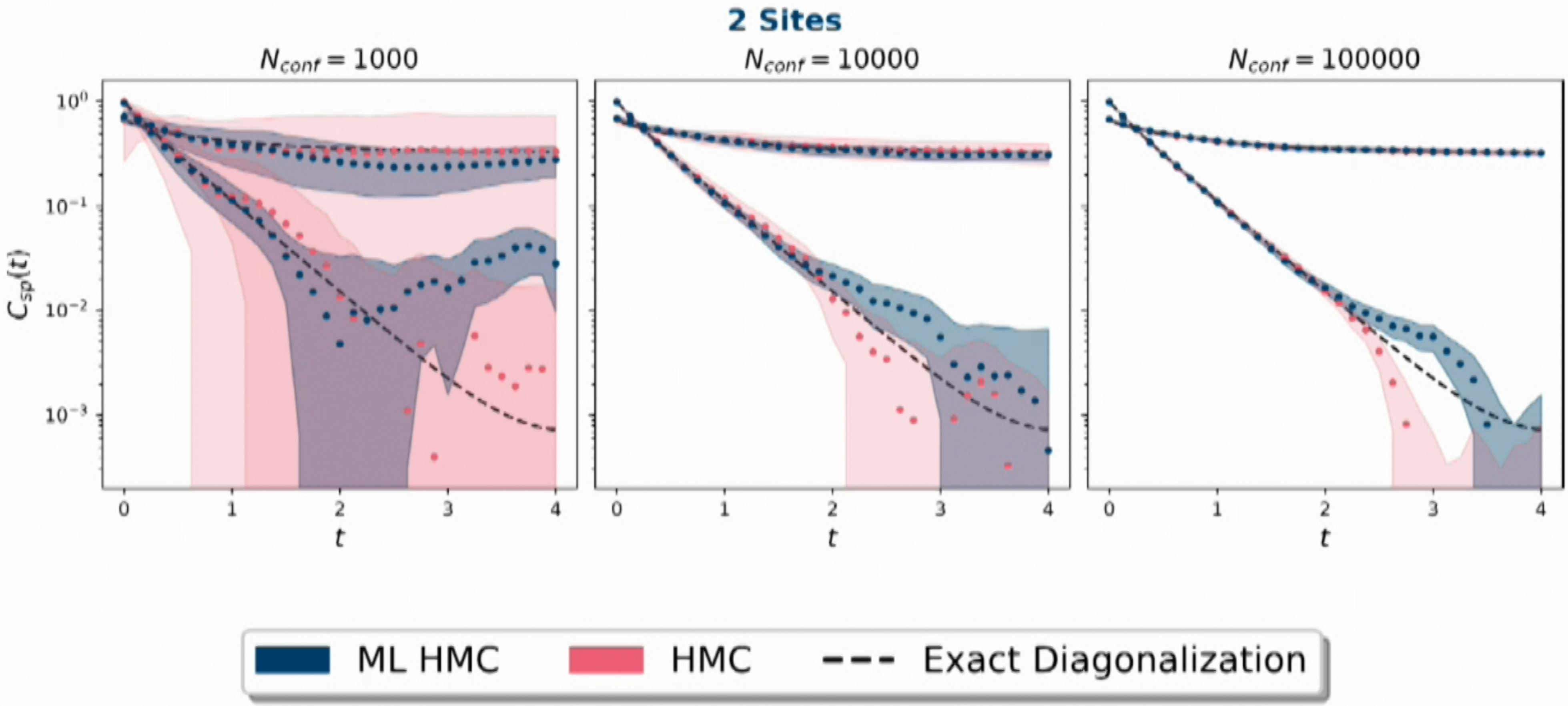
	real	complex
Trafo	$\phi \mapsto \phi + i\mathcal{NN}(\phi)$	$\phi \mapsto \mathcal{NN}(\phi)$
Derivative	$\frac{\partial f(x)}{\partial x}$	$\frac{\partial f(z)}{\partial z} = \frac{1}{2}\left(\frac{\partial f(z)}{\partial \Re z} - i\frac{\partial f(z)}{\partial \Im z}\right)$ $\frac{\partial f(z)}{\partial z^*} = \frac{1}{2}\left(\frac{\partial f(z)}{\partial \Re z} + i\frac{\partial f(z)}{\partial \Im z}\right)$
Layers	dense	affine
$\det J$ runtime	$\mathcal{O}\left(V^3\right)$	$\mathcal{O}\left(V\right)$

Benchmark lattices

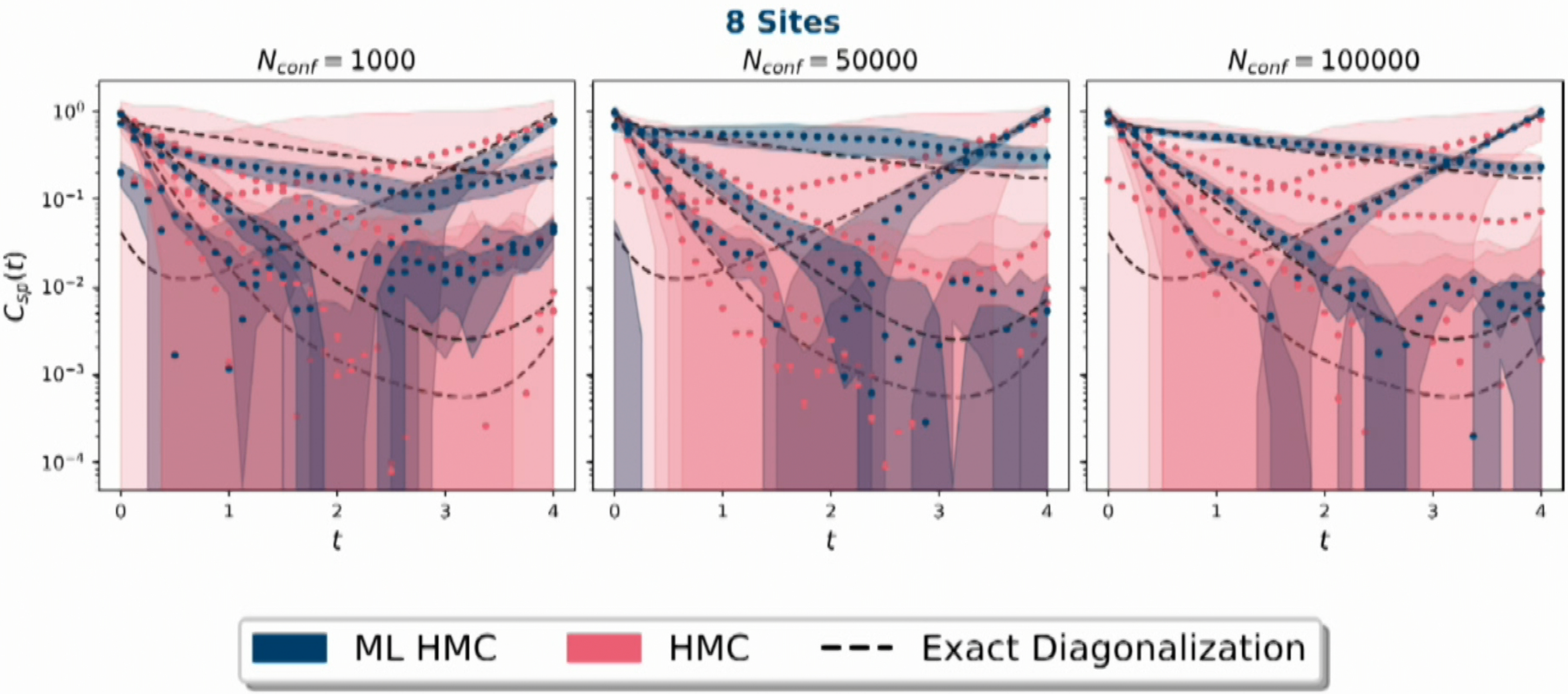


(a) 2 Sites (b) 4 Sites (c) 8 Sites

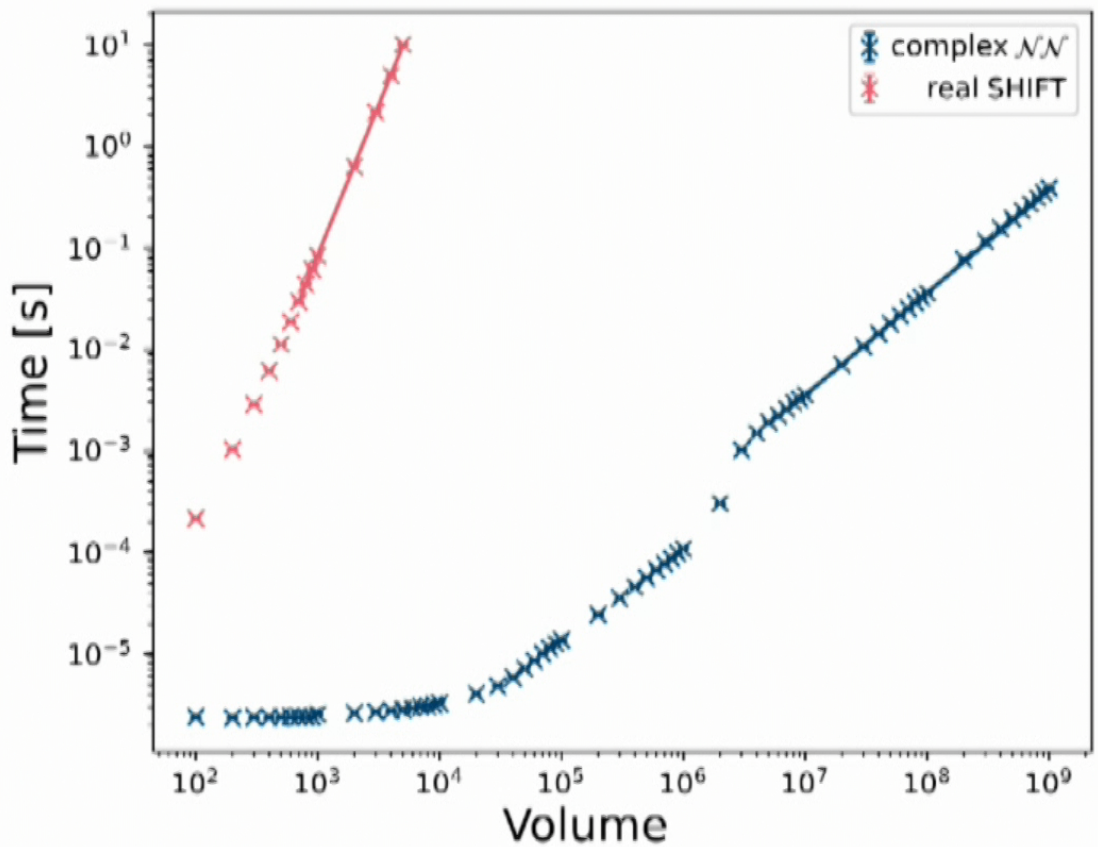
Single Particle Correlators



Single Particle Correlators



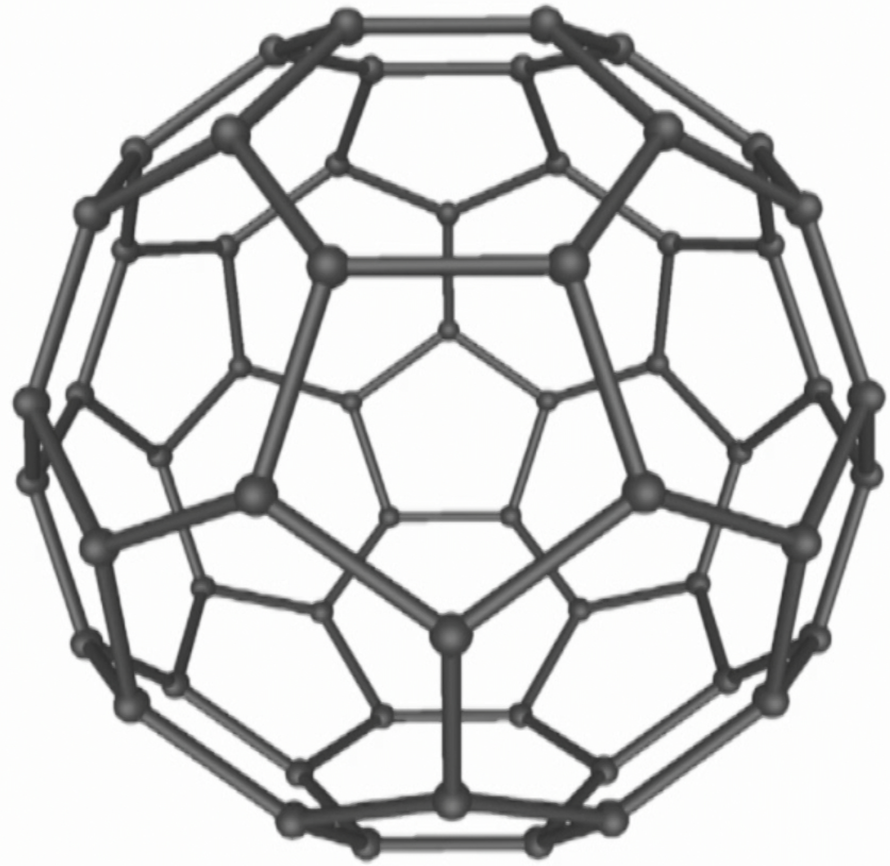
Runtime



Speedup $\mathcal{O}\left(V^2\right)$!

What next?

- C20, C60 (buckyballs), ...

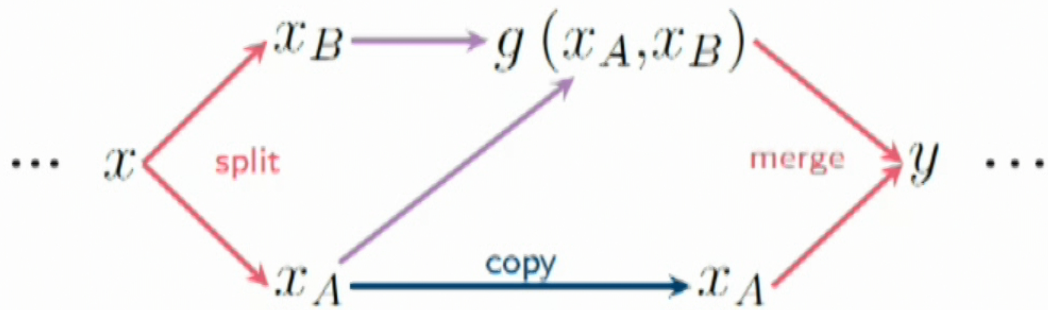


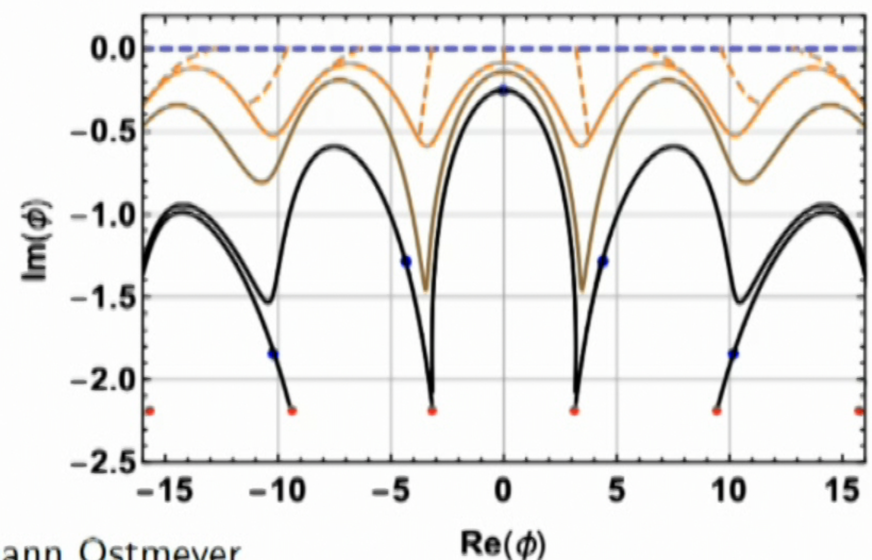
[Wikipedia 2021]

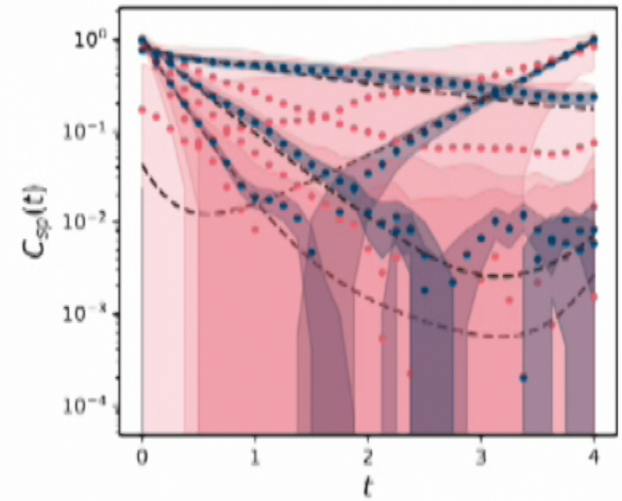
What next?

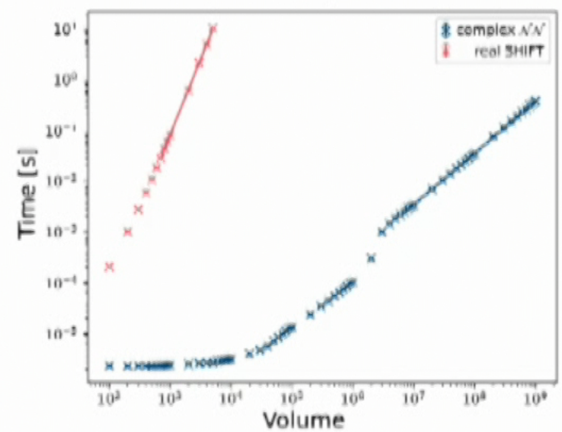
- C20, C60 (buckyballs), ...
- Convolutional / Sparse Neural Networks?

Mitigating the Hubbard Sign Problem with Complex-Valued Neural Networks [Rodekamp *et al.* 2022]









Johann Ostmeyer