Fundamental Particles & Interactions

Illustration: Typoform

Gravity Force

Electromagnetic force

Strong force

Weak force

Fundamental forces:
- Electromagnetic force
- Strong force
- Weak force

Quarks:
- Up quark
- Down quark
- Strange quark
- Charm quark
- Bottom quark
- Top quark

Mesons:
- Protons
- Neutrons

Nuclei:
- Protons
- Neutrons
- Electrons

Forces:
- Electromagnetic force
- Strong force
- Weak force

Particles:
- Photons
- Quarks
- Leptons

Interactions:
- Neutron decay
- Beta decay
- Neutrino interactions
- Burning of the sun
\[ \mathcal{L}_{SM} = \]
\[ \frac{1}{4} W_{\mu \nu} \cdot W^{\mu \nu} - \frac{1}{4} B_{\mu \nu} B^{\mu \nu} - \frac{1}{4} G_{\mu \nu}^a G^{\mu \nu}_a \]
\[ \text{kinetic energies and self-interactions of the gauge bosons} \]
\[ + \hat{L} \gamma^\mu (i \partial_\mu - \frac{1}{2} g \tau \cdot W_\mu - \frac{1}{2} g' Y B_\mu) L + \hat{R} \gamma^\mu (i \partial_\mu - \frac{1}{2} g' Y B_\mu) R \]
\[ \text{kinetic energies and electroweak interactions of fermions} \]
\[ + \frac{1}{2} \left| (i \partial_\mu - \frac{1}{2} g \tau \cdot W_\mu - \frac{1}{2} g' Y B_\mu) \phi \right|^2 - V(\phi) \]
\[ W^\pm, Z, \gamma \text{ and Higgs masses and couplings} \]
\[ + g'' q_T a \bar{q} q \]
\[ \text{interactions between quarks and gluons} \]
\[ + (G_1 \hat{L} \phi R + G_2 \hat{R} \phi L + h.c.) \]
\[ \text{fermion masses and couplings to Higgs} \]
Interaction strength depends on energy

\[ \alpha_s(Q^2) = \alpha_s(0) + \alpha_s(1) s + \alpha_s(2) s^2 + \ldots \]

Low-energy QCD is non-perturbative

\[ QCD \alpha_s(M_Z) = 0.1177 \pm 0.0013 \]
The strong force: Quantum Chromodynamics

Interaction strength depends on energy

\[ \alpha_s(Q^2) = O_0 + O_1 + O_2 + \ldots \]

QCD is weak at high-energies, small coupling, perturbation theory works

\( QCD \alpha_s(M_Z) = 0.1177 \pm 0.0013 \)
QCD is strong at low-energies, no small coupling, perturbation theory fails.

Emergent phenomena: protons, pions, etc.

QCD is weak at high-energies, small coupling, perturbation theory works.
Like the silicon chips of more recent years, the Feynman diagram was bringing computation to the masses.

—JULIAN SCHWINGER
Simulating particle physics processes

Theory parameters $\theta$
Simulating particle physics processes

Latent variables

Parton-level momenta

Theory parameters

Evolution
Simulating particle physics processes

Latent variables

Shower splittings
\( z_s \)

Parton-level momenta
\( z_p \)

Theory parameters
\( \theta \)

Evolution

[F. Krauss]
Simulating particle physics processes

Latent variables

Detector interactions \( z_d \)

Shower splittings \( z_s \)

Parton-level momenta \( z_p \)

Theory parameters \( \theta \)

Evolution

[CMS]
Simulating particle physics processes

Latent variables

Observables

Detector interactions

Shower splittings

Parton-level momenta

Theory parameters

Evolution

$x$  $z_d$  $z_s$  $z_p$  $\theta$
**Simulating particle physics processes**

<table>
<thead>
<tr>
<th>Observables</th>
<th>Detector interactions</th>
<th>Shower splittings</th>
<th>Parton-level momenta</th>
<th>Theory parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x$</td>
<td>$z_d$</td>
<td>$z_s$</td>
<td>$z_p$</td>
<td>$\theta$</td>
</tr>
<tr>
<td>Sample from</td>
<td>$p(x</td>
<td>z_d)$</td>
<td>$p(z_d</td>
<td>z_s)$</td>
</tr>
</tbody>
</table>

**Prediction (simulation)**
Simulating particle physics processes

\[
p(x|\theta) = \int dz_d \int dz_s \int dz_p \ p(x|z_d) \quad p(z_d|z_s) \quad p(z_s|z_p) \quad p(z_p|\theta)
\]

Inference
Simulating particle physics processes

It's infeasible to calculate the integral over this enormous space!
Simulators are causal, generative models of the data generating process
Science is replete with high-fidelity simulators

The expressiveness of programming languages facilitates the development of complex, high-fidelity simulations, and the power of modern computing provides the ability to generate synthetic data from them.

Science is replete with high-fidelity simulators

Unfortunately, these simulators are poorly suited for statistical inference.
A toy example

Imagine the entire board is slightly tilted, which biases the probability to bounce left/right.

Say we want to infer $\theta$, the probability to bounce right based on distribution of $x$. 
A toy example

Imagine the entire board is slightly tilted, which biases the probability to bounce left/right.

Say we want to infer $\theta$, the probability to bounce right based on distribution of $x$. 
A toy example

Imagine the entire board is slightly tilted, which biases the probability to bounce left/right.

Say we want to infer $\theta$, the probability to bounce right based on distribution of $x$. 
A toy example

Imagine the entire board is slightly tilted, which biases the probability to bounce left/right.

Say we want to infer $\theta$, the probability to bounce right based on distribution of $x$. 
A toy example

Imagine the entire board is slightly tilted, which biases the probability to bounce left/right.

Say we want to infer $\theta$, the probability to bounce right based on distribution of $x$. 

**A toy example**

Imagine the entire board is slightly tilted, which biases the probability to bounce left/right.

Say we want to infer $\theta$, the probability to bounce right based on distribution of $x$. 
A toy example

Imagine the entire board is slightly tilted, which biases the probability to bounce left/right.

Say we want to infer $\theta$, the probability to bounce right based on distribution of $x$.
A toy example

Imagine the entire board is slightly tilted, which biases the probability to bounce left/right.

Say we want to infer $\theta$, the probability to bounce right based on distribution of $x$
A toy example

Imagine the entire board is slightly tilted, which biases the probability to bounce left/right.

Say we want to infer $\theta$, the probability to bounce right based on distribution of $x$.

The probability of ending in bin $x$ corresponds to the total probability of all the paths $z$ from start to $x$.

$$p(x|\theta) = \int p(x, z|\theta)dz = \binom{n}{x} \theta^x (1-\theta)^{n-x}$$
Uh oh!

The actual situation is much more complicated.

It’s not a Binomial distribution!

What is it?
Uh oh!

The actual situation is much more complicated.

It’s not a Binomial distribution!

What is it?
Uh oh!

The actual situation is much more complicated.

It’s not a Binomial distribution!

What is it?

I have no idea, but I can simulate it!
Properties of simulators

Two broad classes:

• Deterministic evolution of initial state
  • (eg. differential equations, fluid dynamics, N-body simulations, etc.)

• Stochastic evolution
  • (eg. Markov processes, molecular dynamics, Gibbs / Boltzmann distribution in statistical mechanics, stochastic differential equations, etc.)

Integral over latent variables is typically intractable

\[ p(x|\theta) = \int p(x, z | \theta)dz \]
Two broad classes:

- **Deterministic evolution of initial state**
  - (eg. differential equations, fluid dynamics, N-body simulations, etc.)
- **Stochastic evolution**
  - (eg. Markov processes, molecular dynamics, Gibbs / Boltzmann distribution in statistical mechanics, stochastic differential equations, etc.)

Integral over latent variables is typically intractable

\[
p(x|\theta) = \int p(x, z | \theta)dz
\]
An example

The probability of landing in a bin $x$ corresponds to cumulative probability of all the latent paths $z$ that end in $x$

$$p(x|\theta) = \int p(x, z | \theta) dz$$

- But the integral (sum) can no longer be simplified analytically
- As the latent space grows, the number of possible paths grows rapidly.
- The integral becomes **intractable**
- But generating synthetic observations remains easy
An example

The probability of landing in a bin $x$ corresponds to cumulative probability of all the latent paths $z$ that end in $x$

$$p(x|\theta) = \int p(x, z | \theta) dz$$

- But the integral (sum) can no longer be simplified analytically
- As the latent space grows, the number of possible paths grows rapidly.
- The integral becomes **intractable**
- But generating synthetic observations remains easy
A rose by any other name

This motivates a class of inference methods for a stochastic simulator where

• evaluating the likelihood is intractable, but

• it is possible to sample synthetic data $x \sim p(x | \theta)$

This setting is often referred to as likelihood-free inference, but I prefer the term simulation-based inference because usually one approximates the likelihood (or likelihood ratio) and then use established inference techniques

• applies to both Bayesian or Frequentist inference
Gold mining: augmenting the training data

Sample efficiency is a major concern for these methods as many simulators are computationally expensive
Gold mining: augmenting the training data

Sample efficiency is a major concern for these methods as many simulators are computationally expensive.

Recently, we realized we can **extract more from the simulator**. We can use **augmented data** to improve training.
Gold mining: augmenting the training data

Sample efficiency is a major concern for these methods as many simulators are computationally expensive.

Recently, we realized we can extract more from the simulator. We can use augmented data to improve training.

While implicit density is intractable

$$p(x|\theta) = \int dz p(x, z|\theta)$$

We can augment the simulator to calculate some quantities conditioned on latent $z$, which are tractable:

Joint likelihood ratio:

$$r(x, z|\theta_0, \theta_1) = \frac{p(x, z|\theta_0)}{p(x, z|\theta_1)}$$

and joint score:

$$t(x, z|\theta_0) = \frac{\nabla_\theta p(x, z|\theta)|_{\theta_0}}{p(x, z|\theta_0)} = \nabla_\theta \log p(x, z|\theta)|_{\theta_0}$$
Gold mining: augmenting the training data

The augmented training data converts supervised classification into supervised regression with lower variance

- improvement in training efficiency
Gold mining: augmenting the training data

The augmented training data converts supervised classification into supervised regression with lower variance

- improvement in training efficiency

Figure 5: Illustration of some key concepts with a one-dimensional Gaussian toy example. Left: classifiers trained to distinguish two sets of events generated from different hypotheses (green dots) converge to an optimal decision function \( s(x|\theta_0, \theta_1) \) given in Eq. (17). This lets us extract the likelihood ratio. Right: regression on the joint likelihood ratios \( r(x_e, z_e|\theta_0, \theta_1) \) of the simulated events (green dots) converges to the likelihood ratio \( r(x|\theta_0, \theta_1) \).

2. Particle-physics structure

As we have argued in Sec. II C, particle physics processes have a specific structure that allow us to extract additional information. Most processes satisfy the factorization of Eq. (2) with a tractable parton-level likelihood \( p(z|\theta) \). The generators do not only provide samples \( \{x_e\} \), but also the corresponding parton-level momenta (latent variables) \( \{z_e\} \) with \( (x_e, z_e) \sim p(x, z|\theta_0, \theta_1) \).

By evaluating the matrix elements at the generated momenta \( z_e \) for different hypotheses \( \theta_0 \) and \( \theta_1 \), we can extract the parton-level likelihood ratio \( p(z_e|\theta_0) / p(z_e|\theta_1) \). Since the distribution of \( x \) is conditionally independent of the theory parameters, this is the same as the joint likelihood ratio \( r(x_e, z_e|\theta_0, \theta_1) = p(x_e, z_{\text{detector}}, z_{\text{shower}}|\theta_0) p(x_e, z_{\text{detector}}, z_{\text{shower}}|\theta_1) = p(x_e|z_{\text{detector}}) p(x_e|z_{\text{shower}}) p(z_{\text{detector}}|z_{\text{shower}}) p(z_{\text{shower}}) p(z_e|\theta_0) p(z_e|\theta_1) \).

So while we cannot directly evaluate the likelihood ratio at the level of measured observables \( r(x|\theta_0, \theta_1) \), we can calculate the likelihood ratio for a generated event conditional on the latent parton-level momenta. The same is true for the score, i.e., the tangent vectors or relative change of the (log) likelihood under infinitesimal changes of the parameters of interest. While the score \( t(x_e|\theta_0) = r(\log p(x|\theta)|\theta_0) \).

New techniques require less data than without augmented data

Traditional Approach no NN

Estimation error vs. training sample size

**Impact on science: The Higgs boson**

Massive gains in precision of a flagship measurement at the LHC!

Equivalent to increasing data collected by LHC by several factors

(based on a 42-Dim observation $x$)

$-\sum \log (\theta, \theta'_{\text{true}})$

$1.16x$ larger $\Lambda$ reach

$1.9x$ more luminosity

$\tilde{C}_{\text{HD}}$ Profiled

$L = 300 \text{ fb}^{-1}$
Impact on science: The Higgs boson

Massive gains in precision of a flagship measurement at the LHC!

Equivalent to increasing data collected by LHC by several factors

(based on a 42-Dim observation $x$)
Normalizing flows in the ambient data space

\[ u \sim p_u(u) \]

\( d \)-dim. latent variables

invertible NN

tractable density over ambient data space

\[ p_x(x) = p_u(f^{-1}(x)) \cdot |\det J_f(f^{-1}(x))|^{-1} \]

[G. Papamakarios et al 1912.02762]
Dynamical systems like

- the Lorenz attractor
- Attractor networks in theoretical Neuroscience

\[
\frac{dx_0}{dt} = \sigma(x_1 - x_0), \quad \frac{dx_1}{dt} = x_0(\rho - x_2) - x_1, \quad \frac{dx_2}{dt} = x_0 x_1 - \beta x_2.
\]
Why the data lives on a manifold

Dynamical systems like

- the Lorenz attractor
- Attractor networks in theoretical neuroscience

\[
\begin{align*}
\frac{dx_0}{dt} &= \sigma(x_1 - x_0), \\
\frac{dx_1}{dt} &= x_0(\rho - x_2) - x_1, \\
\frac{dx_2}{dt} &= x_0x_1 - \beta x_2.
\end{align*}
\]
Why the data lives on a manifold

Conservation laws
Redundant features
Other Constraints

particle masses ("on-shell condition")

energy-momentum conservation

14-dimensional manifold embedded in 40-dimensional data space
Why the data lives on a manifold

Conservation laws
Redundant features
Other Constraints

14-dimensional manifold embedded in 40-dimensional data space

particle masses ("on-shell condition")
energy-momentum conservation

Animation by Johann Brehmer
Flows on a prescribed manifold

\[ \tilde{u} \sim p_{\tilde{u}}(\tilde{u}) \quad \quad h \quad \quad u \quad \quad g^* \quad \quad \text{prescribed chart} \quad \quad \text{tractable density over } M^* \]

\[ p_{M^*}(x) = p_{\tilde{u}}(\tilde{u}) \left| \det J_h(\tilde{u}) \right|^{-1} \cdot \left| \det [J_{g^*}^T(u) J_{g^*}(u)] \right|^{-\frac{1}{2}} \]
$\mathcal{M}$-flows

See Spotlight talk by Johann Brehmer “NOTAGAN"
See Spotlight talk by Johann Brehmer "NOTAGAN"

\[ \tilde{u} \sim p_{\tilde{u}}(\tilde{u}) \quad \xrightarrow{h} \quad p_u(u) \]

\[ p_{uv}(u, v) \]

\[ \text{Zero-pad} \]

\[ \text{Project} \]

\[ n\text{-dim. latents} \quad \text{inv. NN} \quad n\text{-dim. latents} \quad \text{embed} \quad d\text{-dim. latents} \]
$\tilde{u} \sim p_\tilde{u}(\tilde{u})$ \hspace{1cm} $h$ \hspace{1cm} $u$

$n$-dim. latents \hspace{1cm} inv. NN \hspace{1cm} $n$-dim. latents

embed \hspace{1cm} $d$-dim. latents \hspace{1cm} inv. NN

tractable density over $\mathcal{M}$

\[
p_{\mathcal{M}}(x) = p_\tilde{u}(\tilde{u}) \left| \det J_h(\tilde{u}) \right|^{-1} \cdot \left| \det \left( \begin{pmatrix} 1 & 0 \end{pmatrix} J_f(u)^T J_f(u) \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right) \right|^{-\frac{1}{2}}
\]
Evaluating data on or off the manifold
Evaluating data on or off the manifold

\[ x_1 \]

\[ x_0 \]

\[ v \]

\[ 0 \]

\[ p_{\tilde{u}}(\tilde{u}) \]
Evaluating data on or off the manifold
Evaluating data on or off the manifold
Evaluating data on or off the manifold
Evaluating data on or off the manifold
Evaluating data on or off the manifold
Evaluating data on or off the manifold

Input $x$ ➞ Representation $\tilde{u}$ (dimensionality reduction)

➤ Projection to manifold $x'$ (denoising)

➤ Reconstruction error $\|x - x'\|$ (training, OOD detection)

➤ Likelihood after projection $p_M(x')$ (training, inference)
## Generative models vs. the data manifold

<table>
<thead>
<tr>
<th>Model</th>
<th>Manifold</th>
<th>Chart</th>
<th>Generative</th>
<th>Tractable density</th>
<th>Restr. to manifold</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ambient flow (AF)</td>
<td>no</td>
<td>no</td>
<td>✓</td>
<td>✓</td>
<td>no</td>
</tr>
<tr>
<td>Flow on prescr. manifold</td>
<td>prescribed</td>
<td>prescribed</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>GAN</td>
<td>learned</td>
<td>no</td>
<td>✓</td>
<td>no</td>
<td>✓</td>
</tr>
<tr>
<td>VAE</td>
<td>learned</td>
<td>no</td>
<td>✓</td>
<td>only ELBO</td>
<td>(no)</td>
</tr>
<tr>
<td>$\mathcal{M}$-flow</td>
<td>learned</td>
<td>learned</td>
<td>✓</td>
<td>✓ (potentially slow)</td>
<td>✓</td>
</tr>
</tbody>
</table>
The likelihood is not what it seems

Likelihood defined after projection to $\mathcal{M}$, which is defined through NN weights $\phi_f$

Family of likelihoods $p_{\phi_f}(x|\phi_h)$ rather than one likelihood $p(x|\phi_f, \phi_h)$

$\implies$ Learning $\phi_f$ by maximizing $p_{\phi_f}(x|\phi_h)$ is unstable

$p_{\phi_f}(x|\phi_h)$ is not really a likelihood function in the parameter $\phi_f$

We call it the “naive likelihood”
The likelihood is not what it seems

Likelihood defined after projection to $\mathcal{M}$, which is defined through NN weights $\phi_f$

Family of likelihoods $p_{\phi_f}(x|\phi_h)$ rather than one likelihood $p(x|\phi_f, \phi_h)$

$\Rightarrow$ Learning $\phi_f$ by maximizing $p_{\phi_f}(x|\phi_h)$ is unstable

$p_{\phi_f}(x|\phi_h)$ is not really a likelihood function in the parameter $\phi_f$

We call it the “naive likelihood”
M/D training

Solution: separate training in two phases!

- **Manifold phase:**
  update $\phi_f$ (and thus $\mathcal{M}$) by minimizing $\|x - x'\|$

![Reconstruction error diagram](image-url)
M/D training

Solution: separate training in two phases!

- **Manifold phase:**
  update $\phi_f$ (and thus $\mathcal{M}$) by minimizing $\|x - x'\|$.

- **Density phase:**
  update $\phi_h$ (and thus $p_M(x)$) by maximum likelihood (keeping $\mathcal{M}$ fixed).
Quantum Field Theory
Flow models for lattice QCD

- MIT-led program to develop flow model architectures for applications across lattice QCD
- Ongoing industry collaboration w/ Google DeepMind
Interaction strength depends on energy

\[ \alpha_s(Q^2) = \alpha_s^{(0)} + \alpha_s^{(1)} \frac{1}{Q^2} + \alpha_s^{(2)} \frac{1}{Q^4} + \ldots \]

\[ \alpha_s^{(0)} = \alpha_s^{(0)} + \alpha_s^{(1)} \frac{1}{Q^2} + \alpha_s^{(2)} \frac{1}{Q^4} + \ldots \]

Low-energy QCD is non-perturbative
The strong force: Quantum Chromodynamics

Interaction strength depends on energy

\[ \alpha_s(Q^2) = \mathcal{O}(\alpha_s^0) + \mathcal{O}(\alpha_s) s + \mathcal{O}(\alpha_s^2) s^2 + \ldots \]

- QCD is weak at high-energies, small coupling, perturbation theory works

QCD is weak at high-energies, small coupling, perturbation theory works
The strong force: Quantum Chromodynamics

QCD is strong at low-energies, no small coupling, perturbation theory fails.

Emergent phenomena: protons, pions, etc.

QCD is weak at high-energies, small coupling, perturbation theory works.
Lattice Field Theory

Lattice field theory is a computational approach to studying interacting field theory on a discretized space-time lattice.

Each link on the lattice has data corresponding to the symmetry group of the theory. For the strong force (QCD) each link has a 3x3 unitary matrix.

This animation is a single configuration of the lattice. Think of a 4-d image playing like a movie.
Lattice field theory is a computational approach to studying interacting field theory on a discretized space-time lattice.

Each link on the lattice has data corresponding to the symmetry group of the theory. For the strong force (QCD) each link has a 3x3 unitary matrix.
Distribution over Configurations

We don't want just a single "image" (lattice configuration), we want to sample the high-dimensional distribution of configurations predicted by the theory.

- **Path integral**: each "path" is a sample from distribution of lattice configurations $\text{path} \sim \exp(-\text{Action}[\text{path}])$.
- Predictions are expectations of quantum operators w.r.t. this distribution.
- That integral is intractable. Typically people use Hamiltonian Monte Carlo for this, but it has limitations.
Distribution over Configurations

We don't want just a single "image" (lattice configuration), we want to sample the high-dimensional distribution of configurations predicted by the theory.

- **Path integral**: each “path” is a sample from distribution of lattice configurations $\text{path} \sim \exp(-\text{Action}[\text{path}])$.
- Predictions are expectations of quantum operators w.r.t. this distribution.
- That integral is intractable. Typically people use Hamiltonian Monte Carlo for this, but it has limitations.
Lattice QCD works

- Ground state hadron spectrum reproduced
- p-n mass splitting reproduced
- Predictions for new states with controlled uncertainties

![Graph showing comparison between Lattice QCD and experiment results for charmed and/or bottom baryons' masses.]

Note that the uncertainties of our results for nearby states are highly correlated, and hyperfine splittings such as $M_{\Omega A} - M_{\Omega B}$ can in fact be resolved with much smaller uncertainties than apparent from this figure (see Table XIX).

Determined post-facto by LHCb experiment

[Z Brown et al. PRD 2014]
Predictions are taken seriously

The Standard Model is successful

Magnetic moment of the electron: (torque an electron feels in a magnetic field) \( a_e = (g - 2)/2 \)

Most accurately verified prediction in the history of physics

<table>
<thead>
<tr>
<th>Theory</th>
<th>( a_e = 0.001159652181643(764) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exp.</td>
<td>( a_e = 0.00115965218073(28) )</td>
</tr>
</tbody>
</table>

\[ a_e = \left( \frac{g - 2}{2} \right) \]
So what’s the problem?

QCD gauge field configurations sampled via Hamiltonian dynamics + Markov Chain Monte Carlo

Updates diffusive

Lattice spacing 0

Number of updates to change fixed physical length scale ∞

“Critical slowing-down” of generation of uncorrelated samples
Flow-based generative models for Markov chain Monte Carlo in lattice field theory

M. S. Albergo,1,2,3 G. Kanwar,4 and P. E. Shanahan4,1
1Perimeter Institute for Theoretical Physics, Waterloo, Ontario N2L 2Y5, Canada
2Cavendish Laboratories, University of Cambridge, Cambridge CB3 0HE, U.K.
3University of Waterloo, Waterloo, Ontario N2L 3G1, Canada
4Center for Theoretical Physics, Massachusetts Institute of Technology, Cambridge, MA 02139, U.S.A.

A Markov chain update scheme using a machine-learned flow-based generative model is proposed for Monte Carlo sampling in lattice field theories. The generative model may be optimized (trained) to produce samples from a distribution approximating the desired Boltzmann distribution determined by the lattice action of the theory being studied. Training the model systematically improves autocorrelation times in the Markov chain, even in regions of parameter space where standard Markov chain Monte Carlo algorithms exhibit critical slowing down in producing decorrelated updates. Moreover, the model may be trained without existing samples from the desired distribution. The algorithm is compared with HMC and local Metropolis sampling for φ^4 theory in two dimensions.
The main approach is thus to start with one configuration, e.g., the folded protein state, and make tiny changes to it over time, e.g., by using Markov-chain Monte Carlo or molecular dynamics (MD). However, these simulations get trapped in metastable (long-lived) states: For example, sampling a single folding or unfolding event with atomistic MD may take a year on a supercomputer.
The action is invariant to local gauge transformations, so the distribution is constant in those directions. It’s a huge product group!

Many more pure gauge degrees of freedom than physical ones

We would like to enforce this symmetry in the network, and not have to learn it.
We designed flows on compact manifolds like Spheres and Tori that correspond to Lie groups:

\[
\begin{align*}
T_{s \rightarrow c} & \quad r' = g(r) \quad \theta' = f(\theta; r') \\
S^2 & \quad S^1 \times [-1, 1] \\
T_{c \rightarrow s} &
\end{align*}
\]

Figure 3. Learned densities on $T^2$ using NCP, Möbius and CS flows. Densities shown on the torus are from NCP.

Figure 5. Learned multi-modal density on $SU(2) \equiv S^3$ using the recursive flow. Each column shows an $S^2$ slice of the $S^3$ density.
Step 2:

We came up with a way to build flows that are equivariant to space-time translations and local gauge transformations.
Essentially, MCMC can get stuck for a while in a certain mode.

- Our new “flow-based” proposal does much better!
- It learns to propose configurations that look like our target distribution.
- 1000x reduction in autocorrelation time

The topological charge $Q$ will be constant for thousands of MCMC steps.
Space-time & Local, Non-Abelian Gauge Symmetry

Convolution and Masking Pattern

Input Configuration

Eigen Decomposition

 Canonical Permutation

 Flow in Canonical Cell

 Invert Canonical Permutation

 Recover Eigenvalues

 Eigen Recomposition

Output Configuration

Eigenvalues $\{\lambda_i\}$

Extract Angle Coordinates

SU(3) flows

SU(9) flows

Floor

Target

Haar SU(3)
Flow models for QCD

First gauge theory application: 2D U(1) field theory

Cost per independent sample

![Cost per independent sample graph]

- Conventional approaches
- ML algorithm

Parameter of theory

$\tau_Q^{\text{int}}$

$\beta$

$\beta$-values: 1, 2, 3, 4, 5, 6, 7

$\tau_Q^{\text{int}}$ values: 1, 10, 100, 1000, 10000

[Phys.Rev.Lett. 125, 121601 (2020)]

Phiala Shanahan, MIT
Flow models for QCD

Systems with complex topologies

Need: Unbiased sampling from multi-modal distributions
Flow models for QCD

First gauge + fermion theory application:
2D Schwinger model

Flows on compact, connected manifolds
Multimodal distributions
Gauge-equivariant flows
Fermions (quarks)
Fermions + gauge theory

Measured value of observable

Number of samples

Truth line

# Flow models for QCD

First gauge + fermion theory application: 2D Schwinger model

---

## Schwinger model

**Article**  |  **Talk**
--- | ---
From Wikipedia, the free encyclopedia

In physics, the Schwinger model, named after Julian Schwinger, is the model describing 1+1D (1 spatial dimension + time) Lorentzian quantum electrodynamics which includes electrons, coupled to photons.

The model defines the usual QED Lagrangian

\[ \mathcal{L} = -\frac{1}{4g^2} F_{\mu\nu} F^{\mu\nu} + \bar{\psi} \left( i \gamma^\mu D_\mu - m \right) \psi \]

over a spacetime with one spatial dimension and one temporal dimension. Where \( F_{\mu\nu} = \partial_\mu A_\nu - \partial_\nu A_\mu \) is the U(1) photon field strength, \( D_\mu = \partial_\mu - iA_\mu \) is the gauge covariant derivative, \( \psi \) is the fermion spinor, \( m \) is the fermion mass and \( \gamma^\mu, \gamma^5 \) form the two-dimensional representation of the Clifford algebra.

This model exhibits confinement of the fermions and as such, is a toy model for QCD. A handwaving argument why this is so is because in two dimensions, classically, the potential between two charged particles goes linearly as \( 1/r \), instead of \( 1/r^2 \) in 4 dimensions, 3 spatial, 1 time. This model also exhibits a spontaneous symmetry breaking of the U(1) symmetry due to a chiral condensate due to a pool of instantons. The photon in this model becomes a massive particle at low temperatures. This model can be solved exactly and is used as a toy model for other more complex theories.\(^\text{[1,3]}\)

## References

Flow models for QCD in 4D

Initial QCD demonstration [this talk +upcoming manuscripts on scaling and 4D]

- Direct combination of published results on gauge-equivariant flows and pseudofermions [Boyda et al., 2008.05456, Abbott et al., 2207.08945]
- Illustration at straightforward parameters $V=4^4$, $N_f=2$, $\beta=1$, $\kappa=0.1$
- Observables from flow ensemble in precise agreement with HMC at high statistics (65k samples)
- Development and scaling of QCD-specific architectures in full swing — stay tuned!

Marginal:
- Haar-uniform base distribution
- 48 gauge-equivariant spline coupling layers
- Spatially separated convolutions in spectral flow to define spline parameters

Conditional:
- Gaussian base distribution
- 36 pseudofermion coupling layers built from parallel transport convolutional networks
- Alternating spin and spatial masking pattern
Flow models for QCD in 4D

Initial QCD demonstration
[this talk +upcoming manuscripts on scaling and 4D]

- Direct combination of published results on gauge-equivariant flows and pseudofermions [Boyda et al., 2008.05456, Abbott et al., 2207.08945]
- Illustration at straightforward parameters $V=4^4$, $N_f=2$, $\beta=1$, $\kappa=0.1$
- Observables from flow ensemble in precise agreement with HMC at high statistics (65k samples)
- Development and scaling of QCD-specific architectures in full swing — stay tuned!

Plaquette

Topological charge (Wilson flowed)

Pion correlation function (t=1)
Flow models for QCD

Machine learning for QCD

- **Provably-exact** machine-learning-accelerated sampling algorithm
- Orders of magnitude more **efficient** than conventional algorithms overcoming critical slowing-down
- **Unbiased** results where traditional approaches fail

Deployment for state-of-the-art QCD scheduled for Aurora 2023 first science time

Inductive Bias

Separation

Compositionality

Symmetry

Relationships

Causality
What if we had the right causal structure?

CLAIM: Under the hypothesis of independent mechanisms and small changes across different distributions:
  - smaller sample complexity to recover from a distribution change
  - E.g. for transfer learning, agent learning, domain adaptation, etc.

Another way of seeing it is: the right causal graph suggests a particular factorization of the joint distribution (a directed bayesian network). A causal intervention means that you only change one of these factors (or a few factors) while leaving the other ones unchanged. Therefore if your generative model is the right causal model, meaning that it factorizes the joint in the same way, it will be easy to adapt it to the change because only a few parameters need changing (those associated with the factors that actually changed).
The message from human cognition:

Richly structured models of objects and their relations are a powerful tool for reasoning about, and interacting with, the world.

- Objects and relations reflect *decisions* made by evolution, experience, and task demands about how to represent the world in an *efficient and useful* way.
- Intelligence is about *model-building*, beyond just recognizing patterns (Tenenbaum).
- *Combinatorial generalization* via abstraction and compositionality ("infinite use of finite means").
Main challenge: wide range of spatiotemporal scales

Coarse-graining idea: Can we design a minimalist model speeding up the simulations significantly while preserving the relevant physics?

Figure from J.E. Shea
Emergence: Philosophical Musings
Scale separation & emergence

Scale separation can lead to different effective descriptions & ontologies that describe the phenomena that emerge at different scales

- Identifying and naming the relevant objects / concepts already significant
- Understanding how they interact and developing an effective law or theory at that scale is even more significant
- Understanding how these objects and interactions emerge from a more fundamental scale is profound

This has generally been done by humans, and there is an opportunity for AI to assist / accelerate / automate this process.
Questions:

- How arbitrary or unambiguous are:
  - the scales where the “right” effective description applies?
  - the right objects / degrees of freedom in the effective description?
  - the laws that describe the interactions among those objects?

- Is there a principle that can help guide us or allow us to judge or rank different approaches?
Observation:

• Dynamics of lower-dimensional coarse-grained model sweep out a manifold in the state space of the fine-grained model

• Coarse graining and emergence can be seen as geometrical structure of the “data manifold”

• Useful insight for generative models, up-sampling, denoising etc.
Evaluating data on or off the manifold

Vanilla autoencoder acting general-purpose like compression.

- When trained on L2 loss, not specialized for any particular down-stream task
Vanilla autoencoder acting general-purpose like compression.

- When trained on L2 loss, not specialized for any particular down-stream task
Evaluating data on or off the manifold

Vanilla autoencoder acting general-purpose like compression.

- When trained on L2 loss, not specialized for any particular down-stream task
Evaluating data on or off the manifold

Vanilla autoencoder acting general-purpose like compression.

- When trained on L2 loss, not specialized for any particular down-stream task
Supervised learning and sufficiency

In contrast, say we have some down-stream task, then some of the information in $x$ will be useful, but other information can be thrown away without any significant loss in performance.

- This happens automatically in learned representations for supervised learning tasks

For example, say $\theta$ represents some property that is useful for my downstream task, and abstractly I can think about the joint $p(x, \theta)$ or conditional $p(x | \theta)$ — example: think of $\theta$ as a reaction coordinate

- A function (encoder) $T(x)$ is called a sufficient statistic for $\theta$ if it can be factorized as
  
  $$p(x | \theta) = g(T | \theta) h(x)$$

- Equivalently
  
  $$I(\theta; T(X)) = I(\theta; X)$$
  
  $$p(\theta | X = x) = p(\theta | T(X) = t(x))$$

Closely related to collective variables, order parameters, etc.

- Exact sufficient statistics don’t usually exist, but approximate sufficient globally
  
  $$t(x | \theta^r) = \nabla_\theta \log p(x | \theta) |_{\theta^r}$$ is “locally sufficient“
Coarse Graining Auto-Encoding Framework

\[ L_{	ext{CGAE}} = \beta E(x) + \beta V_{FZ}(E(x)) \| \dot{x} E(x) - \dot{x}_0 \|^2 + \gamma V_CG(x, z) \]
Coarse Graining Auto-Encoding Framework

- **AutoEncoder** automatically coarse-grains atomistic coordinates to CG coordinates in a data-driven way.
- **Force matching** also helps to shape the learning of CG and obtain $V_{CG}(z = E(x))$ for CG simulations.
Geometrical Picture

Coarse Graining Auto-Encoding Framework

\[ L_{	ext{CGAE}} = \| \mathbf{V} \cdot \mathbf{x} + \mathbf{F}_E(\mathbf{x}(s)) \|^2 + \| \mathbf{E}(\mathbf{x}(s)) - \mathbf{x}(s) \|^2 + \rho_{\text{Ian}}(E(x)) \|^2 \]

- **AutoEncoder** automatically coarse-grains atomistic coordinates to CG coordinates in a data-driven way
- **Force matching** also helps to shape the learning of CG and obtain \( V_{\text{CG}}(z = E(x)) \) for CG simulations

Coarse Graining Auto-Encoding Framework

- **AutoEncoder** automatically coarse-grains atomistic coordinates to CG coordinates in a data-driven way.
- **Force matching** also helps to shape the learning of CG and obtain $V_{CG}(z = E(x))$ for CG simulations.

$$L_{CGAE} = | - b \cdot \nabla E + V_{CG}(E(x)) |^2 + |D E(x) - x_i^2 + pF_{acc}(E(x)) |^2$$
Equivariant generative decoder

Information is always lost in CG – needs to be recovered statistically.
All atom to CG is surjective, a generative (non-deterministic) model is needed
Avoid FF refinement.
Create latent variable to hold info for decoding (depends on x and X at train and only on X during inference).
Equivariant decoding through inter-bead vectors.

Coarse Graining Auto-Encoding Framework

- **AutoEncoder** automatically coarse-grains atomistic coordinates to CG coordinates in a data-driven way
- **Force matching** also helps to shape the learning of CG and obtain $V_{CG}(z = E(x))$ for CG simulations

$$L_{GCE} = \| \theta_x V_x + \theta_z V_{CG}(E(x)) \|^2_2 + D(E(x_1) - x_1^2) + \rho_F \text{ReLU}(E(x))^2$$
Coarse Graining Auto-Encoding Framework

\[ L_{CGE} = | - \mathbf{b} \cdot \nabla \mathbf{x} + \nabla E(x)^2 | + pF_{int}(E(x))^2 \]

- AutoEncoder automatically coarse-grains atomistic coordinates to CG coordinates in a data-driven way
- Force matching also helps to shape the learning of CG and obtain \( V_{CG}(z = E(x)) \) for CG simulations

Equivariant generative decoder

Information is always lost in CG – needs to be recovered statistically.
All atom to CG is surjective, a generative (non-deterministic) model is needed
Avoid FF refinement.
Create latent variable to hold info for decoding (depends on x and X at train and only on X during inference.
Equivariant decoding through inter-bead vectors.
**Geometrical Picture**

**Coarse Graining Auto-Encoding Framework**

- **AutoEncoder** automatically coarse-grains atomistic coordinates to CG coordinates in a data-driven way.
- **Force matching** also helps to shape the learning of CG and obtain $V_{CG}(z = E(x))$ for CG simulations.

\[ L_{CGAE} = | - \sum_b \sum_z \nabla_x V(z) \cdot \nabla_x E(x) |^2 + | \sum_b \sum_z \nabla_x V(z) \cdot \nabla_x E(x) |^2 + \lambda D E(x) - \nabla_x E(x) |^2 + \rho \sum_z \nabla_x E(x) |^2 \]

**Equivariant generative decoder**

- Information is always lost in CG – needs to be recovered statistically.
- All atom to CG is surjective, a generative (non-deterministic) model is needed.
- Avoid FF refinement.

Create latent variable to hold info for decoding (depends on $x$ and $X$ at train and only on $X$ during inference).

Equivariant decoding through inter-bead vectors.
We designed flows on compact manifolds like Spheres and Tori

Figure 3. Learned densities on $\mathbb{T}^2$ using NCP, Möbius and CS flows. Densities shown on the torus are from NCP.

Figure 5. Learned multi-modal density on $SU(2) \equiv S^3$ using the recursive flow. Each column shows an $S^2$ slice of the $S^3$ density.
Coarse graining

Deterministic Back mapping
Sparse Identification of Nonlinear Dynamics (SINDy)

Innovation 1: Enforcing known constraints
- Skew-symmetric quadratic nonlinearities to enforce energy conservation
- Improved stability

Innovation 2: Higher-order Nonlinearities
- Cubic, Quintic, Septic terms approximate truncated terms in Galerkin expansion

\[
\begin{align*}
\dot{x} &= \mu x - \omega y + A x z \\
\dot{y} &= \omega x + \mu y + A y z \\
\dot{z} &= -\lambda(z - x^2 - y^2).
\end{align*}
\]

SLB, Proctor, Kutz, PNAS 2016.
Noiseau & SLB, JFM 838, 2018

Machine Learning for Scientific Discovery, with Examples in Fluid Mechanics

Steve Brunton
University of Washington
Various strategies

Take advantage of already known multi-scale, emergent phenomena
- Enhanced sampling, coarse graining, ...
- Engineered features, inductive bias of models, ...

Add the coarse graining by hand and learn the dynamics
- Learned force fields, force matching, etc.
- Learn Markov transitions between fixed clustering of states

Add the coarse graining by hand, and learn the effective “dynamics” & how to map back to fine-grained representation
- Steve Brunton’s talk: Reduced models, SINDy
- AlphaFold / OpenFold etc. Sequence $\Rightarrow$ structure (not really dynamics)

Simultaneously learn a (latent) coarse-grained representation and “dynamics” & how to map back to fine-grained representation
- VAEs, Diffusion Models, $\mathcal{M}$-flows

Simultaneously learn a coarse-grained representation and dynamics (discovery emergence)
- Learned Koopman operators, learned dynamics of latent space
- SSL techniques like VicREG, Barlow Twins, etc. where encoder, but no decoder.
- Much of ML does this, but interpretability of latent state is a challenge. When would we call this “emergence”?
  - Need a way to “operationalize” the latent space representation for some down-stream task
Humans are remarkable at being able to have a library of mental models at different levels of abstraction and finding which is most appropriate to use for a given task.

- In my work as a particle physicist I switch between ~5 mental models

Finding the right level of abstraction / coarse graining is key and depends on task

Eventually AI / ML systems may develop causal representations needed to efficiently design experiments, generate hypotheses, etc.

- It may be a foreign ontology, but I suspect that it will need to be causal to be effective