Machine Learning in Quantum Mechanics

Normalizing Flows for Computing Molecular Vibrational Wave Functions

Nicolas Mendoza

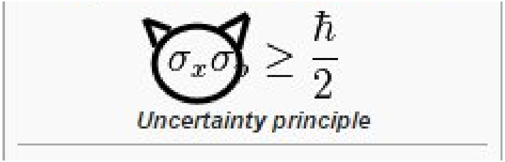
Hamburg, 07.09.2022

Contents

The Challenge

- Solving Schrödinger's Equation is **hard**
- Usually turn to numerical approximations
- …but numerics have limitations
- Amount of data needed depends exponentially on d

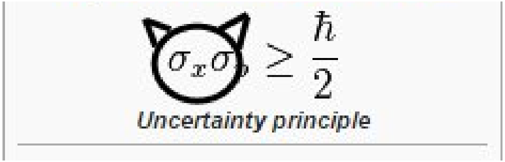
- *The Curse of Dimensionality*
- How do we improve dependency?
- More flexibility to our basis elements
- Machine Learning comes into play



The Challenge

- Solving Schrödinger’s Equation is **hard**
- Usually turn to numerical approximations
- …but numerics have limitations
- Amount of data needed depends exponentially on d

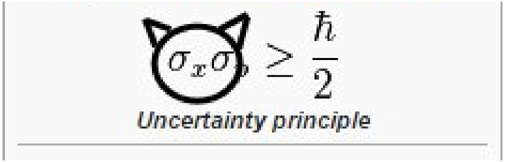
- *The Curse of Dimensionality*
- How do we improve dependency?
- More flexibility to our basis elements
- Machine Learning comes into play



The Challenge

- Solving Schrödinger's Equation is **hard**
- Usually turn to numerical approximations
- …but numerics have limitations
- Amount of data needed depends exponentially on d

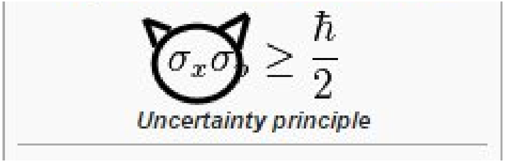
- *The Curse of Dimensionality*
- How do we improve dependency?
- More flexibility to our basis elements
- Machine Learning comes into play



The Challenge

- Solving Schrödinger's Equation is **hard**
- Usually turn to numerical approximations
- …but numerics have limitations
- Amount of data needed depends exponentially on d

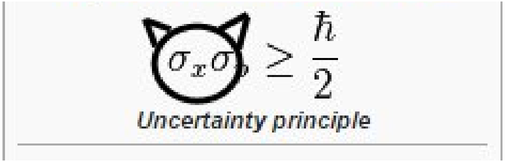
- *The Curse of Dimensionality*
- How do we improve dependency?
- More flexibility to our basis elements
- Machine Learning comes into play



The Challenge

- Solving Schrödinger's Equation is **hard**
- Usually turn to numerical approximations
- …but numerics have limitations
- Amount of data needed depends exponentially on d

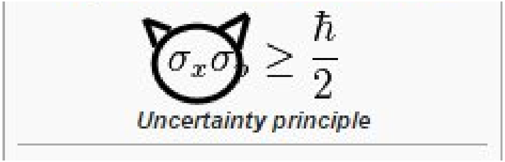
- *The Curse of Dimensionality*
- How do we improve dependency?
- More flexibility to our basis elements
- Machine Learning comes into play



The Challenge

- Solving Schrödinger’s Equation is **hard**
- Usually turn to numerical approximations
- …but numerics have limitations
- Amount of data needed depends exponentially on d

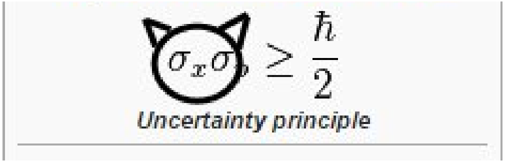
- *The Curse of Dimensionality*
- How do we improve dependency?
- More flexibility to our basis elements
- Machine Learning comes into play



The Challenge

- Solving Schrödinger’s Equation is **hard**
- Usually turn to numerical approximations
- …but numerics have limitations
- Amount of data needed depends exponentially on d

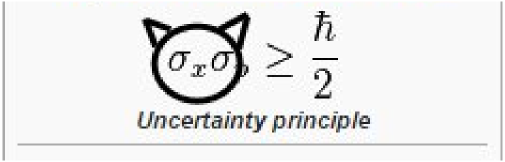
- *The Curse of Dimensionality*
- How do we improve dependency?
- More flexibility to our basis elements
- Machine Learning comes into play



The Challenge

- Solving Schrödinger's Equation is **hard**
- Usually turn to numerical approximations
- …but numerics have limitations
- Amount of data needed depends exponentially on d

- *The Curse of Dimensionality*
- How do we improve dependency?
- More flexibility to our basis elements
- Machine Learning comes into play



CC BY

Contents

Problem Setup

- Time Independent Schrödiger Equation (**TISE**)

$$E\,|\psi\rangle = \hat{H}\,|\psi\rangle$$

- We approximate it using the variational principle [Lib22]
- Fundamentally: We need loss function to optimize
- Energy of approximation state E_Θ is always $\geq$ than groundstate E_0

CC BY

Problem Setup

- Time Independent Schrödinger Equation (**TISE**)

$$E\,|\psi\rangle = \hat{H}\,|\psi\rangle$$

- We approximate it using the variational principle [Lib22]
- Fundamentally: We need loss function to optimize
- Energy of approximation state E_Θ is always $\geq$ than groundstate E_0

Problem Setup

- Time Independent Schrödiger Equation (**TISE**)

$$E\left|\psi\right\rangle = \hat{H}\left|\psi\right\rangle$$

- We approximate it using the variational principle [Lib22]
- Fundamentally: We need loss function to optimize
- Energy of approximation state E_Θ is always $\geq$ than groundstate E_0

Proof.

Assuming normalization of $\left|\psi_\Theta\right\rangle$ and letting $\{\left|k\right\rangle\}_{k=1}^{\infty}$ be eigenstates of $\hat{H}$

$$E_\Theta = \left\langle\psi_\Theta\right|\hat{H}\left|\psi_\Theta\right\rangle = \sum_{m,n=0}^{\infty} \bar{\alpha}_m \alpha_n \underbrace{\left\langle m\right|\hat{H}\left|n\right\rangle}_{E_n \cdot \delta_{m,n}} = \sum_{n=0}^{\infty} |\alpha_n|^2 E_n \geq E_0 \sum_{n=0}^{\infty} |\alpha_n|^2 = E_0 \qquad (1)$$

Problem Setup

- Time Independent Schrödiger Equation (**TISE**)

$$E\,|\psi\rangle = \hat{H}\,|\psi\rangle$$

- We approximate it using the variational principle [Lib22]
- Fundamentally: We need loss function to optimize
- Energy of approximation state E_Θ is always $\geq$ than groundstate E_0

Proof.

Assuming normalization of $|\psi_\Theta\rangle$ and letting $\{|k\rangle\}_{k=1}^{\infty}$ be eigenstates of $\hat{H}$

$$E_\Theta = \langle\psi_\Theta|\,\hat{H}\,|\psi_\Theta\rangle = \sum_{m,n=0}^{\infty} \bar{\alpha}_m \alpha_n \underbrace{\langle m|\,\hat{H}\,|n\rangle}_{E_n\cdot\delta_{m,n}} = \sum_{n=0}^{\infty} |\alpha_n|^2 E_n \geq E_0 \sum_{n=0}^{\infty} |\alpha_n|^2 = E_0 \qquad (1)$$

Problem Setup

- Time Independent Schrödiger Equation (**TISE**)

$$E\,|\psi\rangle = \hat{H}\,|\psi\rangle$$

- We approximate it using the variational principle [Lib22]
- Fundamentally: We need loss function to optimize
- Energy of approximation state E_Θ is always $\geq$ than groundstate E_0

Proof.

Assuming normalization of $|\psi_\Theta\rangle$ and letting $\{|k\rangle\}_{k=1}^{\infty}$ be eigenstates of $\hat{H}$

$$E_\Theta = \langle\psi_\Theta|\,\hat{H}\,|\psi_\Theta\rangle = \sum_{m,n=0}^{\infty} \bar{\alpha}_m \alpha_n \underbrace{\langle m|\,\hat{H}\,|n\rangle}_{E_n\cdot\delta_{m,n}} = \sum_{n=0}^{\infty} |\alpha_n|^2 E_n \geq E_0 \sum_{n=0}^{\infty} |\alpha_n|^2 = E_0 \qquad (1)$$

Variational Principle

- This principle applies to higher order eigenenergies
- Idea: Consider orthogonal subspace to $|0\rangle$ and reapply Eq. (1)
- Can be done for all space (if it is separable)
- More rigorous approach in [Lib22]
- $\Rightarrow$ must diagonalize $[\widetilde{H}]_{ij} = \langle \varphi_i | \hat{H} | \varphi_j \rangle$ given arbitrary orthonormal basis $\{|\varphi_k\rangle\}_{k=1}^{\infty}$
- Can be an infinite dimensional matrix
- ∞ is a problem numerically

Variational Principle

- This principle applies to higher order eigenenergies
- Idea: Consider orthogonal subspace to $|0\rangle$ and reapply Eq. (1)
- Can be done for all space (if it is separable)
- More rigorous approach in [Lib22]
- $\Rightarrow$ must diagonalize $[\widetilde{H}]_{ij} = \langle\varphi_i|\,\hat{H}\,|\varphi_j\rangle$ given arbitrary orthonormal basis $\{|\varphi_k\rangle\}_{k=1}^{\infty}$
- Can be an infinite dimensional matrix
- ∞ is a problem numerically

Variational Principle

- This principle applies to higher order eigenenergies
- Idea: Consider orthogonal subspace to $|0\rangle$ and reapply Eq. (1)
- Can be done for all space (if it is separable)
- More rigorous approach in [Lib22]
- $\Rightarrow$ must diagonalize $[\widetilde{H}]_{ij} = \langle \varphi_i | \hat{H} | \varphi_j \rangle$ given arbitrary orthonormal basis $\{|\varphi_k\rangle\}_{k=1}^{\infty}$
- Can be an infinite dimensional matrix
- ∞ is a problem numerically

cc BY

Variational Principle

- This principle applies to higher order eigenenergies
- Idea: Consider orthogonal subspace to $|0\rangle$ and reapply Eq. (1)
- Can be done for all space (if it is separable)
- More rigorous approach in [Lib22]
- $\Rightarrow$ must diagonalize $[\widetilde{H}]_{ij} = \langle\varphi_i|\,\hat{H}\,|\varphi_j\rangle$ given arbitrary orthonormal basis $\{|\varphi_k\rangle\}_{k=1}^{\infty}$
- Can be an infinite dimensional matrix
- ∞ is a problem numerically

Variational Principle

- This principle applies to higher order eigenenergies
- Idea: Consider orthogonal subspace to $|0\rangle$ and reapply Eq. (1)
- Can be done for all space (if it is separable)
- More rigorous approach in [Lib22]
- $\Rightarrow$ must diagonalize $[\widetilde{H}]_{ij} = \langle\varphi_i|\,\hat{H}\,|\varphi_j\rangle$ given arbitrary orthonormal basis $\{|\varphi_k\rangle\}_{k=1}^{\infty}$
- Can be an infinite dimensional matrix
- ∞ is a problem numerically

CC BY

Variational Principle

- This principle applies to higher order eigenenergies
- Idea: Consider orthogonal subspace to $|0\rangle$ and reapply Eq. (1)
- Can be done for all space (if it is separable)
- More rigorous approach in [Lib22]
- $\Rightarrow$ must diagonalize $[\widetilde{H}]_{ij} = \langle\varphi_i|\,\hat{H}\,|\varphi_j\rangle$ given arbitrary orthonormal basis $\{|\varphi_k\rangle\}_{k=1}^{\infty}$
- Can be an infinite dimensional matrix
- ∞ is a problem numerically

Variational Principle

- This principle applies to higher order eigenenergies
- Idea: Consider orthogonal subspace to $|0\rangle$ and reapply Eq. (1)
- Can be done for all space (if it is separable)
- More rigorous approach in [Lib22]
- $\Rightarrow$ must diagonalize $[\widetilde{H}]_{ij} = \langle\varphi_i|\,\hat{H}\,|\varphi_j\rangle$ given arbitrary orthonormal basis $\{|\varphi_k\rangle\}_{k=1}^{\infty}$
- Can be an infinite dimensional matrix
- ∞ is a problem numerically

cc BY

Managing ∞

- Define a truncation parameter N_{max}
- Approximate state as

$$|\psi\rangle \approx \sum_{k=0}^{N_{\text{max}}} c_k |\varphi_k\rangle$$

- Becomes finite-dimensional problem
- Recall: Curse of dimensionality!
- N_{max} needed to converge to real results scales *exponentially* with d
- Approach: define a more flexible ‘augmented basis’ $\{|\varphi_i^A\rangle\}_i$
- Reduce N_{max} needed when augmented basis is optimized

Managing ∞

- Define a truncation parameter N_{max}
- Approximate state as

$$|\psi\rangle \approx \sum_{k=0}^{N_{\text{max}}} c_k |\varphi_k\rangle$$

- Becomes finite-dimensional problem
- Recall: Curse of dimensionality!
- N_{max} needed to converge to real results scales *exponentially* with d
- Approach: define a more flexible 'augmented basis' $\{|\varphi_i^A\rangle\}_i$
- Reduce N_{max} needed when augmented basis is optimized

cc BY

Managing ∞

- Define a truncation parameter N_{max}
- Approximate state as

$$|\psi\rangle \approx \sum_{k=0}^{N_{\text{max}}} c_k \, |\varphi_k\rangle$$

- Becomes finite-dimensional problem
- Recall: Curse of dimensionality!
- N_{max} needed to converge to real results scales *exponentially* with d
- Approach: define a more flexible 'augmented basis' $\{|\varphi_i^A\rangle\}_i$
- Reduce N_{max} needed when augmented basis is optimized

Managing ∞

- Define a truncation parameter N_{max}
- Approximate state as

$$|\psi\rangle \approx \sum_{k=0}^{N_{\text{max}}} c_k |\varphi_k\rangle$$

- Becomes finite-dimensional problem
- Recall: Curse of dimensionality!
- N_{max} needed to converge to real results scales *exponentially* with d
- Approach: define a more flexible 'augmented basis' $\{|\varphi_i^A\rangle\}_i$
- Reduce N_{max} needed when augmented basis is optimized

cc BY

Managing ∞

- Define a truncation parameter N_{max}
- Approximate state as

$$|\psi\rangle \approx \sum_{k=0}^{N_{\text{max}}} c_k \, |\varphi_k\rangle$$

- Becomes finite-dimensional problem
- Recall: Curse of dimensionality!
- N_{max} needed to converge to real results scales *exponentially* with d
- Approach: define a more flexible 'augmented basis' $\{|\varphi_i^A\rangle\}_i$
- Reduce N_{max} needed when augmented basis is optimized

cc BY

Managing ∞

- Define a truncation parameter N_{max}
- Approximate state as

$$|\psi\rangle \approx \sum_{k=0}^{N_{\text{max}}} c_k \, |\varphi_k\rangle$$

- Becomes finite-dimensional problem
- Recall: Curse of dimensionality!
- N_{max} needed to converge to real results scales *exponentially* with d
- Approach: define a more flexible 'augmented basis' $\{|\varphi_i^A\rangle\}_i$
- Reduce N_{max} needed when augmented basis is optimized

CC BY

Managing ∞

- Define a truncation parameter N_{max}
- Approximate state as

$$|\psi\rangle \approx \sum_{k=0}^{N_{\text{max}}} c_k \, |\varphi_k\rangle$$

- Becomes finite-dimensional problem
- Recall: Curse of dimensionality!
- N_{max} needed to converge to real results scales *exponentially* with d
- Approach: define a more flexible 'augmented basis' $\{|\varphi_i^A\rangle\}_i$
- Reduce N_{max} needed when augmented basis is optimized

CC BY

Contents

Role of ML

- Will start considering coordinate space: $|\psi\rangle \mathrel{\hat{=}} \psi(x)$
- Define augmented basis as:

$$\varphi_k^A(x) = \varphi_k(g(x)) \cdot \sqrt{\det\left|\frac{\mathrm{d}g}{\mathrm{d}x}\right|} \tag{2}$$

- g is a **Normalizing Flow**
- This preserves orthonormality

Proof:

Role of ML

- Will start considering coordinate space: $|\psi\rangle \mathrel{\hat{=}} \psi(x)$
- Define augmented basis as:

$$\varphi_k^A(x) = \varphi_k(g(x)) \cdot \sqrt{\det\left|\frac{\mathrm{d}g}{\mathrm{d}x}\right|} \tag{2}$$

- g is a **Normalizing Flow**
- This preserves orthonormality

Role of ML

- Will start considering coordinate space: $|\psi\rangle \mathrel{\hat{=}} \psi(x)$
- Define augmented basis as:

$$\varphi_k^A(x) = \varphi_k(g(x)) \cdot \sqrt{\det\left|\frac{\mathrm{d}g}{\mathrm{d}x}\right|} \tag{2}$$

- g is a **Normalizing Flow**
- This preserves orthonormality

Proof

$$\langle\varphi_k^A|\varphi_k^A\rangle = \int \mathrm{d}x \varphi_i(g(x))\varphi_j(g(x)) \cdot \det\left|\frac{\mathrm{d}g}{\mathrm{d}x}\right| = \int \mathrm{d}g \left|\frac{\mathrm{d}x}{\mathrm{d}g}\right| \varphi_i(g)\varphi_j(g) \cdot \det\left|\frac{\mathrm{d}g}{\mathrm{d}x}\right| = \delta_{i,j}$$

Role of ML

- Will start considering coordinate space: $|\psi\rangle \hat{=} \psi(x)$
- Define augmented basis as:

$$\varphi_k^A(x) = \varphi_k(g(x)) \cdot \sqrt{\det \left| \frac{\mathrm{d}g}{\mathrm{d}x} \right|} \tag{2}$$

- g is a **Normalizing Flow**
- This preserves orthonormality

Proof.

$$\langle \varphi_k^A | \varphi_k^A \rangle = \int \mathrm{d}x \varphi_i(g(x)) \varphi_j(g(x)) \cdot \det \left| \frac{\mathrm{d}g}{\mathrm{d}x} \right| = \int \mathrm{d}g \left| \frac{\mathrm{d}x}{\mathrm{d}g} \right| \varphi_i(g) \varphi_j(g) \cdot \det \left| \frac{\mathrm{d}g}{\mathrm{d}x} \right| = \delta_{i,j}$$

Role of ML

- Will start considering coordinate space: $|\psi\rangle \mathrel{\hat{=}} \psi(x)$
- Define augmented basis as:

$$\varphi_k^A(x) = \varphi_k(g(x)) \cdot \sqrt{\det\left|\frac{\mathrm{d}g}{\mathrm{d}x}\right|} \tag{2}$$

- g is a **Normalizing Flow**
- This preserves orthonormality

Proof.

$$\langle\varphi_k^A|\varphi_k^A\rangle = \int \mathrm{d}x \varphi_i(g(x))\varphi_j(g(x)) \cdot \det\left|\frac{\mathrm{d}g}{\mathrm{d}x}\right| = \int \mathrm{d}g \left|\frac{\mathrm{d}x}{\mathrm{d}g}\right| \varphi_i(g)\varphi_j(g) \cdot \det\left|\frac{\mathrm{d}g}{\mathrm{d}x}\right| = \delta_{i,j}$$

cc BY

Normalizing Flows

- Basic idea: A chain of diffeomorphisms
- Invertible and differentiable

$$z = (f_n \circ f_{n-1} \circ \cdots \circ f_1)(x)$$

- Several different paradigms
- Need to find which ones best improve flexibility of basis states
- We concentrate on RNVP

Normalizing Flows

- Basic idea: A chain of diffeomorphisms
- Invertible and differentiable

$$z = (f_n \circ f_{n-1} \circ \cdots \circ f_1)(x)$$

- Several different paradigms
- Need to find which ones best improve flexibility of basis states
- We concentrate on RNVP

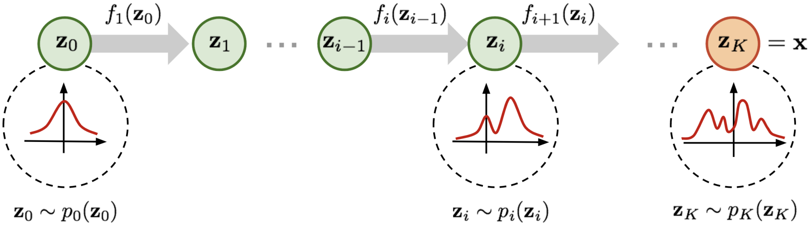


Figure: Normalizing Flow [Lip22]

Normalizing Flows

- Basic idea: A chain of diffeomorphisms
- Invertible and differentiable

$$z = (f_n \circ f_{n-1} \circ \cdots \circ f_1)(x)$$

- Several different paradigms
- Need to find which ones best improve flexibility of basis states
- We concentrate on RNVP

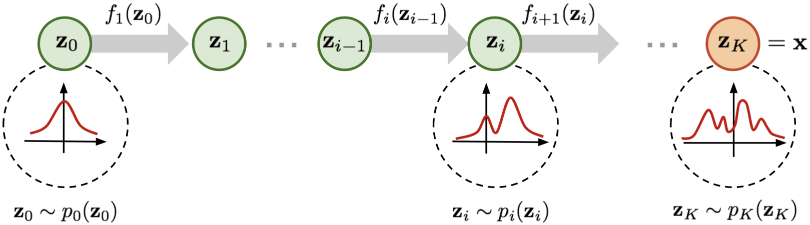


Figure: Normalizing Flow [Lip22]

Normalizing Flows

- Basic idea: A chain of diffeomorphisms
- Invertible and differentiable

$$z = (f_n \circ f_{n-1} \circ \cdots \circ f_1)(x)$$

- Several different paradigms
- Need to find which ones best improve flexibility of basis states
- We concentrate on RNVP

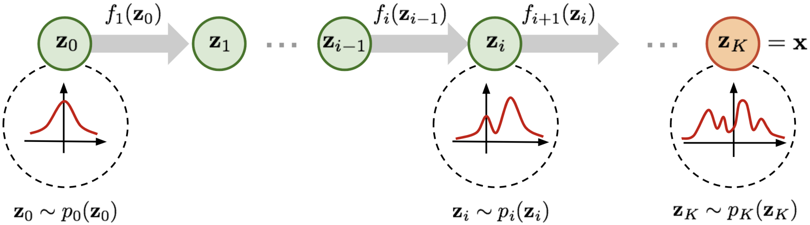


Figure: Normalizing Flow [Lip22]

Normalizing Flows

> Basic idea: A chain of diffeomorphisms

> Invertible and differentiable

$$z = (f_n \circ f_{n-1} \circ \cdots \circ f_1)(x)$$

> Several different paradigms

> Need to find which ones best improve flexibility of basis states

> We concentrate on RNVP

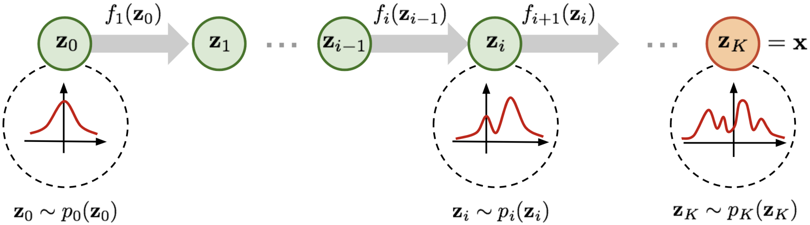


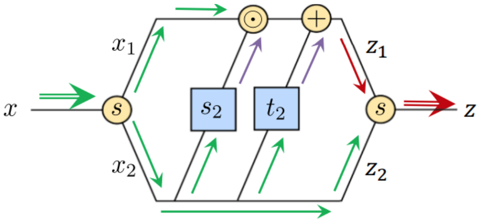
Figure: Normalizing Flow [Lip22]

RNVP

- The Real-valued Non-Volume-Preserving model
- Let $\mathcal{P}_k$ be a projection over half of the basis vectors and $\mathcal{Q}_k \equiv \mathbb{1} - \mathcal{P}_k$
- layer $g_k(x)$ is given by

$$
\begin{aligned}
g_k(x) &= \mathcal{P}_k[x] + \mathcal{Q}_k[f_k(x)] \quad \text{with} \\
f_k(x) &= e^{s_k(\mathcal{P}_k[x])} \odot x + t_k(\mathcal{P}_k[x])
\end{aligned}
\tag{3}
$$

- Inverse can be shown rigorously



$$z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} x_1 \cdot s_2(x_2) + t_2(x_2) \\ x_2 \end{bmatrix}$$

inverse is equally efficient:

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} (z_1 - t_2(z_2))/s(z_2) \\ z_2 \end{bmatrix}$$

Figure: Basic layer of RNVP [Köt21]

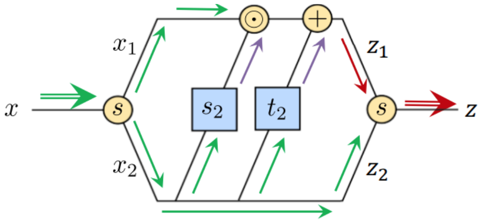
CC BY

RNVP

- The Real-valued Non-Volume-Preserving model
- Let $\mathcal{P}_k$ be a projection over half of the basis vectors and $\mathcal{Q}_k \equiv \mathbb{1} - \mathcal{P}_k$
- layer $g_k(x)$ is given by

$$g_k(x) = \mathcal{P}_k[x] + \mathcal{Q}_k[f_k(x)] \quad \text{with} \\ f_k(x) = e^{s_k(\mathcal{P}_k[x])} \odot x + t_k(\mathcal{P}_k[x]) \tag{3}$$

- Inverse can be shown rigorously



$$z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} x_1 \cdot s_2(x_2) + t_2(x_2) \\ x_2 \end{bmatrix}$$

inverse is equally efficient:

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} (z_1 - t_2(z_2))/s(z_2) \\ z_2 \end{bmatrix}$$

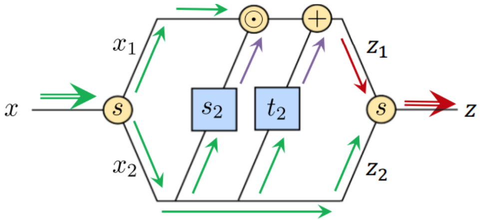
Figure: Basic layer of RNVP [Köt21]

RNVP

- The Real-valued Non-Volume-Preserving model
- Let $\mathcal{P}_k$ be a projection over half of the basis vectors and $\mathcal{Q}_k \equiv \mathbb{1} - \mathcal{P}_k$
- layer $g_k(x)$ is given by

$$
\begin{aligned}
g_k(x) &= \mathcal{P}_k[x] + \mathcal{Q}_k[f_k(x)] \quad \text{with} \\
f_k(x) &= e^{s_k(\mathcal{P}_k[x])} \odot x + t_k(\mathcal{P}_k[x])
\end{aligned}
\tag{3}
$$

- Inverse can be shown rigorously



$$z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} x_1 \cdot s_2(x_2) + t_2(x_2) \\ x_2 \end{bmatrix}$$

inverse is equally efficient:

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} (z_1 - t_2(z_2))/s(z_2) \\ z_2 \end{bmatrix}$$

Figure: Basic layer of RNVP [Köt21]

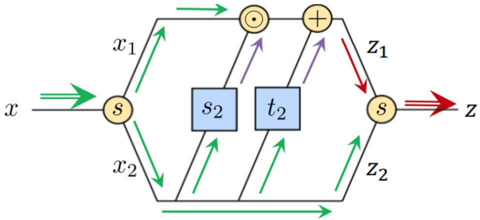
CC BY

RNVP

- The Real-valued Non-Volume-Preserving model
- Let $\mathcal{P}_k$ be a projection over half of the basis vectors and $\mathcal{Q}_k \equiv \mathbb{1} - \mathcal{P}_k$
- layer $g_k(x)$ is given by

$$\begin{aligned} g_k(x) &= \mathcal{P}_k[x] + \mathcal{Q}_k[f_k(x)] \quad \text{with} \\ f_k(x) &= e^{s_k(\mathcal{P}_k[x])} \odot x + t_k(\mathcal{P}_k[x]) \end{aligned} \tag{3}$$

- Inverse can be shown rigorously



$$z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} x_1 \cdot s_2(x_2) + t_2(x_2) \\ x_2 \end{bmatrix}$$

inverse is equally efficient:

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} (z_1 - t_2(z_2))/s(z_2) \\ z_2 \end{bmatrix}$$

Figure: Basic layer of RNVP [Köt21]

Contents

Convergence

How do we know $\sum_{k=0}^{N_{\text{max}}} c_k \varphi_k^A$ converges to the real result as $N_{\text{max}} \to \infty$?

- Let g be the normalizing flow
- Analogous to showing $f \circ g \in \mathcal{S} \ \forall f \in \mathcal{S}$
- $\mathcal{S}$ are rapidly decreasing, infinitely differentiable functions
-

$$= \left\{ f \in C^\infty \mid \|x^\beta \cdot \frac{\partial^\alpha}{\partial \alpha}\| < \infty \right\}$$

- If g is RNVP (with a slight modification), then $f \circ g \in \mathcal{S}$
- For now only in 1 dimensional case
- *Proof left as an exercise to the reader*

Convergence

How do we know $\sum_{k=0}^{N_{\text{max}}} c_k \varphi_k^A$ converges to the real result as $N_{\text{max}} \to \infty$?

- Let g be the normalizing flow
- Analogous to showing $f \circ g \in \mathcal{S} \; \forall f \in \mathcal{S}$
- $\mathcal{S}$ are rapidly decreasing, infinitely differentiable functions
-

$$= \left\{ f \in C^{\infty} \mid \|x^{\beta} \cdot \frac{\partial^{\alpha}}{\partial \alpha}\| < \infty \right\}$$

- If g is RNVP (with a slight modification), then $f \circ g \in \mathcal{S}$
- For now only in 1 dimensional case
- *Proof left as an exercise to the reader*

Convergence

How do we know $\sum_{k=0}^{N_{\text{max}}} c_k \varphi_k^A$ converges to the real result as $N_{\text{max}} \to \infty$?

- Let g be the normalizing flow
- Analogous to showing $f \circ g \in \mathcal{S}\ \forall f \in \mathcal{S}$
- $\mathcal{S}$ are rapidly decreasing, infinitely differentiable functions
-

$$\equiv \left\{ f \in C^{\infty} \mid \|x^{\beta} \cdot \frac{\partial^{\alpha}}{\partial^{\alpha}}\| < \infty \right\}$$

- If g is RNVP (with a slight modification), then $f \circ g \in \mathcal{S}$
- For now only in 1 dimensional case
- *Proof left as an exercise to the reader*

cc BY

Convergence

How do we know $\sum_{k=0}^{N_{\text{max}}} c_k \varphi_k^A$ converges to the real result as $N_{\text{max}} \to \infty$?

- Let g be the normalizing flow
- Analogous to showing $f \circ g \in \mathcal{S}\ \forall f \in \mathcal{S}$
- $\mathcal{S}$ are rapidly decreasing, infinitely differentiable functions
-

$$\equiv \left\{ f \in C^{\infty} \mid \|x^{\beta} \cdot \frac{\partial^{\alpha}}{\partial^{\alpha}}\| < \infty \right\}$$

- If g is RNVP (with a slight modification), then $f \circ g \in \mathcal{S}$
- For now only in 1 dimensional case
- *Proof left as an exercise to the reader*

cc BY

Convergence

How do we know $\sum_{k=0}^{N_{\text{max}}} c_k \varphi_k^A$ converges to the real result as $N_{\text{max}} \to \infty$?

- Let g be the normalizing flow
- Analogous to showing $f \circ g \in \mathcal{S} \; \forall f \in \mathcal{S}$
- $\mathcal{S}$ are rapidly decreasing, infinitely differentiable functions
- $$\equiv \left\{ f \in C^\infty \mid \|x^\beta \cdot \frac{\partial^\alpha}{\partial^\alpha}\| < \infty \right\}$$
- If g is RNVP (with a slight modification), then $f \circ g \in \mathcal{S}$
- For now only in 1 dimensional case
- *Proof left as an exercise to the reader*

cc BY

Convergence

How do we know $\sum_{k=0}^{N_{\text{max}}} c_k \varphi_k^A$ converges to the real result as $N_{\text{max}} \to \infty$?

- Let g be the normalizing flow
- Analogous to showing $f \circ g \in \mathcal{S} \; \forall f \in \mathcal{S}$
- $\mathcal{S}$ are rapidly decreasing, infinitely differentiable functions
- $$\equiv \left\{ f \in C^\infty \mid \|x^\beta \cdot \frac{\partial^\alpha}{\partial^\alpha}\| < \infty \right\}$$
- If g is RNVP (with a slight modification), then $f \circ g \in \mathcal{S}$
- For now only in 1 dimensional case
- *Proof left as an exercise to the reader*

cc BY

Convergence

How do we know $\sum_{k=0}^{N_{\text{max}}} c_k \varphi_k^A$ converges to the real result as $N_{\text{max}} \to \infty$?

- Let g be the normalizing flow
- Analogous to showing $f \circ g \in \mathcal{S} \ \forall f \in \mathcal{S}$
- $\mathcal{S}$ are rapidly decreasing, infinitely differentiable functions
- $$\equiv \left\{ f \in C^{\infty} \mid \| x^{\beta} \cdot \frac{\partial^{\alpha}}{\partial^{\alpha}} \| < \infty \right\}$$
- If g is RNVP (with a slight modification), then $f \circ g \in \mathcal{S}$
- For now only in 1 dimensional case
- *Proof left as an exercise to the reader*

Contents

Performance?

> We compared results in H2S molecule

> Normalizing Flows greatly improved stretching case (Fig. 3)

> RNVP behaves similarly to IResNet

> Possibly perform better on higher dimensional data

> Outperform no-neural-network (LIN)

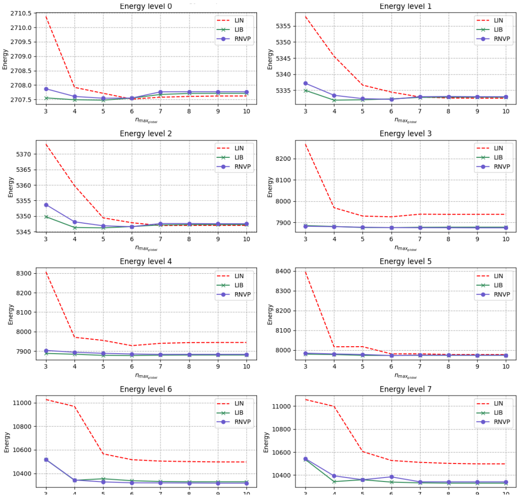


Figure: Stretching case for H2S molecule

CC BY

Performance?

> We compared results in H2S molecule

> Normalizing Flows greatly improved stretching case (Fig. 3)

> RNVP behaves similarly to IResNet

> Possibly perform better on higher dimensional data

> Outperform no-neural-network (LIN)

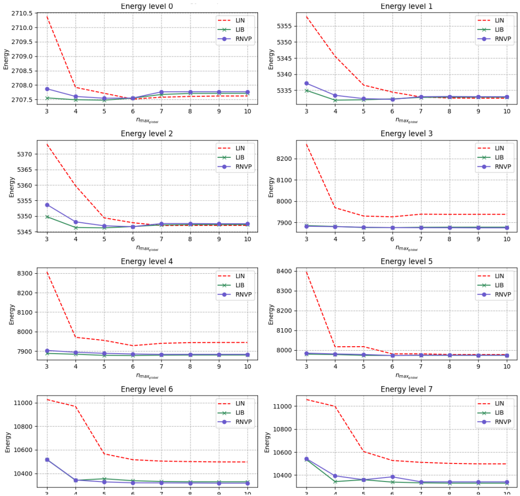


Figure: Stretching case for H2S molecule

cc BY

Performance?

> We compared results in H2S molecule

> Normalizing Flows greatly improved stretching case (Fig. 3)

> RNVP behaves similarly to IResNet

> Possibly perform better on higher dimensional data

> Outperform no-neural-network (LIN)

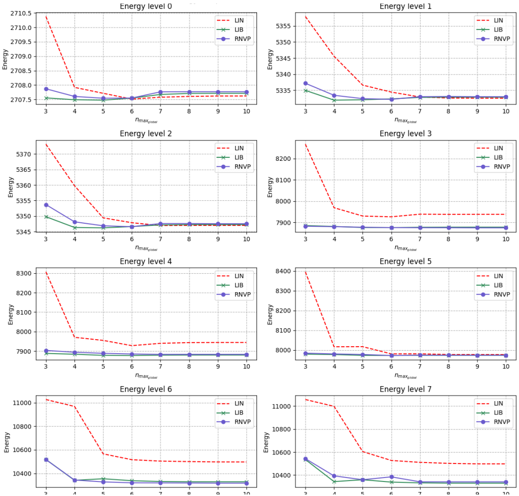


Figure: Stretching case for H2S molecule

Performance?

> We compared results in $H2S$ molecule

> Normalizing Flows greatly improved stretching case (Fig. 3)

> RNVP behaves similarly to IResNet

> Possibly perform better on higher dimensional data

> Outperform no-neural-network (LIN)

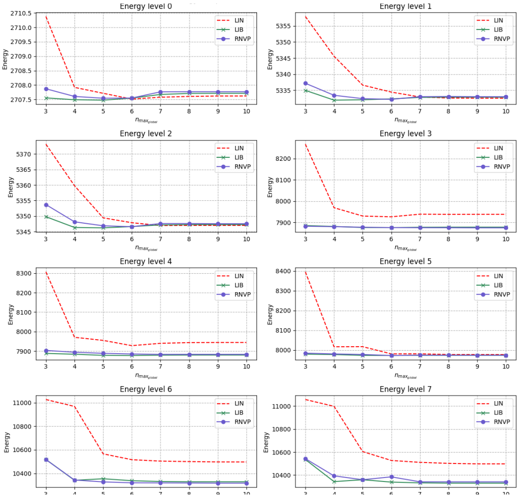


Figure: Stretching case for H2S molecule

Performance?

- We compared results in H2S molecule
- Normalizing Flows greatly improved stretching case (Fig. 3)
- RNVP behaves similarly to IResNet
- Possibly perform better on higher dimensional data
- Outperform no-neural-network (LIN)

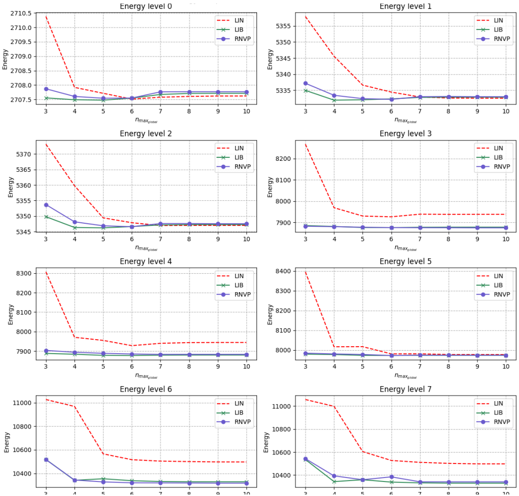
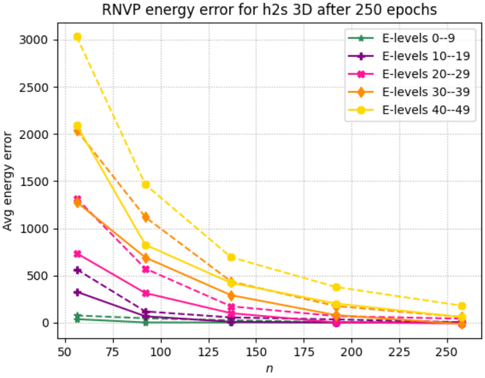


Figure: Stretching case for H2S molecule

Performance



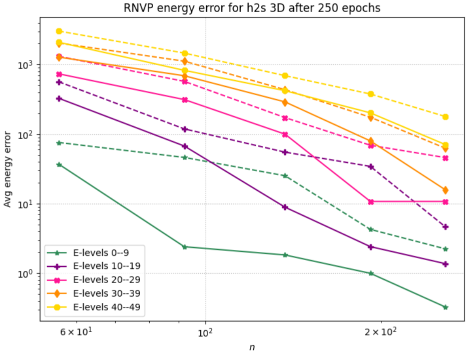
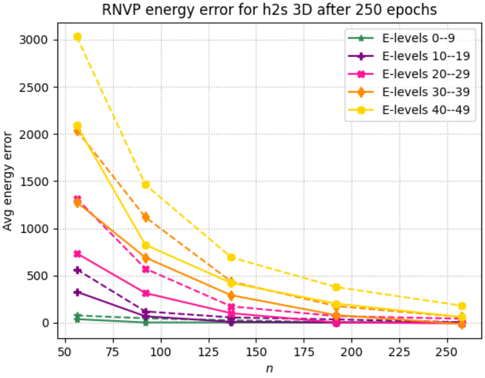


Figure: RNVP performance vs No neural network (- -)

> RNVP clearly outperformed on H2S

Performance



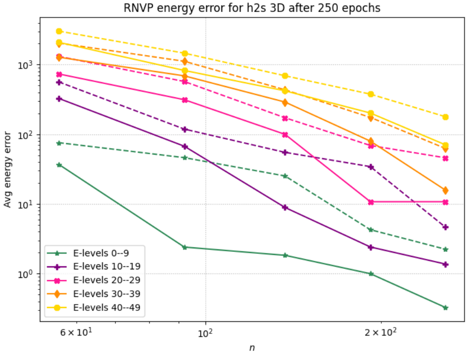
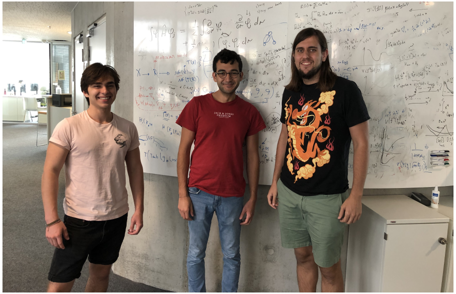


Figure: RNVP performance vs No neural network (- -)

> RNVP clearly outperformed on H2S

Thank you!



Contact

Deutsches Elektronen-Synchrotron DESY

www.desy.de

Nicolas Mendoza
0000–0001–5561–1392
CMI
snmendozav@gmail.com
+49–152–2768–2024

References

[KPB19] Ivan Kobyzev, Simon J. D. Prince, and Marcus A. Brubaker. “Normalizing Flows: An Introduction and Review of Current Methods”. In: (2019). DOI: 10.1109/TPAMI.2020.2992934. eprint: arXiv:1908.09257.

[Köt21] Ullrich Köthe. Introduction to Normalizing Flows. Mar. 2021.

[Lib22] Libretexts. 7.2: Linear variational method and the secular determinant. July 2022.

[Lip22] Phillip Lippe. Tutorial 11: Normalizing flows for image modeling. 2022. URL: https://uvadlc-notebooks.readthedocs.io/en/latest/tutorial_notebooks/tutorial11/NF_image_modeling.html.

cc BY