Leveraging Periodic as a Quantum PDE Solver

Related references

Leveraging Periodic as a Quantum PDE Solver, to appear,
Joint with **Mariia Sobchuk, Xiaoran Li, Arsalan Motamedi, Grecia Castelazo, Ala Shayeghi**

Gibbs Sampling of Periodic Potentials on a Quantum Computer. arXiv:2210.08104 (2022), and Proceedings of the 41st International Conference on Machine Learning, PMLR 235:36322-36371, 2024
Joint with **Arsalan Motamedi**.

Pooya Ronagh
Research Assistant Professor | IQC | University of Waterloo
Chief Technology Officer | 1QBit

Need for structure

Need for structure

- Is quantum utility limited to the simulation of inherently quantum systems?

Need for structure

- Is quantum utility limited to the simulation of inherently quantum systems?

"Nature isn't classical, dammit, and if you want to make a simulation of nature, you'd better make it quantum mechanical, and by golly it's a wonderful problem, because it doesn't look so easy."

Need for structure

- Is quantum utility limited to the simulation of inherently quantum systems?

"Nature isn't classical, dammit, and if you want to make a simulation of nature, you'd better make it quantum mechanical, and by golly it's a wonderful problem, because it doesn't look so easy."

- What properties should a a Boolean function have?

$$|x\rangle|y\rangle \rightarrow |x\rangle|y \oplus f(x)\rangle$$

THEORY OF COMPUTING, Volume 10 (6), 2014, pp. 133–166
www.theoryofcomputing.org

The Need for Structure in Quantum Speedups

Scott Aaronson* Andris Ambainis†

Received October 22, 2012; Revised July 10, 2014; Published August 12, 2014

… Unfortunately, this "need for structure" has essentially limited the prospects for superpolynomial quantum speedups to those areas of mathematics that are liable to produce things like periodic sequences or sequences of quadratic residues. This is the fundamental reason why the greatest successes of quantum algorithms research have been in cryptography, and specifically in number theoretic cryptography. It helps to explain why we do not have a fast quantum algorithm to solve NP-complete problems (for example), or to break arbitrary one-way functions.

Need for structure

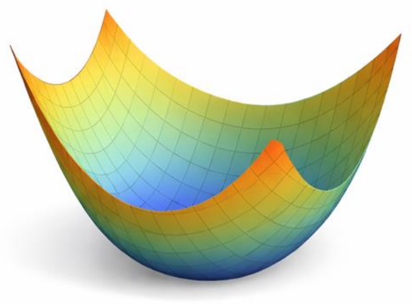
- Is quantum utility limited to the simulation of inherently quantum systems?

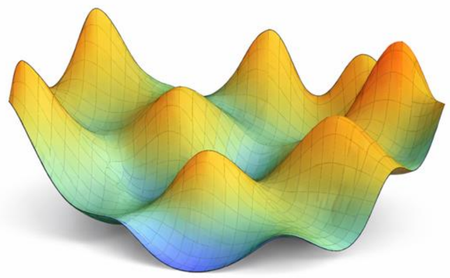
"Nature isn't classical, dammit, and if you want to make a simulation of nature, you'd better make it quantum mechanical, and by golly it's a wonderful problem, because it doesn't look so easy."

- What properties should a a Boolean function have?

$$|x\rangle|y\rangle \rightarrow |x\rangle|y \oplus f(x)\rangle$$

- What about when f is continuous?





THEORY OF COMPUTING, Volume 10 (6), 2014, pp. 133–166
www.theoryofcomputing.org

The Need for Structure in Quantum Speedups

Scott Aaronson* Andris Ambainis†

Received October 22, 2012; Revised July 10, 2014; Published August 12, 2014

… Unfortunately, this "need for structure" has essentially limited the prospects for superpolynomial quantum speedups to those areas of mathematics that are liable to produce things like periodic sequences or sequences of quadratic residues. This is the fundamental reason why the greatest successes of quantum algorithms research have been in cryptography, and specifically in number theoretic cryptography. It helps to explain why we do not have a fast quantum algorithm to solve NP-complete problems (for example), or to break arbitrary one-way functions.

Historic motivation: Solving diffusion equations

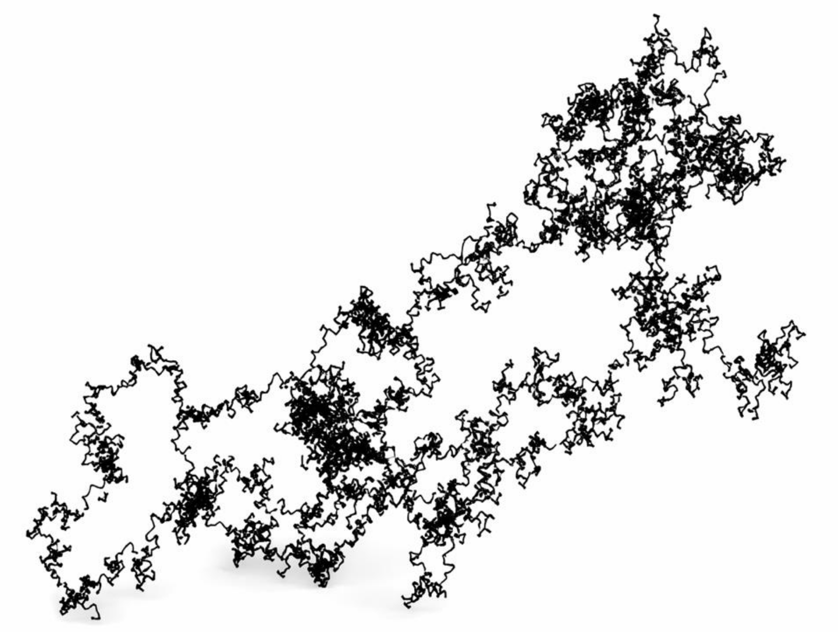
Historic motivation: Solving diffusion equations

- Du and Mordatch (2019): large-scale training of **energy-based models**.

Historic motivation: Solving diffusion equations

- Du and Mordatch (2019): large-scale training of **energy-based models**.
- Gibbs sampling using Langevin dynamics:

$$dX_t = -\frac{1}{2}\beta\sigma^2\nabla_X E_\theta(X)\,dt + \sigma\,dW_t.$$

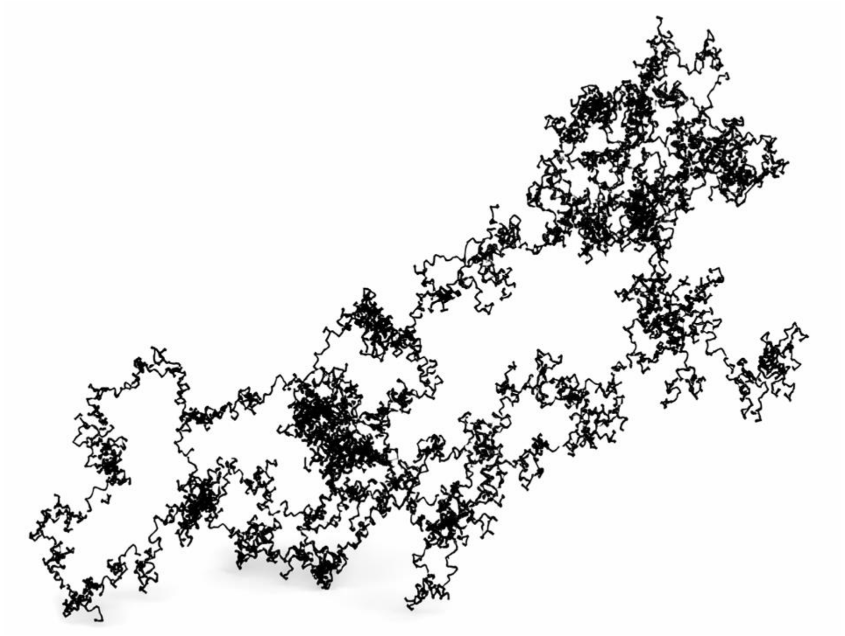


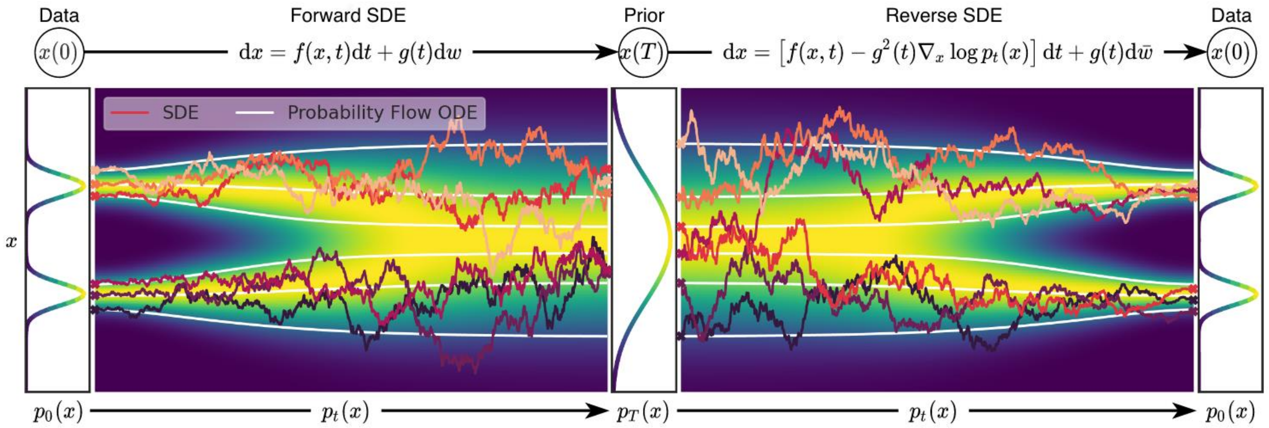
Historic motivation: Solving diffusion equations

- Du and Mordatch (2019): large-scale training of **energy-based models**.
- Gibbs sampling using Langevin dynamics:

$$dX_t = -\frac{1}{2}\beta\sigma^2\nabla_X E_\theta(X)\,dt + \sigma\,dW_t.$$

- Song et al. (2021): **Diffusion models**.





Feynman–Kac formula

Feynman–Kac formula

- An SDE

$$dX_t = b(X_t)dt + \sigma(X_t)dW_t$$

corresponds to a PDE with generator:

$$\mathcal{L} = \sum b_j \frac{\partial}{\partial x_j} + \frac{1}{2} \sum \sum_k \sigma_{ik}(x)\sigma_{jk}(x) \frac{\partial^2}{\partial x_i \partial x_j}$$

Feynman–Kac formula

- An SDE

$$dX_t = b(X_t)dt + \sigma(X_t)dW_t$$

corresponds to a PDE with generator:

$$\mathcal{L} = \sum b_j \frac{\partial}{\partial x_j} + \frac{1}{2}\sum\sum_k \sigma_{ik}(x)\sigma_{jk}(x)\frac{\partial^2}{\partial x_i \partial x_j}$$

- Transition operators $P(t, x, dy) = p(t, x, y)dy$
- Forward and backward Kolmogorov equations:

$$\frac{\partial}{\partial t}p(.,.,y) = \mathcal{L}_x p(.,.,y) \qquad \frac{\partial}{\partial t}p(.,x,.) = \mathcal{L}_y^* p(.,x,.)$$

Feynman–Kac formula

- An SDE

$$dX_t = b(X_t)dt + \sigma(X_t)dW_t$$

corresponds to a PDE with generator:

$$\mathcal{L} = \sum b_j \frac{\partial}{\partial x_j} + \frac{1}{2}\sum\sum_k \sigma_{ik}(x)\sigma_{jk}(x)\frac{\partial^2}{\partial x_i \partial x_j}$$

- Transition operators $P(t,x,dy) = p(t,x,y)dy$
- Forward and backward Kolmogorov equations:

$$\frac{\partial}{\partial t}p(.,.,y) = \mathcal{L}_x p(.,.,y) \qquad \frac{\partial}{\partial t}p(.,x,.) = \mathcal{L}_y^* p(.,x,.)$$

- Parabolic Fokker-Planck equation $\frac{\partial}{\partial t}\rho = \mathcal{L}\rho$;

Feynman–Kac formula

- An SDE

$$dX_t = b(X_t)dt + \sigma(X_t)dW_t$$

corresponds to a PDE with generator:

$$\mathcal{L} = \sum b_j \frac{\partial}{\partial x_j} + \frac{1}{2}\sum\sum_k \sigma_{ik}(x)\sigma_{jk}(x)\frac{\partial^2}{\partial x_i \partial x_j}$$

- Transition operators $P(t, x, dy) = p(t, x, y)dy$
- Forward and backward Kolmogorov equations:

$$\frac{\partial}{\partial t}p(.,.,y) = \mathcal{L}_x p(.,.,y) \qquad \frac{\partial}{\partial t}p(.,x,.) = \mathcal{L}_y^* p(.,x,.)$$

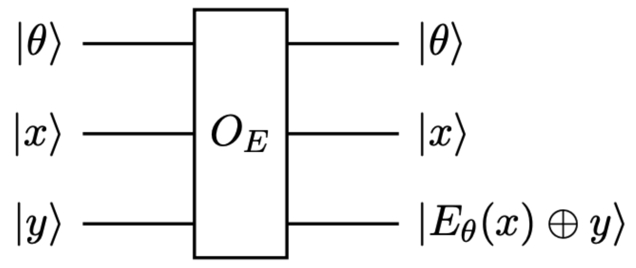
- Parabolic Fokker-Planck equation $\frac{\partial}{\partial t}\rho = \mathcal{L}\rho$;
- Elliptic (stationary) Fokker-Planck equation: $\mathcal{L}\rho = 0$.

A quantum Gibbs sampling algorithm

A quantum Gibbs sampling algorithm

- Instead of Monte-Carlo simulation of Langevin dynamics $dX_t = -\nabla E(X_t)dt + \sqrt{2\beta^{-1}}dW_t$ …
- … solve the **Fokker-Planck equation**

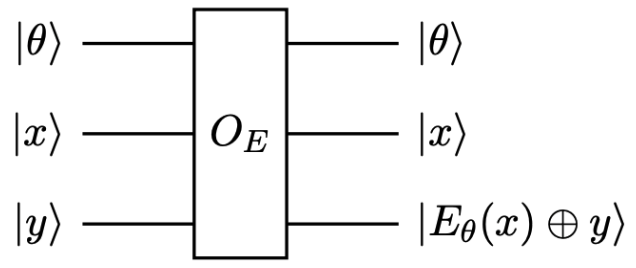
$$\frac{\partial \rho}{\partial t} = \nabla.(\rho \nabla E) + \beta^{-1}\Delta\rho.$$



A quantum Gibbs sampling algorithm

- Instead of Monte-Carlo simulation of Langevin dynamics $dX_t = -\nabla E(X_t)dt + \sqrt{2\beta^{-1}}dW_t$ …
- … solve the **Fokker-Planck equation**

$$\frac{\partial \rho}{\partial t} = \nabla.(\rho \nabla E) + \beta^{-1}\Delta\rho.$$

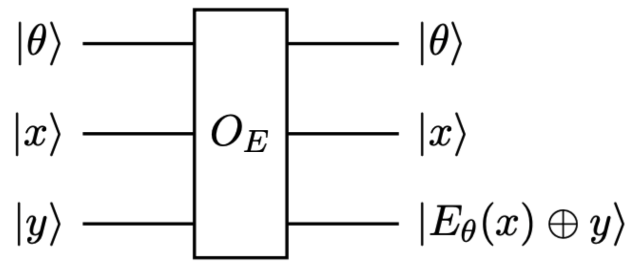


- Our approach: Linearize $\mathcal{L}(-) = \nabla.(e^{-E}\nabla(e^{E} -))$ and solve $\frac{d}{dt}u = \mathbb{L}u$ with quantum linear algebra.

A quantum Gibbs sampling algorithm

- Instead of Monte-Carlo simulation of Langevin dynamics $dX_t = -\nabla E(X_t)dt + \sqrt{2\beta^{-1}}dW_t$ …
- … solve the **Fokker-Planck equation**

$$\frac{\partial \rho}{\partial t} = \nabla.(\rho \nabla E) + \beta^{-1}\Delta\rho.$$



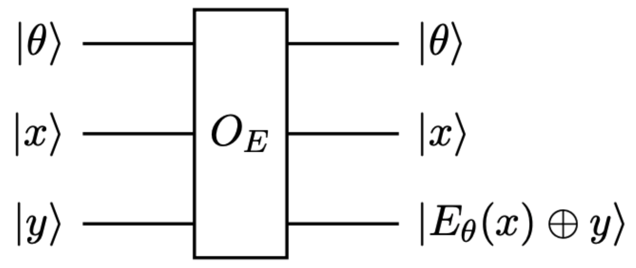
- Our approach: Linearize $\mathcal{L}(-) = \nabla.(e^{-E}\nabla(e^{E} -))$ and solve $\frac{d}{dt}u = \mathbb{L}u$ with quantum linear algebra.
- Issue: High precision requires an exponentially large meshing. Complexity grows with $(1/\epsilon)^d$.

 Pooya Ronagh | University of Waterloo

A quantum Gibbs sampling algorithm

- Instead of Monte-Carlo simulation of Langevin dynamics $dX_t = -\nabla E(X_t)dt + \sqrt{2\beta^{-1}}dW_t$ …
- … solve the **Fokker-Planck equation**

$$\frac{\partial \rho}{\partial t} = \nabla.(\rho \nabla E) + \beta^{-1}\Delta\rho.$$

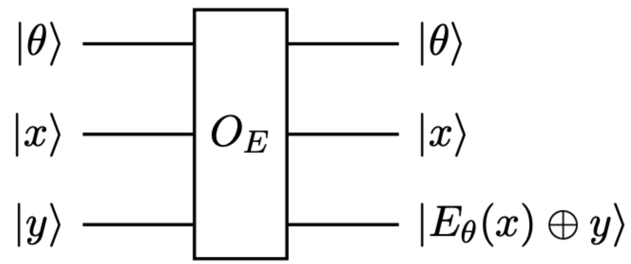


- Our approach: Linearize $\mathcal{L}(-) = \nabla.(e^{-E}\nabla(e^{E} -))$ and solve $\frac{d}{dt}u = \mathbb{L}u$ with quantum linear algebra.
- Issue: High precision requires an exponentially large meshing. Complexity grows with $(1/\epsilon)^d$.
- Motamedi-R 2024: High precision can be achieved, if the solution to the PDE is **<u>periodic</u> and <u>analytic</u>**; high precision $=\ O(C_{PI}\mathrm{polylog}(1/\epsilon))$ complexity.

A quantum Gibbs sampling algorithm

- Instead of Monte-Carlo simulation of Langevin dynamics $dX_t = -\nabla E(X_t)dt + \sqrt{2\beta^{-1}}dW_t$ …
- … solve the **Fokker-Planck equation**

$$\frac{\partial \rho}{\partial t} = \nabla.(\rho \nabla E) + \beta^{-1}\Delta\rho.$$

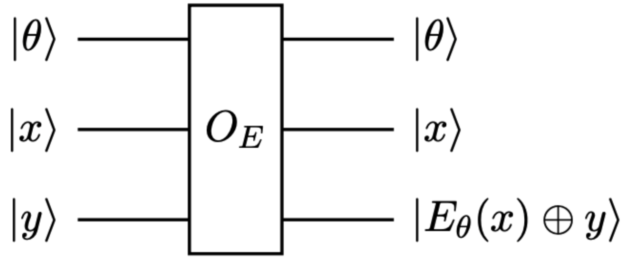


- Our approach: Linearize $\mathcal{L}(-) = \nabla.(e^{-E}\nabla(e^{E}\,-))$ and solve $\frac{d}{dt}u = \mathbb{L}u$ with quantum linear algebra.
- Issue: High precision requires an exponentially large meshing. Complexity grows with $(1/\epsilon)^d$.
- Motamedi-R 2024: High precision can be achieved, if the solution to the PDE is **<u>periodic</u> and <u>analytic</u>**; $\text{high precision} = O(C_{PI}\text{polylog}(1/\epsilon))$ complexity.
- Leng-Ding-Chen-Lin 2025: Improving the operator-level scaling to $O\left(\sqrt{C_{PI}}\text{polylog}(1/\epsilon)\right)$ by solving the elliptic FPE $\mathcal{L}\rho = 0$.

A quantum Gibbs sampling algorithm

- Instead of Monte-Carlo simulation of Langevin dynamics $dX_t = -\nabla E(X_t)dt + \sqrt{2\beta^{-1}}dW_t$ …
- … solve the **Fokker-Planck equation**

$$\frac{\partial \rho}{\partial t} = \nabla.(\rho \nabla E) + \beta^{-1}\Delta\rho.$$



- Our approach: Linearize $\mathcal{L}(-) = \nabla.(e^{-E}\nabla(e^{E}\,-))$ and solve $\frac{d}{dt}u = \mathbb{L}u$ with quantum linear algebra.
- Issue: High precision requires an exponentially large meshing. Complexity grows with $(1/\epsilon)^d$.
- Motamedi-R 2024: High precision can be achieved, if the solution to the PDE is **periodic and analytic**; high precision $=$ $O(C_{PI}\text{polylog}(1/\epsilon))$ complexity.

Why analyticity?

- Leng-Ding-Chen-Lin 2025: Improving the operator-level scaling to $O\left(\sqrt{C_{PI}}\text{polylog}(1/\epsilon)\right)$ by solving the elliptic FPE $\mathcal{L}\rho = 0$.

Critical role of QFT

$$|f_N\rangle := \frac{1}{\sqrt{\langle\!\langle f^2 \rangle\!\rangle_{\Gamma_N}}} \sum_{x \in \Gamma_N} f(x)\,|x\rangle \xrightarrow{F_N^{\otimes d}} |\widetilde{f}_N\rangle := \frac{1}{\sqrt{\langle f^2 \rangle_{\Gamma_N}}} \sum_{\omega \in \Lambda_N} \widetilde{f}_\omega \; |\omega\rangle$$

Pooya Ronagh | University of Waterloo

Critical role of QFT

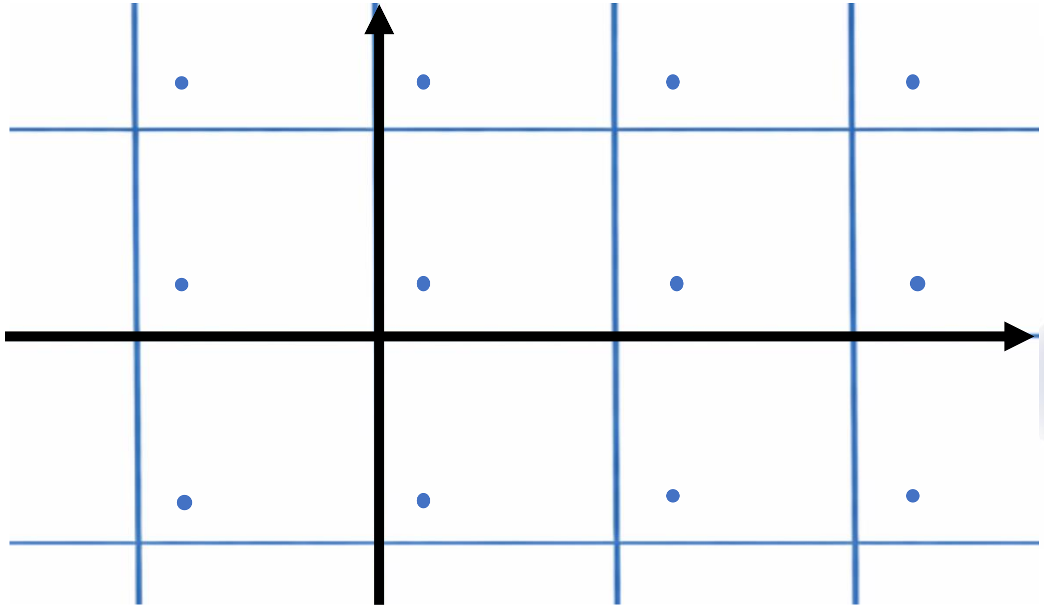
$$|f_N\rangle := \frac{1}{\sqrt{\langle\!\langle f^2 \rangle\!\rangle_{\Gamma_N}}} \sum_{x \in \Gamma_N} f(x)\,|x\rangle \xrightarrow{F_N^{\otimes d}} |\widetilde{f}_N\rangle := \frac{1}{\sqrt{\langle f^2 \rangle_{\Gamma_N}}} \sum_{\omega \in \Lambda_N} \widetilde{f}_\omega\,|\omega\rangle$$

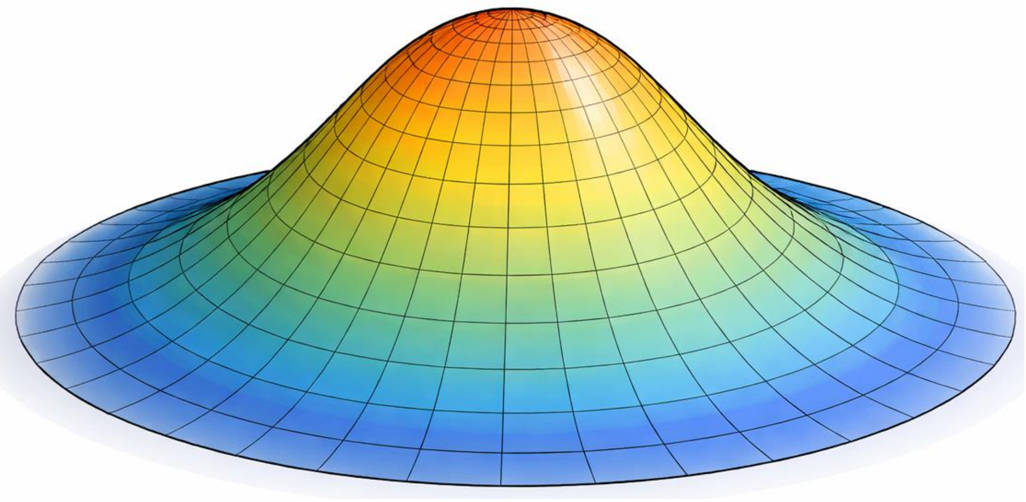
$f(x)$	$= \sum_{\omega \in \mathbb{Z}^d} \widehat{f}_\omega\, e^{i\langle \omega, x\rangle}$	Fourier transform / series
$\widehat{f}_\omega$	$= \frac{1}{(2\pi)^d} \int_{\mathbb{T}^d} f(x) e^{-i\langle \omega, x\rangle}\, dx = \mathbb{E}_{\mathbb{T}^d}[f(x) e^{-i\langle \omega, x\rangle}]$	Fourier coefficients
$\widetilde{f}_\omega$	$= \mathbb{E}_{\Gamma_N}[f(x) e^{-i\langle \omega, x\rangle}]$	Discrete Fourier coefficients
$\|\widetilde{f}_N\rangle$	$= \frac{1}{\sqrt{N^d}} \sum_{\omega \in \Lambda_N} \widetilde{f}_\omega\, \|\omega\rangle = F_N^{\otimes d}\, \|f_N\rangle$	Discrete / quantum Fourier transform
$\widetilde{f}_\omega$	$= \sum_{m \in \mathbb{Z}^d} \widehat{f}_{\omega + (2N+1)m}$	Aliasing sums

Pooya Ronagh | University of Waterloo

Critical role of QFT

$$|f_N\rangle := \frac{1}{\sqrt{\langle\!\langle f^2 \rangle\!\rangle_{\Gamma_N}}} \sum_{x \in \Gamma_N} f(x)\,|x\rangle \xrightarrow{F_N^{\otimes d}} |\widetilde{f}_N\rangle := \frac{1}{\sqrt{\langle f^2 \rangle_{\Gamma_N}}} \sum_{\omega \in \Lambda_N} \widetilde{f}_\omega\,|\omega\rangle$$





$$\widetilde{f}_\omega = \sum_{m \in \mathbb{Z}^d} \widehat{f}_{\omega+(2N+1)m}$$

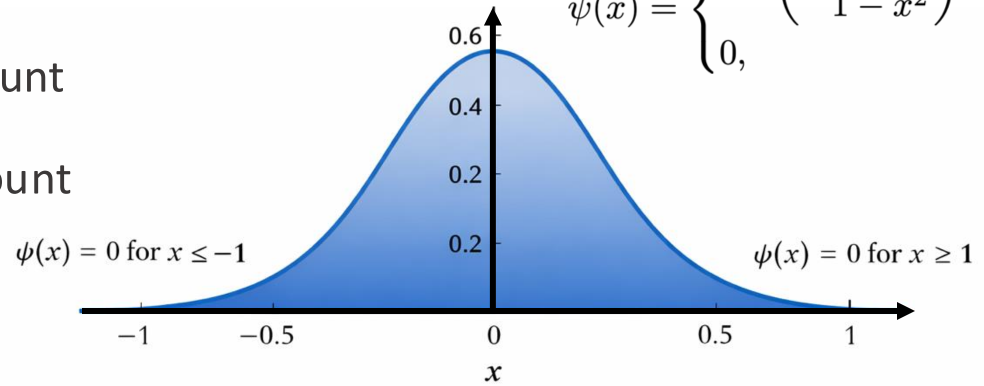
Aliasing sums

More smoothness = Faster decay of spectrum

Regularity class	Decay rate of Fourier coefficients ($\lvert\widehat{f}_\omega\rvert$)	Taildedness at cutoff N (truncation error, E_N)	Cutoff N to guarantee target precision ε
p-smooth ($\mathcal{C}^p$)	$C\lVert\omega\rVert_\infty^{-p}$	$C\sqrt{\frac{d}{d-2p}}N^{-p+d/2}$	$\left(\frac{C}{\varepsilon}\right)^{1/(p-d/2)}$
(p,q)-Hölder ($\mathcal{C}^{p,q}$)	$C\lVert\omega\rVert_\infty^{-p-q}$	$C\sqrt{\frac{d}{d-2p-2q}}N^{-p-q+d/2}$	$\left(\frac{C}{\varepsilon}\right)^{1/(p+q-d/2)}$
smooth ($\mathcal{C}^\infty$)	sub-polynomial	sub-polynomial	super-logarithmic
s-Gevrey ($\mathcal{G}^s$ for $s>1$)	$Ce^{-r\lVert\omega\rVert_\infty^{1/s}}$	$C\sqrt{\frac{ds}{r}N^{d-1/s}}\,e^{-rN^{1/s}}$	$\left(\frac{1}{2r}\log\frac{C^2ds}{r\varepsilon}\right)^s$
Analytic ($\mathcal{C}^\omega=\mathcal{G}^1$)	$Ce^{-r\lVert\omega\rVert_\infty}$	$C\sqrt{\frac{d}{r}N^{d-1}}\,e^{-rN}$	$\frac{1}{2r}\log\frac{C^2d}{r\varepsilon}$

- Faster decay of the Fourier coefficients ⇒ Less aliasing effects
- Only continuity ⇒ No guaranteed bound on qubit count
- Only p-smoothness ⇒ Polynomially growing qubit count
- Analytic functions have the fastest decay.

$$\psi(x)=\begin{cases}\exp\left(-\frac{1}{1-x^2}\right), & |x|<1,\\ 0, & |x|\geq 1.\end{cases}$$



Pooya Ronagh | University of Waterloo

Example: Solving general linear PDEs

$$\sum_{\alpha \in \mathcal{J}} g_\alpha(x) D^\alpha u(x) = \eta(x)$$

Pooya Ronagh | University of Waterloo

Example: Solving general linear PDEs

$$\sum_{\alpha \in \mathcal{J}} g_\alpha(x) D^\alpha u(x) = \eta(x)$$

- The goal is to linear the generator and invert it:

$$\mathcal{L} = \sum_{\alpha \in \mathcal{J}} g_\alpha(x) D^\alpha \longrightarrow \mathbb{L}_N \left| u_N \right\rangle = \left| \eta_N \right\rangle$$

Example: Solving general linear PDEs

$$\sum_{\alpha \in \mathcal{J}} g_\alpha(x) D^\alpha u(x) = \eta(x)$$

- The goal is to linear the generator and invert it:

$$\mathcal{L} = \sum_{\alpha \in \mathcal{J}} g_\alpha(x) D^\alpha \quad \longrightarrow \quad \mathbb{L}_N \left| u_N \right\rangle = \left| \eta_N \right\rangle$$

- We wish to perform:

$$D^\alpha f = \mathcal{F}^{-1} \operatorname{diag} \left\{ (i\omega)^\alpha : \omega \in \mathbb{Z}^d \right\} \mathcal{F}(f)$$

Example: Solving general linear PDEs

$$\sum_{\alpha \in \mathcal{J}} g_\alpha(x) D^\alpha u(x) = \eta(x)$$

- The goal is to linear the generator and invert it:

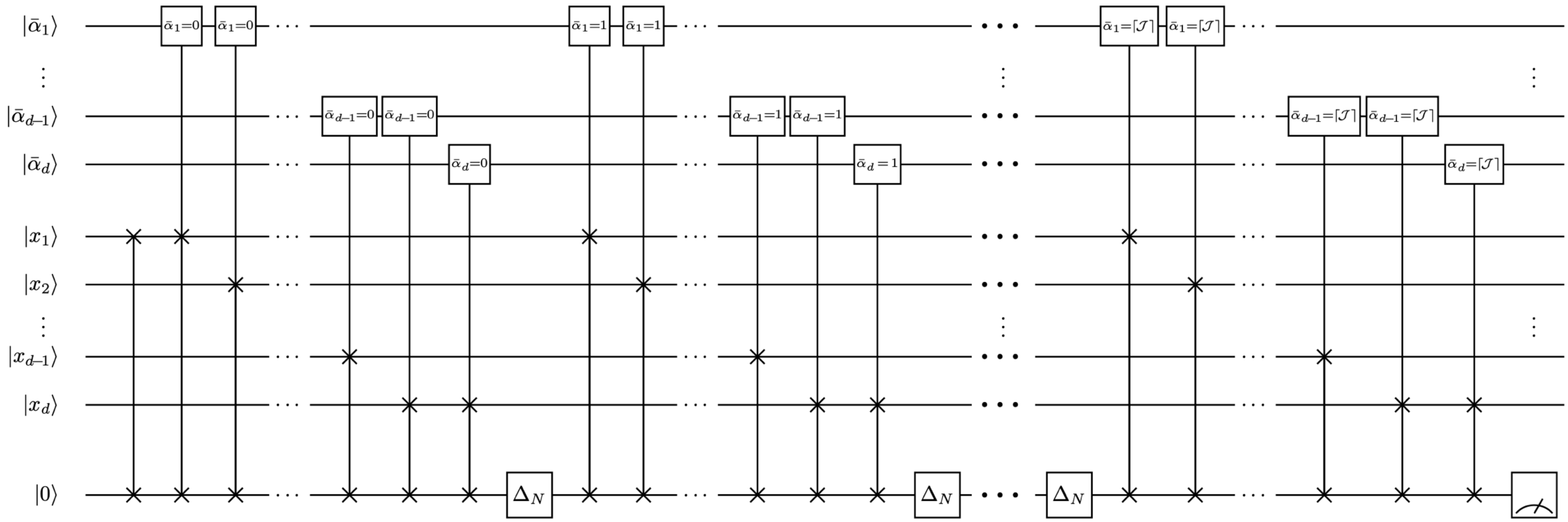
$$\mathcal{L} = \sum_{\alpha \in \mathcal{J}} g_\alpha(x) D^\alpha \longrightarrow \mathbb{L}_N \left| u_N \right\rangle = \left| \eta_N \right\rangle$$

- We wish to perform: $D^\alpha f = \mathcal{F}^{-1} \operatorname{diag} \left\{ (i\omega)^\alpha : \omega \in \mathbb{Z}^d \right\} \mathcal{F}(f)$

- But we only can do: $\widetilde{D}^\alpha = (F_N^{\otimes d})^{-1} \operatorname{diag} \left(\left\{ \prod_{j=1}^{d} (i\omega_j)^{\alpha_j} \right\}_{\omega \in \Lambda_N} \right) F_N^{\otimes d}$

Pooya Ronagh | University of Waterloo

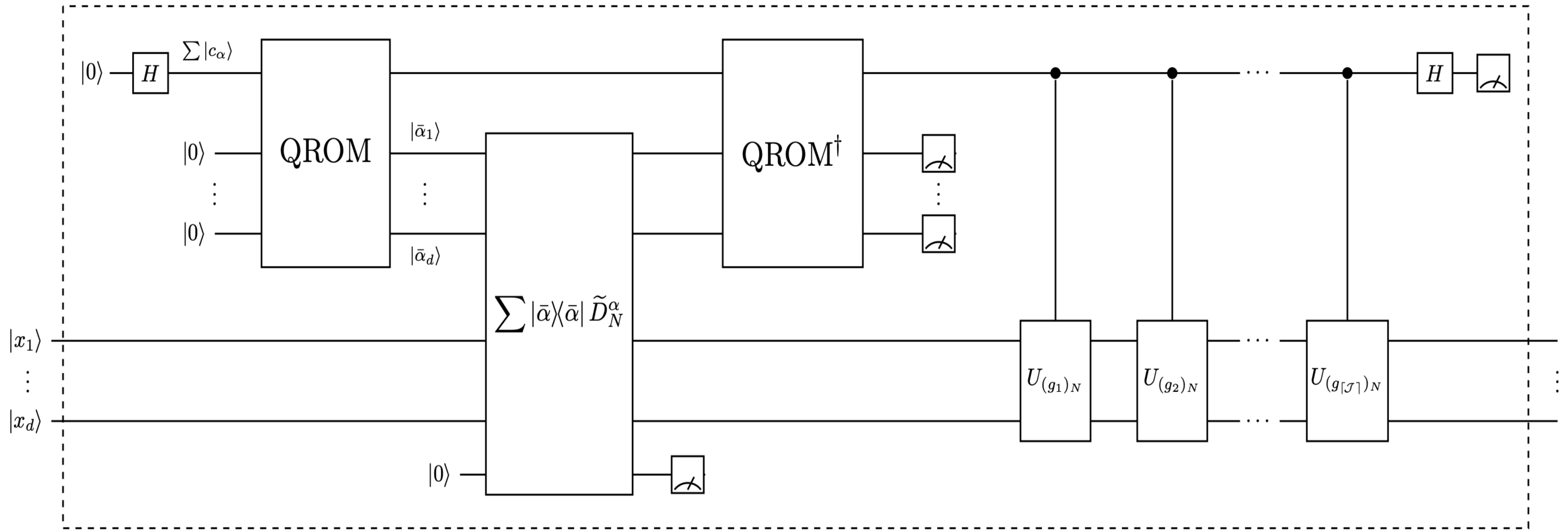
Block encoding of the approximate differentials



$$\Delta_N = F_N^{-1}\,\mathrm{diag}\{(i\omega)_{\omega\in\{-N,\cdots,N\}}\}F_N$$

$O(\log N \log\log N)$ elementary gates
$O(\log N)$ qubits

Block encoding of $\mathbb{L}_N$



$\mathbb{L}_N$

End-to-end resource count

$$\left\| \widetilde{D}_N^{\alpha} f_N - (D^{\alpha} f)_N \right\|_2 \lesssim 2^d N^{|\alpha| + d/2} E_N$$

- To make the curse of dimensionality disappear, E_N must decay exponentially fast.

End-to-end resource count

$$\left\| \widetilde{D}_N^{\alpha} f_N - (D^{\alpha} f)_N \right\|_2 \lesssim 2^d N^{|\alpha|+d/2} E_N$$

- To make the curse of dimensionality disappear, E_N must decay exponentially fast.
- R-Sobchuk-Li-Motamedi-Castelazo-Shayeghi [2026]: N can be chosen as $\Omega\left(\left(ds \log\left(\frac{1}{\epsilon}\right)\right)^s\right)$.

End-to-end resource count

$$\left\| \widetilde{D}_N^{\alpha} f_N - (D^{\alpha} f)_N \right\|_2 \lesssim 2^d N^{|\alpha| + d/2} E_N$$

- To make the curse of dimensionality disappear, E_N must decay exponentially fast.
- R-Sobchuk-Li-Motamedi-Castelazo-Shayeghi [2026]: N can be chosen as $\Omega\left(\left(ds\log\left(\frac{1}{\epsilon}\right)\right)^s\right)$.
- Total gate count: $\widetilde{\mathcal{O}}\left(\frac{\lceil g\rceil(|\mathcal{J}| + \lceil\mathcal{J}\rceil)}{\sigma_{\min}^2 \|u_N\|} s^{\lceil\mathcal{J}\rceil s} d^{\lceil\mathcal{J}\rceil s+1}\left(d\log\left(\frac{1}{\sigma_{\min}\|u_N\|\varepsilon}\right)\right)^{\lceil\mathcal{J}\rceil s+1}\right)$
- Total qubit count: $\mathcal{O}(d(\log N + \log\lceil\mathcal{J}\rceil))$

$$\sum_{\alpha\in\mathcal{J}} g_\alpha(x) D^\alpha u(x) = \eta(x)$$

Application 1: Atomistic simulation $H|\psi\rangle = |\eta\rangle$

Application 1: Atomistic simulation $H|\psi\rangle = |\eta\rangle$

$$H = -\frac{1}{2}\sum_{i=1}^{M}\Delta_{x_i} + V(x), \quad \text{where} \quad V(x) = -\sum_{i=1}^{M}\sum_{k=1}^{P}\frac{Z_k}{r_{ik}} + \sum_{1\leq i<j\leq M}\frac{1}{r_{ij}}$$

Application 1: Atomistic simulation $H|\psi\rangle = |\eta\rangle$

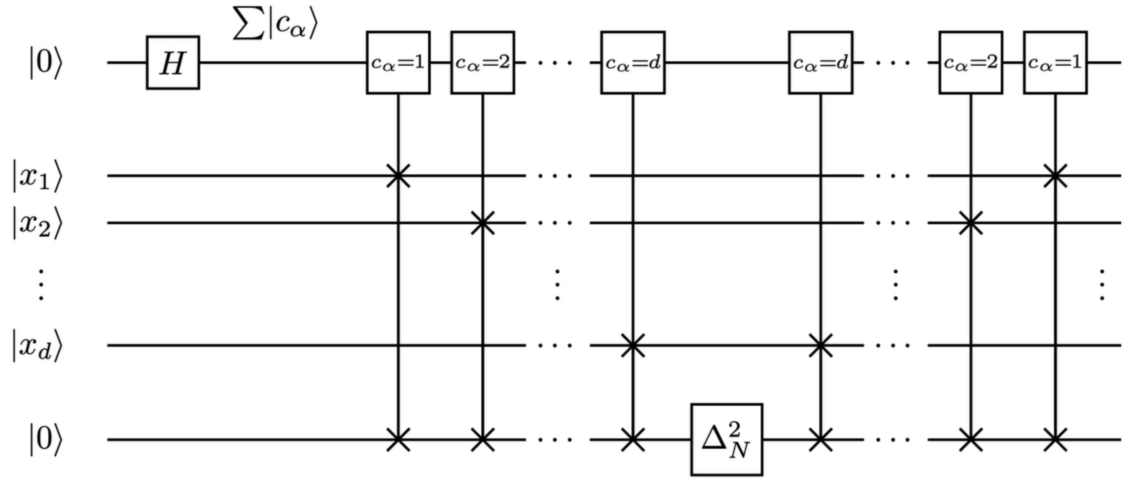
$$H = -\frac{1}{2}\sum_{i=1}^{M}\Delta_{x_i} + V(x), \quad \text{where} \quad V(x) = -\sum_{i=1}^{M}\sum_{k=1}^{P}\frac{Z_k}{r_{ik}} + \sum_{1\le i<j\le M}\frac{1}{r_{ij}}$$

- Total gate count, without V: $\widetilde{\mathcal{O}}\left(\frac{d^4}{\sigma_{\min}^2\|u_N\|}\log^3\left(\frac{1}{\sigma_{\min}\|u_N\|\varepsilon}\right)\right)$
- Total qubit count, without V: $O\left(d\log\frac{1}{\epsilon}\right)$

Application 1: Atomistic simulation $H|\psi\rangle = |\eta\rangle$

$$H = -\frac{1}{2}\sum_{i=1}^{M}\Delta_{x_i} + V(x), \quad \text{where} \quad V(x) = -\sum_{i=1}^{M}\sum_{k=1}^{P}\frac{Z_k}{r_{ik}} + \sum_{1\le i<j\le M}\frac{1}{r_{ij}}$$

- Total gate count, without V: $\widetilde{\mathcal{O}}\left(\frac{d^4}{\sigma_{\min}^2\|u_N\|}\log^3\left(\frac{1}{\sigma_{\min}\|u_N\|\varepsilon}\right)\right)$
- Total qubit count, without V: $O\left(d\log\frac{1}{\epsilon}\right)$
- Simpler block encoding for the Laplacian



Pooya Ronagh | University of Waterloo

Application 1: Atomistic simulation $H|\psi\rangle = |\eta\rangle$

$$H = -\frac{1}{2}\sum_{i=1}^{M}\Delta_{x_i} + V(x), \quad \text{where} \quad V(x) = -\sum_{i=1}^{M}\sum_{k=1}^{P}\frac{Z_k}{r_{ik}} + \sum_{1\leq i<j\leq M}\frac{1}{r_{ij}}$$

Application 1: Atomistic simulation $H|\psi\rangle = |\eta\rangle$

$$H = -\frac{1}{2}\sum_{i=1}^{M}\Delta_{x_i} + V(x), \quad \text{where} \quad V(x) = -\sum_{i=1}^{M}\sum_{k=1}^{P}\frac{Z_k}{r_{ik}} + \sum_{1\leq i<j\leq M}\frac{1}{r_{ij}}$$

- **Challenge:** The solution is not differentiable.

Application 1: Atomistic simulation $H|\psi\rangle = |\eta\rangle$

$$H = -\frac{1}{2}\sum_{i=1}^{M}\Delta_{x_i} + V(x), \quad \text{where} \quad V(x) = -\sum_{i=1}^{M}\sum_{k=1}^{P}\frac{Z_k}{r_{ik}} + \sum_{1\le i<j\le M}\frac{1}{r_{ij}}$$

- **Challenge:** The solution is not differentiable.
- **Solution:** Use a Jastrow factorization with $J = e^U$ where

$$U(x) = -\sum_{i=1}^{M}\sum_{k=1}^{P} Z_k\, r_{ik} + \frac{1}{2}\sum_{1\le i<j\le M} r_{ij}$$

Application 1: Atomistic simulation $H|\psi\rangle = |\eta\rangle$

$$H = -\frac{1}{2}\sum_{i=1}^{M}\Delta_{x_i} + V(x), \qquad \text{where} \qquad V(x) = -\sum_{i=1}^{M}\sum_{k=1}^{P}\frac{Z_k}{r_{ik}} + \sum_{1\le i<j\le M}\frac{1}{r_{ij}}$$

- **Challenge:** The solution is not differentiable.
- **Solution:** Use a Jastrow factorization with $J = e^U$ where

$$U(x) = -\sum_{i=1}^{M}\sum_{k=1}^{P} Z_k\, r_{ik} + \frac{1}{2}\sum_{1\le i<j\le M} r_{ij}$$

- Total gate count with V:

$$\widetilde{\mathcal{O}}\left(\frac{MP^2Z^2d^4}{\sigma_{\min}^2\|\Phi_N\|}\log^3\left(\frac{1}{\sigma_{\min}\|\Phi_N\|\varepsilon}\right)\right)$$

Pooya Ronagh | University of Waterloo

Application 1: Atomistic simulation $H|\psi\rangle = |\eta\rangle$

$$H = -\frac{1}{2}\sum_{i=1}^{M}\Delta_{x_i} + V(x), \qquad \text{where} \qquad V(x) = -\sum_{i=1}^{M}\sum_{k=1}^{P}\frac{Z_k}{r_{ik}} + \sum_{1\le i<j\le M}\frac{1}{r_{ij}}$$

- **Challenge:** The solution is not differentiable.
- **Solution:** Use a Jastrow factorization with $J = e^U$ where

$$U(x) = -\sum_{i=1}^{M}\sum_{k=1}^{P} Z_k\, r_{ik} + \frac{1}{2}\sum_{1\le i<j\le M} r_{ij}$$

- Total gate count with V: $\widetilde{\mathcal{O}}\left(\frac{MP^2Z^2d^4}{\sigma_{\min}^2\|\Phi_N\|}\log^3\left(\frac{1}{\sigma_{\min}\|\Phi_N\|\varepsilon}\right)\right)$

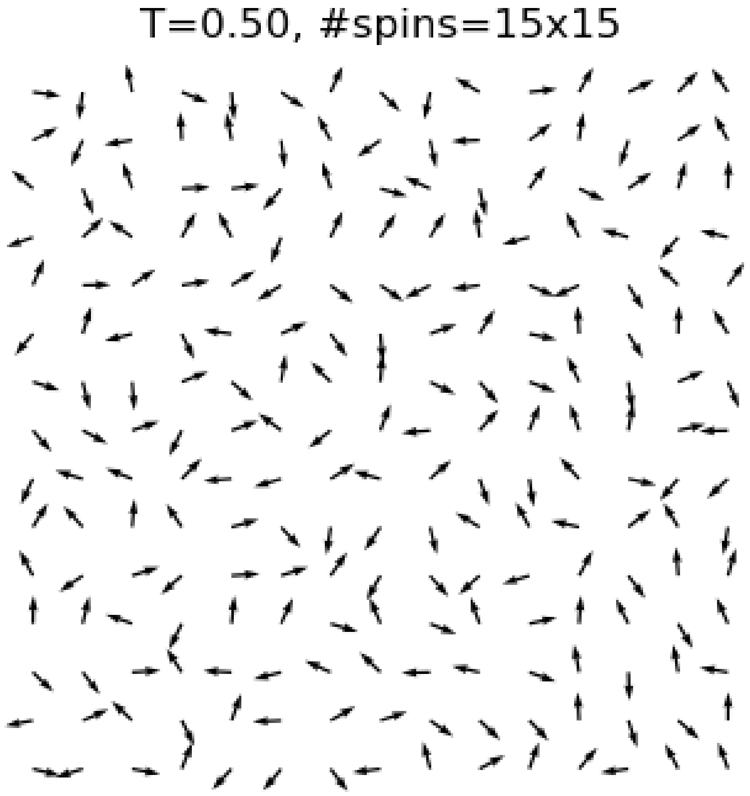
- Note: Classical runtime is $O\big((1/\epsilon)^{3M}\big)$.

Application 2: The classical XY model

- XY Hamiltonian

$$H(\mathbf{s}) = -\sum_{i \neq j} J_{ij}\ \mathbf{s}_i \cdot \mathbf{s}_j - \sum_j \mathbf{h}_j \cdot \mathbf{s}_j$$

$$= -\sum_{i \neq j} J_{ij}\ \cos(\theta_i - \theta_j) - \sum_j h_j \cos\theta_j$$



https://shilingliang.com/XY-MODEL/

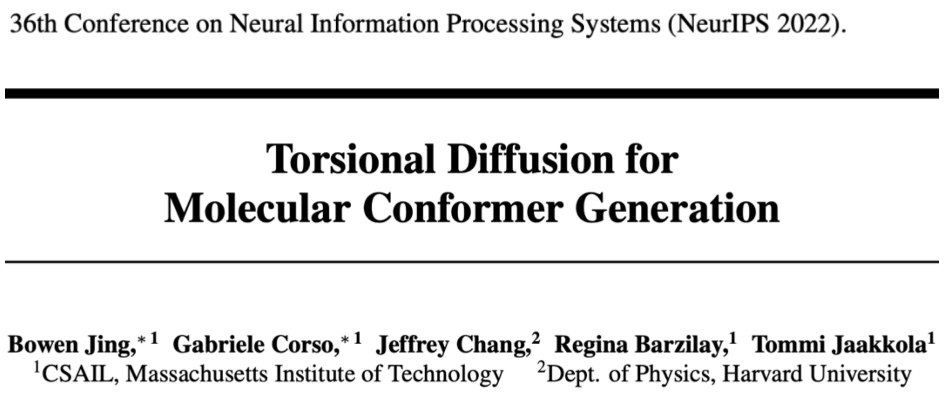
- Generalizations:
  - Heisenberg model
  - The n-vector model
  - Higgs sector of the standard model
- Other applications:
  - U(1) symmetric quantum field theories and gauge theories
  - Translationally invariant problems in condensed matter physics

Application 3: Molecular dynamics

Force field as a function of torsion angles:

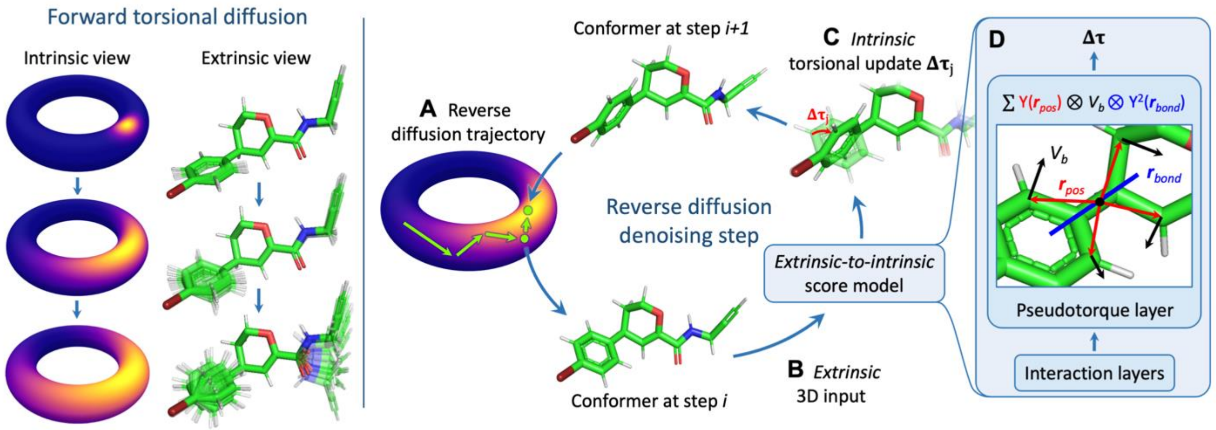
$$U = \sum_{\text{bonds}} \frac{1}{2} k_b (r - r_0)^2 + \sum_{\text{angles}} \frac{1}{2} k_a (\theta - \theta_0)^2 + \sum_{\text{torsions}} \frac{V_n}{2} [1 + \cos(n\phi - \delta)]$$

$$+ \sum_{\text{improper}} V_{imp} + \sum_{\text{LJ}} 4\epsilon_{ij} \left(\frac{\sigma_{ij}^{12}}{r_{ij}^{12}} - \frac{\sigma_{ij}^{6}}{r_{ij}^{6}} \right) + \sum_{\text{elec}} \frac{q_i q_j}{r_{ij}},$$


36th Conference on Neural Information Processing Systems (NeurIPS 2022).

Torsional Diffusion for Molecular Conformer Generation

Bowen Jing,*[1] Gabriele Corso,*[1] Jeffrey Chang,[2] Regina Barzilay,[1] Tommi Jaakkola[1]
[1]CSAIL, Massachusetts Institute of Technology [2]Dept. of Physics, Harvard University



Conclusions

- The hardness of preparing continuous functions on a quantum computer is related to the degree of smoothness of them.

- Quantum ODE/PDE solvers can have interesting applications at the 10s to 100s of logical qubits.

- Quantum simulation using DFT pipelines is an interesting alternative to the common approach of mapping occupancy algebras to the spaces of qubit-based states.

- Potential applications include material science, condensed matter physics, and molecular dynamics simulations.

Thank you for your attention!

pooya.ronagh@uwatelroo.ca

UNIVERSITY OF WATERLOO | IQC Institute for Quantum Computing | 1QBit | PERIMETER INSTITUTE