A Triangle of Influence:

Bringing Together Physics, Pure Mathematics, and Computer Science

Northeastern University

Jim Halverson

+ Three Connections @ Physics / ML Interface

Institute for Artificial Intelligence and Fundamental Interactions (IAIFI)

one of five new NSF AI research institutes, this one at the interface with physics! MIT, Northeastern, Harvard, Tufts.

ML for physics / math discoveries?
Can physics / math help ML?

Colloquia begin in Spring!
www.iaifi.org, @iaifi_news

Physics ∩ ML

Physics Meets ML

virtual seminar series at the interface, "continuation" of 2019 meeting at Microsoft Research.

Bi-weekly seminars from physicists and CS, academia and industry.

Sign up at www.physicsmeetsml.org.

Feel free to contact me!

e-mail: jhh@neu.edu
Twitter: @jhhalverson
web: www.jhhalverson.com

ML for Math:
e.g. "Learning to Unknot": 2010.16263

ML for Strings:
e.g. "Statistical Predictions in String Theory and Deep Generative Models": 2001.00555

The Deep Learning Revolution: Supervised Learning

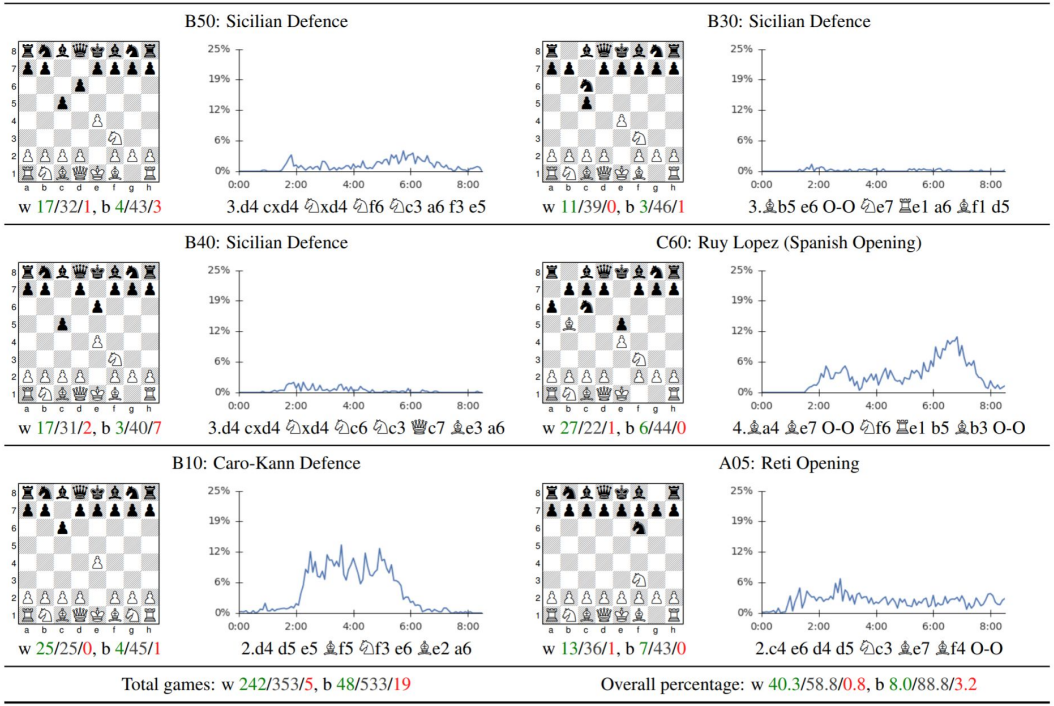
What? learn to predict outputs, given inputs.
test on unseen data, the "test set."

Many simpler algorithms.
Recently deep neural nets have taken over.

Right. image from a famous dataset, MNIST.
Goal: predict 0-9, given the image.

"The usual thing" people mean by deep learning, and "the usual thing" people use in physics.

The Deep Learning Revolution: Reinforcement Learning

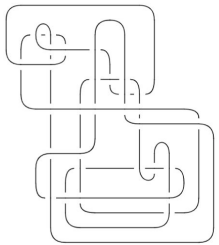


[Silver et al, 2019]

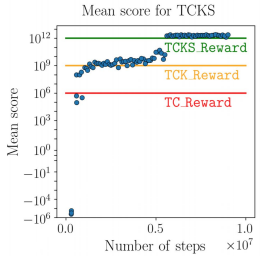
What? agent explores state space according to policy, a state → action map. Receives rewards, updates policy accordingly.

Left. AlphaZero learns Chess openings. Crushes conventional program that crushes the best humans. *Watch AlphaGo on Netflix!*

In physics / math, e.g.:



[J.H., Gukov, Ruehle, Sulkowski]
arXiv: 2010.16263 (appearing in Sept '20)



[J.H., Nelson, Ruehle]
arXiv: 1903.11616.
punctuated equilibria!

The Deep Learning Revolution: Generative Models

What? Learns to generate / fake / simulate data.

Idea: neural net maps learns to map noise N ~ P(N) to draws from some target data distribution.

Right. Images generated with VQ-VAE2.

In physics, e.g.:

Simulate GEANT4 ECAL simulator.

CaloGAN, [Paganini et al, 2018].

Simulate string theory EFTs, ALP kinetic terms.

used Wasserstein GAN, [J.H., Long, 2020]

VQ-VAE2, [Razavi et al, 2019]

These "people" do not exist!
Generated by a neural network from noise.

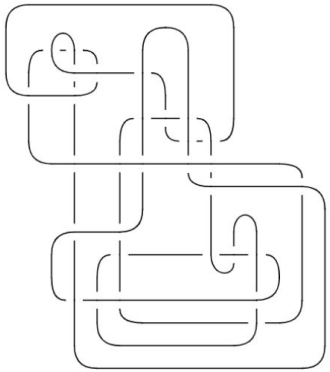
The power and role of neural networks

Supervised:

NN is powerful function that predicts outputs (e.g. class labels), given input.

Generative Models:

NN is powerful function that maps draws from noise distribution to draws from data distribution.



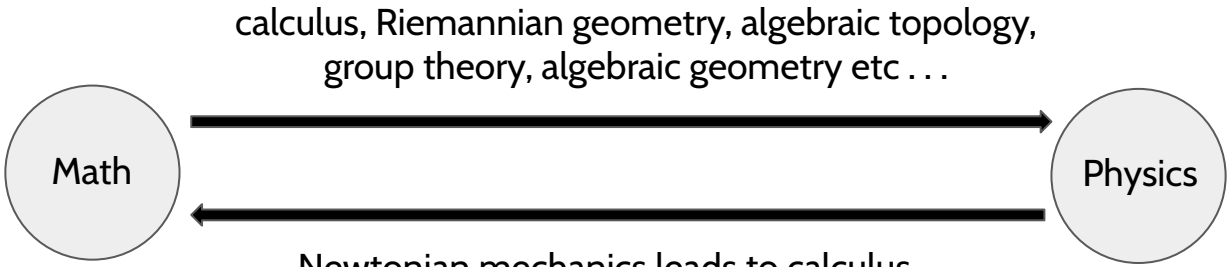
Reinforcement:

NN is powerful function that, e.g., picks intelligent state-dependent actions.

What are we to make of this?

seeing new directions, how does it fit with the way we normally think about the natural sciences and physics specifically?

The Usual Story: Math and Physics belong together

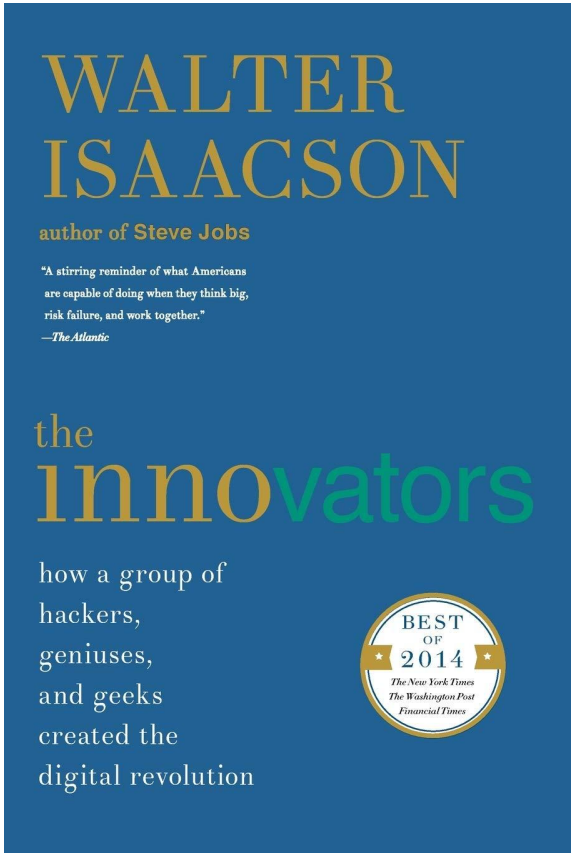


Eugene Wigner

"The unreasonable effectiveness of mathematics in the natural sciences."

But computer science is *young*!

- **1800s:** Lovelace and Babbage lay foundations
- **1940s-50s:** Turing et. al. develop theory of computation, major experimental breakthroughs.
- **1980s:** personal computers take over. **Accessibility!**
- **1990s-2000s:** the internet and power grow.
- **2010s:** deep learning breakthroughs.
- **2020s:** ???



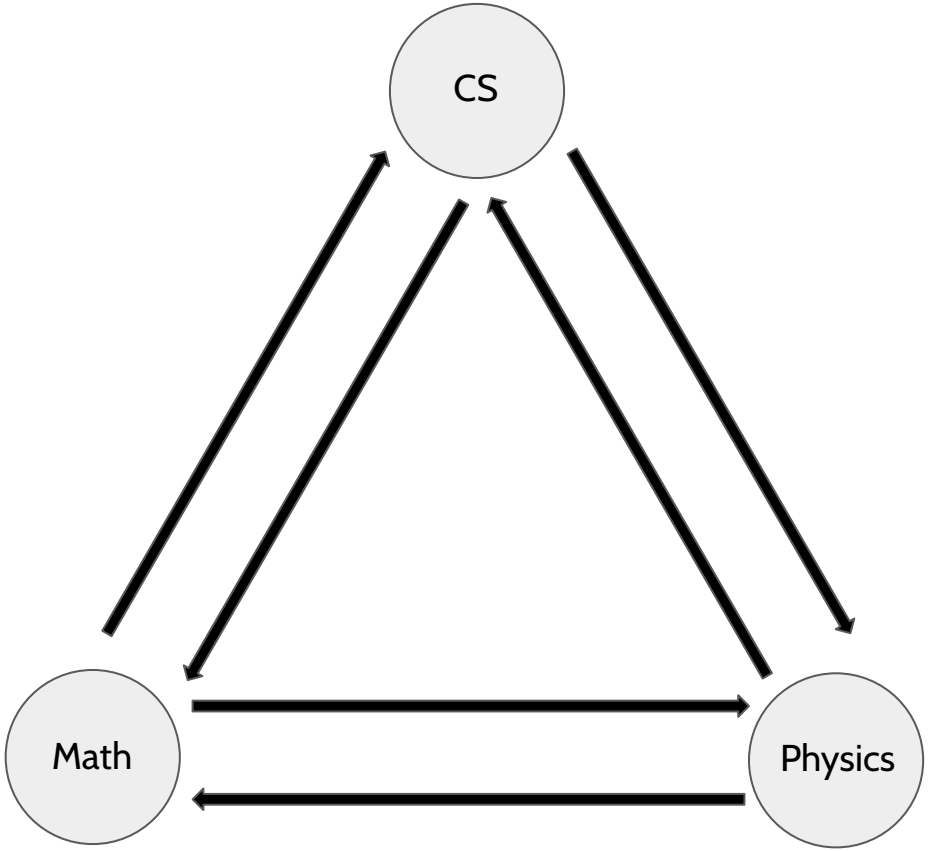
excellent biography of founders of CS

Computer science is still an infant

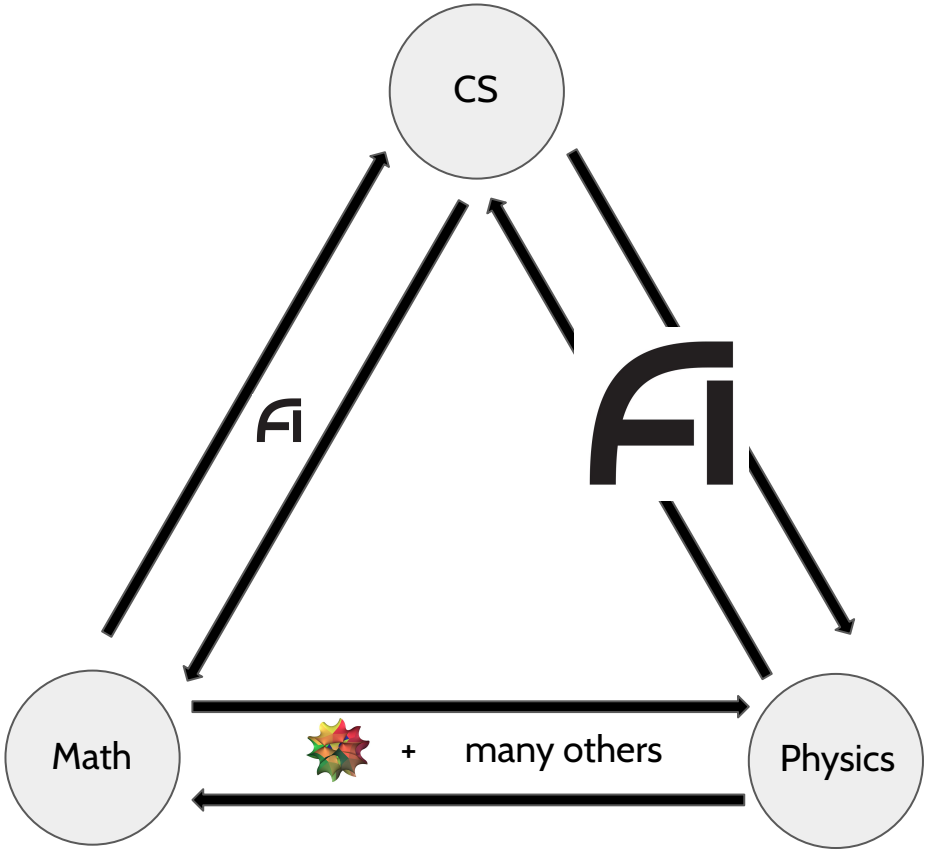
This is not an insult! It is ***exciting***.

What will people say about this first 200 year period in 500 or 1000 years?

How will it fit in?



Towards a triangle of influence?



Towards a triangle of influence?

The NSF AI Institute for Artificial Intelligence and Fundamental Interactions (IAIFI)

one of five new NSF AI research institutes, this one at the interface with physics!

MIT, Northeastern, Harvard, Tufts.

ML for physics / math discoveries?
Can physics / math help ML?

Colloquia begin February 4!
www.iaifi.org
@iaifi_news

Overlaps between large and diverse fields!

can only give you a glimpse of some possibilities.

Outline: Two Legs of the Triangle

Example 1: ML → Math

Knot theory is a beautiful subject in topology.

Can ML help us ***unknot***?

Remarkably similar to natural language.

Example 2: Physics → ML

Neural networks are surprisingly like quantum field theory.

Can techniques from the latter help us understand the former?

Example 1: ML → Math

“Learning to Unknot”

[Gukov, J.H., Ruehle, Sulkowski] arXiv: 2010.16263
to appear in Machine Learning: Science and Technology

Related knots / ML works:
[Hughes] 1610.05744
[Jejjala, Kar, Parrikar] 1902.05547
[Craven, Jejjala, Kar] 2012.03995

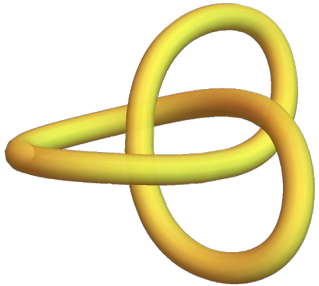
Example 1: Outline

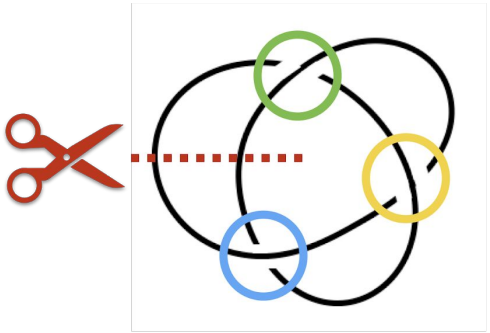
- Knots and Natural Language
- The Unknot Problem
- Decisions, Reformers, and Hard Knots
- Unknotting and Reinforcement

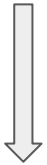
Knots and Natural Language

why are these two things related!?

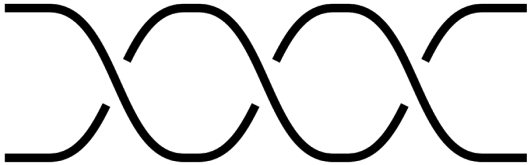
Knots and Braids







$\sigma_1 \ \sigma_1 \ \sigma_1$



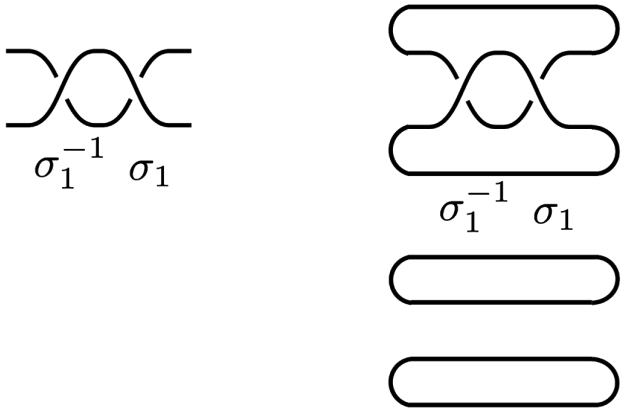
$\sigma_1 \quad \sigma_1 \quad \sigma_1$

Artin Braid Group

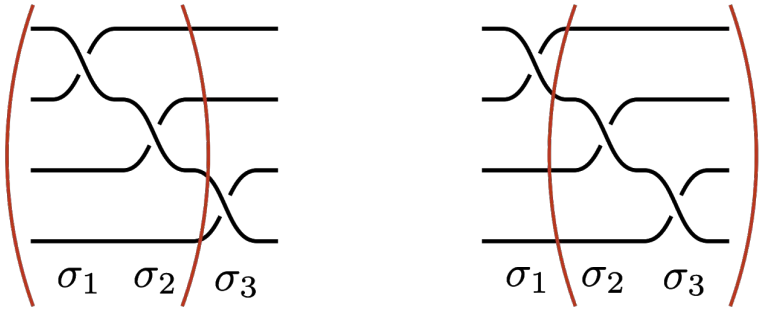
Identity:

Composition:

Inverse:

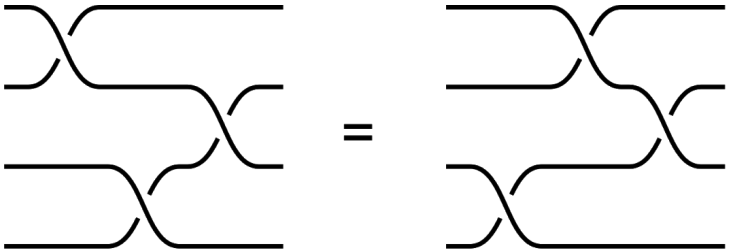


Associativity:



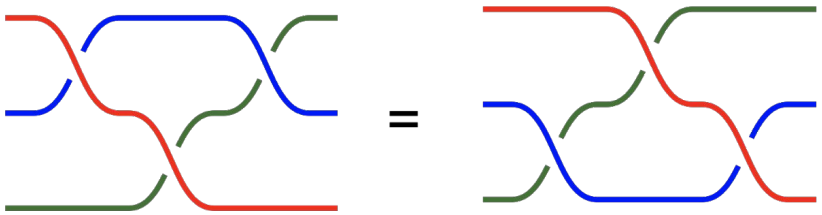
Braid Equivalence

Braid Relation 1:



$$\sigma_1 \ \sigma_3 \ \sigma_2 \qquad \sigma_3 \ \sigma_1 \ \sigma_2$$

Braid Relation 2:



$$\sigma_1 \ \sigma_2 \ \sigma_1 \qquad \sigma_2 \ \sigma_1 \ \sigma_2$$

(i.e. some generators commute)

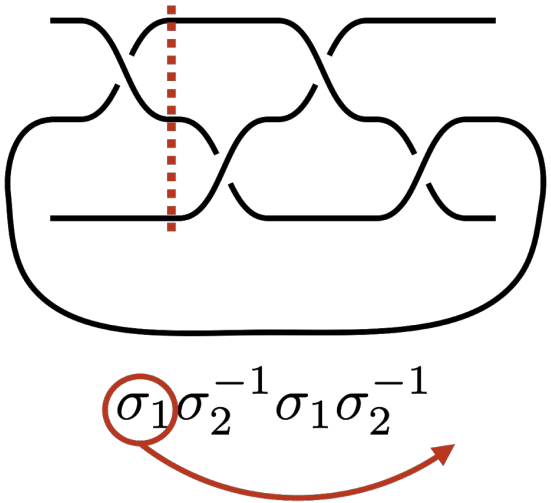
$$\sigma_i \sigma_j = \sigma_j \sigma_i \quad \text{if} \quad |i-j| > 1$$

(i.e. can rearrange braid)

$$\sigma_i \sigma_{i+1} \sigma_i = \sigma_{i+1} \sigma_i \sigma_{i+1}$$

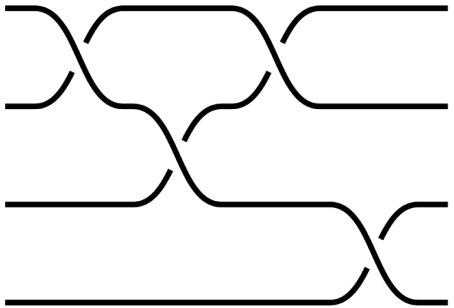
Knot Equivalence

Markov Move 1: Conjugation



$$w \rightarrow \sigma_i^{\pm 1}\, w\, \sigma_i^{\mp 1}$$

Markov Move 2: Stabilization



$$w \rightarrow w\, \sigma_{N+1}^{\pm 1}$$

The Upshot

Thought of as braids, knots are represented by **words**.

Knot equivalence becomes equivalences between different words.

How do we determine when two words (sentences) carry the same meaning?

These questions are the domain of ***natural language processing***.

Natural Language Processing

Learning Semantics:

E embeds words into vector space.
e.g. E(king) - E(man) + E(woman) = E(queen)

Generative language models:

bold text: human-written prompt.
rest: generated by GPT-3 (OpenAI).

Below is a screenplay for a film-noir hard boiled detective story by Raymond Chandler about the boy wizard Harry Potter:

“Harry Potter”
By Raymond Chandler

Harry Potter, Private Eye.
Scene: A small dingy office, early morning, furniture of the Salvation Army store variety. Sordid atmosphere. Harry Potter, in ratty tweed suit, unpressed shirt, and unshined shoes, sits behind the desk looking haggard, rumpled, and embittered. In one corner of the office a torn shade rattles at the wind gusts outside. Harry has one long index finger on the butt of a .38 and he broods over a photograph. The photograph shows a gorgeous blonde wearing an evening gown. She is holding a cocktail in one hand and looking pensively at a cigarette in the other. A tear adorns her cheek. Harry looks at her for a long moment, sighs, then puts his gun down and stands. He straightens his clothes, scratches his head, then with more resolution turns around, picks up his pistol, and leaves the office. He passes through the front room and looks out the dirty window. An old woman is washing the sidewalk outside with a mop. A young man in a double breasted gray suit is leaning against the building. Harry sighs again and goes out the door. He walks up to the young man and without ceremony punches him in the jaw.

NLP in Our Context

Learn commutativity:

He's sometimes right = Sometimes he's right $\sigma_1\sigma_3 = \sigma_3\sigma_1$

The scientist eats the chicken ≠ The chicken eats the scientists $\sigma_1\sigma_2 \neq \sigma_2\sigma_1$

Learn equivalences:

The scientists read the paper = The paper was read by the scientists

$$w = w\sigma_{N+1}$$

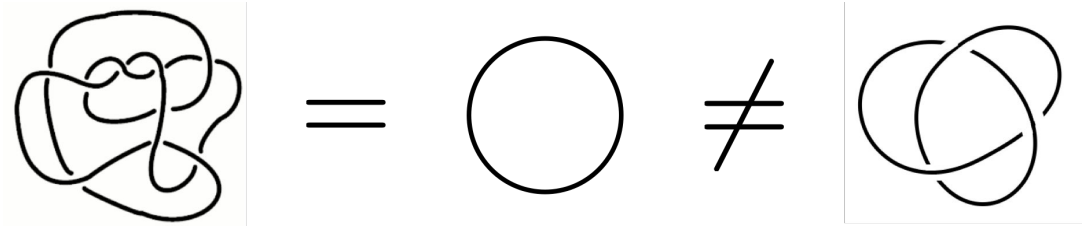
$$w = \sigma_i^{-1} w \sigma_i$$

The Unknot Problem

a simple-to-state but difficult-to-solve tangling problem.

The Unknot Problem

Q: is a given knot K the unknot?



Knot invariants?

Alexander Polynomial: 1 for unknot, converse is not true, + fast.
Jones Polynomial: 1 for unknot, converse not known to be true but is true for up to 24 crossings, but slow #P-hard.
Khovanov Homology: detects the unknot, but slow (fast would contradict #P-hard Jones).

No known fast invariant that detects the unknot.

Knot or not? A game for children

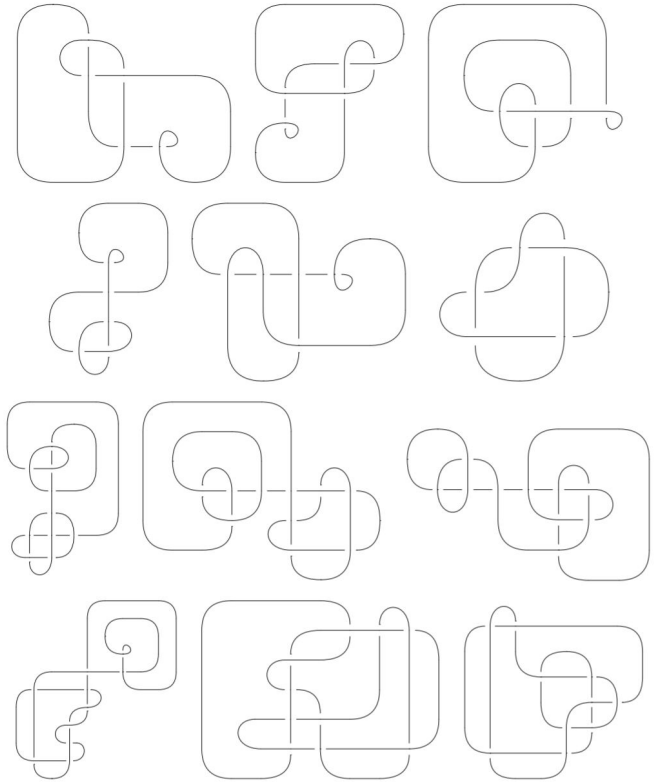


Figure 14: Knot or not? Five and ten crossing in rows 1-2 and 3-4, respectively.

[12]Solutions are presented left-to-right, top-to-bottom, with K and U denoting non-trivial knots and unknots, respectively. Fig. 14: KUUKUKUUKUKK. Fig. 15 UKUKKUKKUKUU. Fig. 16 KUUKUKUUKKUK.

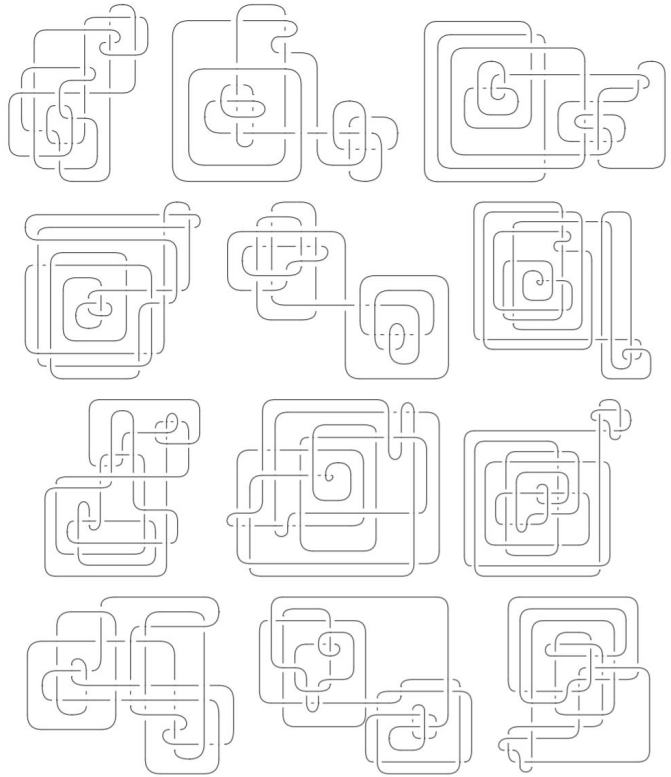
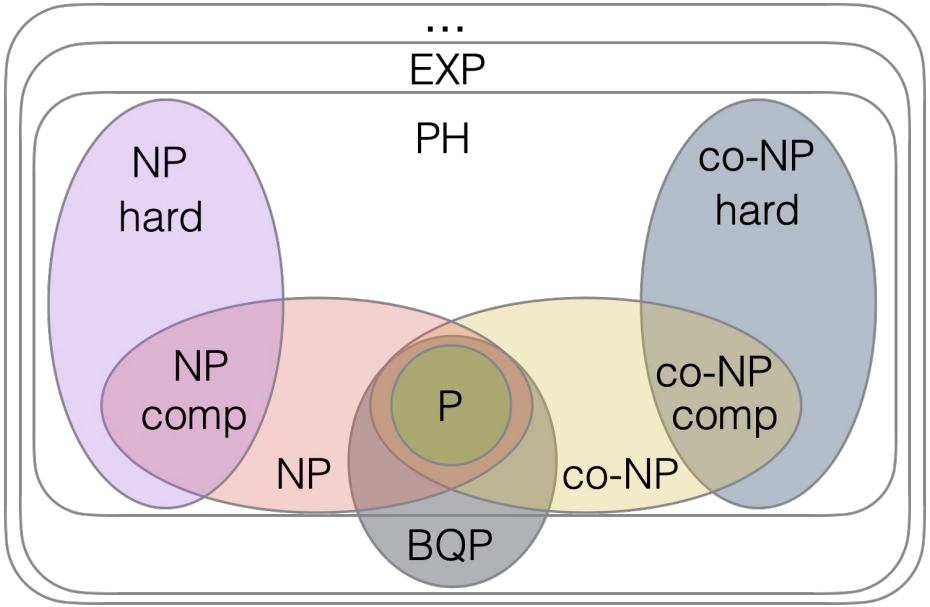


Figure 16: Knot or not? Twenty-five and thirty crossing in rows 1-2 and 3-4, respectively.

point: difficulty increases with crossings.

The Unknot Problem



Try all moves? exp-time

Complexity:

- in NP [Hass, Lagarias, Pippenger '99]
- in co-NP [Kuperberg '14m Lackenby '16]

  means probably not NP-complete since
  then NP = co-NP, which is opposite of consensus.

- but is it P, like primes? [Agrawal, Kayal, Saxena, '02]
  or BQP, like factorizing integers? [Shor '94]

  or something else entirely?

See [Lackenby, 2002.02179] for a recent survey of results.

Decisions and Reformers

can neural networks decide the unknot problem?
do modern NLP architectures help?

Generating Data: Priors for Random Knots and Unknots

Algorithm 5 RandomUnknot: generate random unknot representative.

Require: $n_{\text{letters}}, M \in \mathbb{Z}$.

```
Braid B ← empty braid word.
while |B| ≠ n_letters do
    if |B| > n_letters then
        B ← empty braid word.
    end if
    for k ∈ {1, ..., M} do
        B ← RandomMarkovMove(B).
        if |B| − 1 ≥ 0 then
            B ← BraidRelation2(B, start position ~ U({1, ..., |B|})).
        end if
    end for
    B ← SmartCollapse(B).
end while
return B.
```

Algorithm 6 RandomKnot: generate random non-trivial knot representative.

Require: $n_{\text{letters}}, n_{\text{strands}}, M \in \mathbb{Z}$.

```
Braid B ← empty braid word [].
while |B| ≠ n_letters do
    if |B| > n_letters then
        B ← empty braid word.
    end if
    while |B| < n_letters do
        i ~ U({0, 1}).
        j ~ U({0, ..., n_strands − 1}).
        B ← B + [(−1)^i j]
    end while
    B ← Knotify(B)
    if B ≠ [] then                ▷ Knotify sometimes yields an empty word.
        for k ∈ {1, ..., M} do
            B ← RandomMarkovMove(B).
            B ← BraidRelation2(B, start position ~ U({1, ..., |B|})).
        end for
        B ← SmartCollapse(B).
    end if
end while
return B.
```

Q: how do we generate examples?

The Prior at Low Crossing Numbers

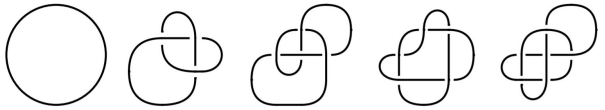


Figure 1: Examples of knots. From left to right: unknot (0_1), trefoil (3_1), figure-eight (4_1), 5_1, and 5_2.

Fact: knots with 9 or fewer crossings are det'd by Jones poly.

Therefore, sample from our priors for N <= 9, map onto Rolfsen.

Notes:

- deviates from uniform distribution.

- small N + trefoil most likely,
but increasingly less likely for larger N.

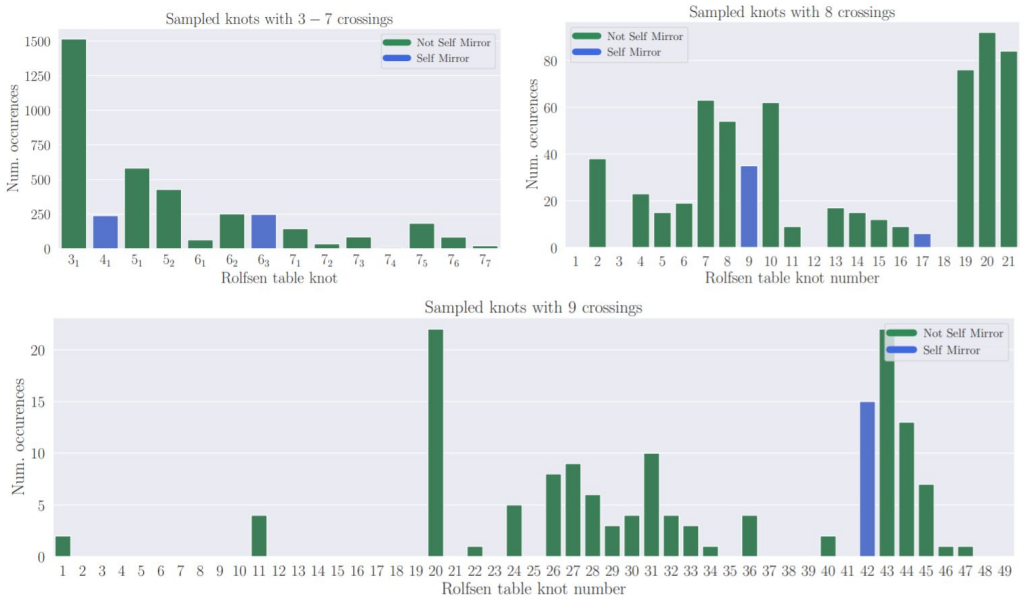


Figure 8: Drawing 6455 $N = 9$ braids from our prior yields knots with 9 or fewer crossings, 4664 of which are prime. Plotted are the number of occurences of knots in the Rolfsen table for knots with 3 through 9 crossings, with mirrors counted for knots that are not self-mirror.

Attention, Attention! Meet Reformers.

Attention Mechanism: learn what in the sequence carries the most meaning, i.e. pay attention to it.

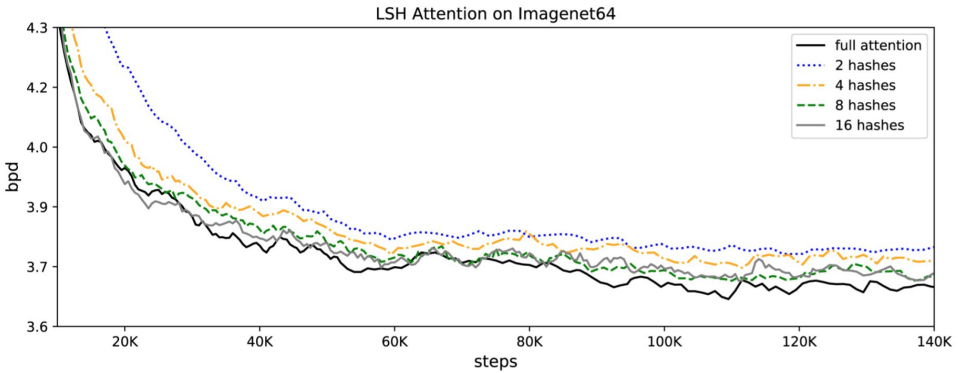
"Attention is all you need" [1706.03762, Vaswani et al.]

Q: Think about beginning of my talk.
what words do you remember and why!?

Reformer: The Efficient Transformer

Upshot: efficiency gains due to replacing scaled dot-product attention with locality-sensitive hashing (LSH) attention. $O(L^2) \rightarrow O(L \log L)$

"Reformer: The Efficient Transformer" [2001.04451, Kitaev et al.]
Good library: reformer-pytorch



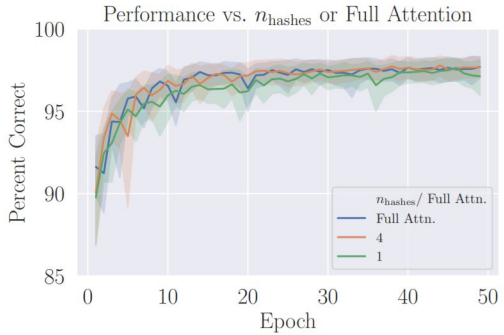
Decisions, Decisions

binary class. on unknot decision.

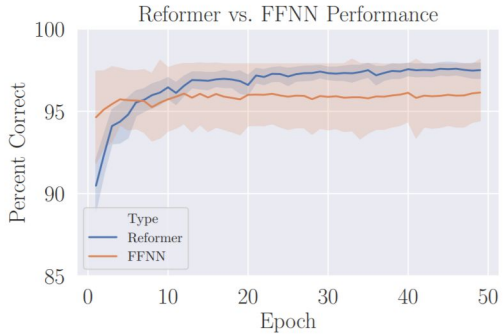
trained on thousands of knots and unknots with diff. #s of crossings.

Comments:

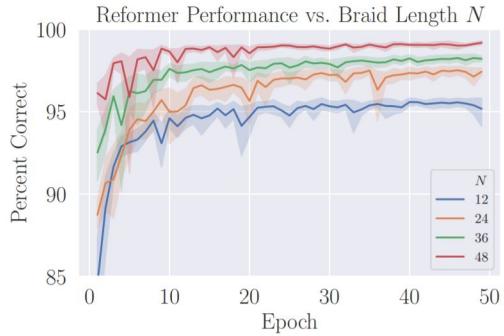
1) NLP wins, but barely. (b/c easy?)
2) Reformers ~ Transformers
3) performance up with N, a lot of fixed # words, less so for fixed # letters.



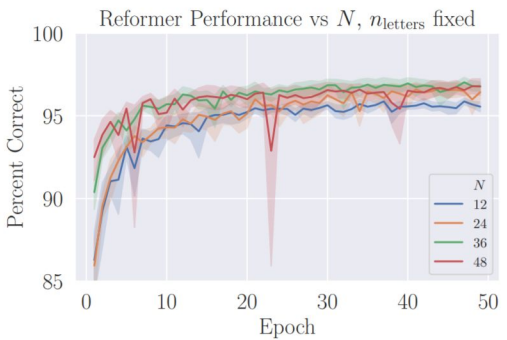
(a) Performance dependence on the number of locality sensitive hashes.



(b) Performance comparison between reformer and feedforward network.



(c) Performance dependence on the braid length. Performance increases with N.



(d) Performance when number of braid letters, rather than number of braid words, is fixed.

Hardness and Jones Correlations

Hardness:

@ right, note some small peaks in wrong spot, networks quite sure of their wrong predictions!

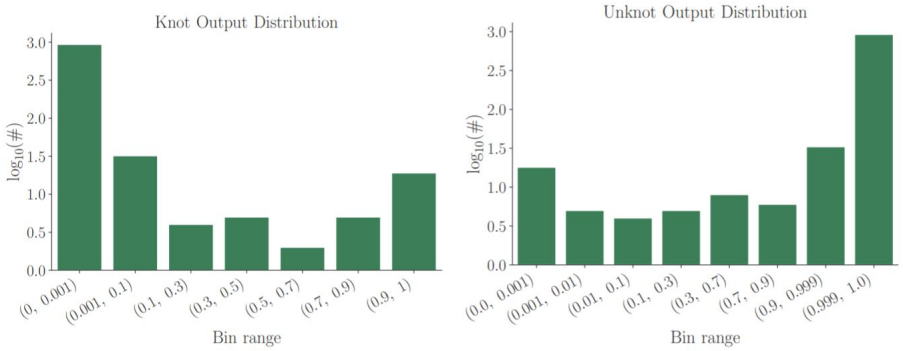
hardness of knot ***persists*** across diff. inits.

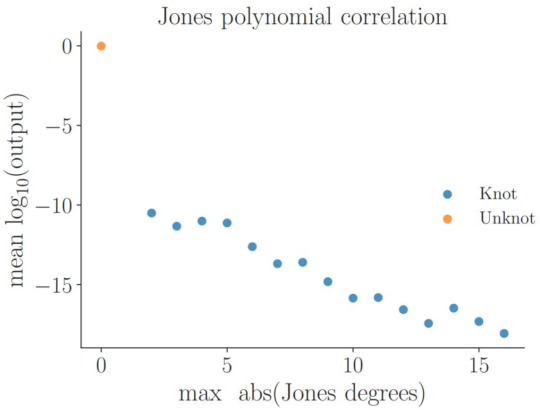
e.g. 1000 N<=9 test braids have 30 hard instances, 19 of which are trefoils, despite ~ ¼ knots being trefoils. **Knots with fewer crossings harder!?**

Jones Polynomial Correlations:

@ right, network confidence on correctly labelling knots correlated with Jones degree.

Jones not used in training at all! Learned feature.





Unknotting and Reinforcement

can reinforcement learning find the
sequence of moves that unknots?

Reinforcement Learning

- an *agent* interacts in an *environment*.
- it perceives a *state* from state space.
- its *policy* picks an action, given state.

$$\pi : \mathcal{S} \to \mathcal{A}$$

- arrives in new state, receives *reward*.
- successive rewards accum. to *return*. future rewards penalized by *discount*.

$$G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

- *state-* and *action-value* functions:

$$v(s) = \mathbb{E}[G_t | S_t = s]$$

$$q(s, a) = \mathbb{E}[G_t | S_t = s, A_t = a]$$

Famous Example: AlphaZero

"Mastering the game of Go without Human Knowledge"
- Silver et al, Nature 2017

RL with ***no human data.***

"A general reinforcement learning algorithms that masters chess, shogi, and Go through self-play."
- Silver at al, Science, 2018

Unknotting with RL

State Space:
zero-padded braids of length 2N.

Action Space: dim = N+5

1) shift left
2) shift right
3) BR1 and shift right
4) BR2 and shift right
5) Markov 1, conjugate by arb. gen.
6) SmartCollapse: destabilize and remove inverses until unchanged

Reward: negative braid length

End of game: empty braid or 300 moves.

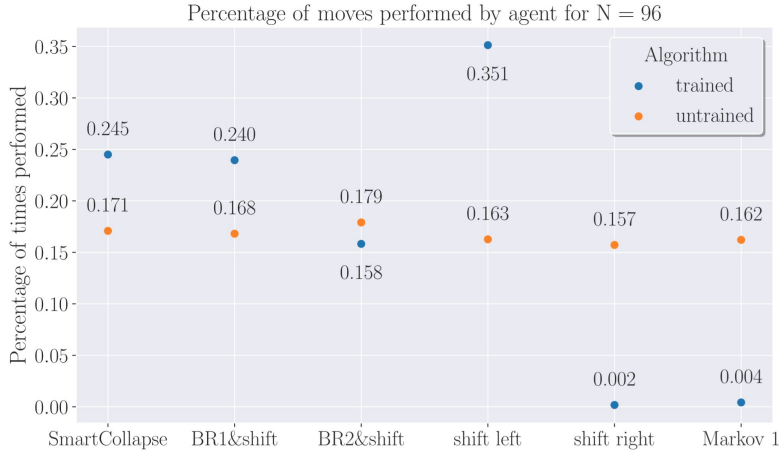
RL Algorithms:
A3C: asynch. advantage actor-critic, worker bees report back to 2 A/C nets.

TRPO: trust region policy optimization. Policy updates and steps depend on loss curvature.

Results + Interpretability

RL wins.
RW decreases rapidly.
TRPO crushes: flat in N.

Wins not just in % solved in <= 300 moves (below), but also in number of moves performed.



Interpretability via rollout.
i.e. learning is a flow in the state-dependent distribution on action space. **how does it change?**

SmartCollapse: only action that reduces N.

Shift left / right asymmetry b/c many shift rights come for free. **See paper for more interpretation!**

Example 1: Conclusions

NLP techniques natural for knot theory.
see e.g. transformers, reformers, GPT-3

Unknot Decision Problem
0) complexity results abound
1) defined a prior, mapped to Rolfsen

2) reformers do better than FFNN
3) but FFNN still do well! easy problem?
4) notions of hardness and Jones correlations arise naturally

Reinforcement Learning
0) agent learns to alter behavior, e.g. as in AlphaZero

1) environment: states are braids; actions are braid and Markov moves, composed; rewards are negative braid length; goal is to unknot the braid.

2) TRPO performance great, flat in N!
3) Interpretability via rollouts.

Example 2: Physics → ML

“Neural Networks and Quantum Field Theory”

[J.H., Maiti, Stoner] arXiv: 2008.08601

Related Non-Gaussian Process works:
[Dyer, Gur-Ari] 1909.11304 on bounding training
[Yaida] 1910.00019 on pre-activation distribution flow

What is learning?

Legend:

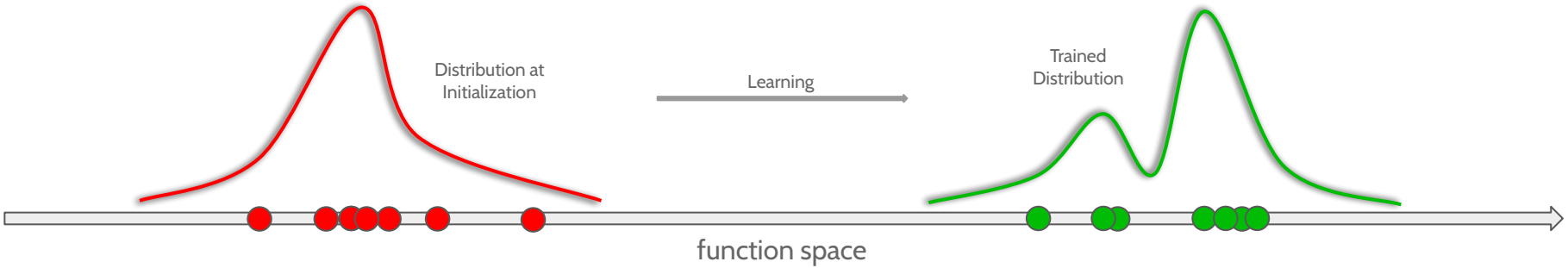
- Randomly initialized NN
- Trained NN

Physics Language:

Learning is a data-induced flow from an initialization function-space distribution to a trained distribution.

Bayesian Language:

Learning is approximating the posterior over functions given a prior and a likelihood.



Then what is supervised learning?

the evolution of the 1-pt function E[f] until convergence.

Example 2: Outline

- **What is QFT?** (physically? origin of Feynman diagrams. statistically?)

- **NN-QFT Correspondence:** model NN distributions with QFT techniques

  i) asymptotic NNs, GPs, and free field theory
  ii) NNs, non-GPs and Wilsonian “effective” field theory.
  iii) renormalization: removes divergences in higher correlators, simplifies NN dist.

- **Experiments:** A slide
- **Discussion and Outlook:** parameter-space / function-space duality, training

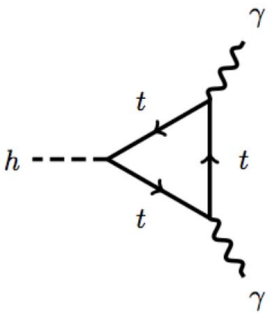
What is QFT?

physically?
what are Feynman diagrams?
statistically?

What is QFT, physically?

- quantum theory of fields,
  and their particle excitations.
- for both fundamental particles, (e.g. Higgs)
  and quasiparticles (e.g. in superconductors)
- **a single** QFT predicts radioactive decay rates,
  strength of particle scattering, etc.
- two main perspectives:
  "canonical quantization" (bra-ket approach)
  Feynman's path integral (today).
- **Many** Nobel prizes. (Could easily rattle off 5-10?)

Example: Higgs boson discovery



The QFT = Standard Model (SM) of Particle Phys.

2012: Discovered Higgs boson at CERN, e.g., in diphoton channel @ left.

Amazing science press.

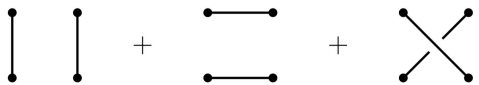
2013: Nobel to Higgs, Englert.

Origin of Feynman diagrams?

Pictures useful for computing moments of Gaussian or near-Gaussian distributions

Example: Gaussian Moments

$$\langle x^{2n} \rangle = \frac{\int_{-\infty}^{+\infty} dx\ \exp\left(-\frac{1}{2}ax^2\right) x^{2n}}{\int_{-\infty}^{+\infty} dx\ \exp\left(-\frac{1}{2}ax^2\right)} = \frac{1}{a^n}(2n-1)!!$$

$\langle x^4 \rangle$ = 

= $\left(\frac{1}{a}\cdot\frac{1}{a}\right) + \left(\frac{1}{a}\cdot\frac{1}{a}\right) + \left(\frac{1}{a}\cdot\frac{1}{a}\right) = \frac{3}{a^2}$

Feynman rules: a picture-expression dictionary

Example: Near-Gaussian Moments via Perturbation Theory

$$\langle x^{2n} \rangle = \frac{\int dx \exp(-\frac{1}{2}ax^2 + \lambda x^4)\, x^{2n}}{\int dx \exp(-\frac{1}{2}ax^2 + \lambda x^4)}$$

$$= \frac{\sum_{k=0}^{\infty} \frac{\lambda^k}{k!} \int dx \exp(-\frac{1}{2}ax^2)\, x^{2n+4k}}{\sum_{k=0}^{\infty} \frac{\lambda^k}{k!} \int dx \exp(-\frac{1}{2}ax^2)} \cdot \frac{\int dx \exp(-\frac{1}{2}ax^2)}{\int dx \exp(-\frac{1}{2}ax^2)}$$

$$= \frac{\sum_{k=0}^{\infty} \frac{\lambda^k}{k!} \langle x^{2n+4k} \rangle_G}{\sum_{k=0}^{\infty} \frac{\lambda^k}{k!} \langle x^{4k} \rangle_G}$$

for small λ, **truncate**

Additions and extra widgets may arise, but **Essence:** approximate non-Gaussian moments in terms of Gaussian moments, diagrammatically.

Sounds like QFT is physics widgets on top of a statistics backbone.

What is QFT, statistically?

- defined by distribution on field space, the so-called Feynman path integral. log-probability S[Φ] is “action”

$$Z = \int D\phi\, e^{-S[\phi]}$$

- Experiments measure n-pt correlation functions and amplitudes.

$$G^{(n)}(x_1, \ldots, x_n) = \frac{1}{Z} \int D\phi\ \phi(x_1) \ldots \phi(x_n)\ e^{-S[\phi]}$$

- **Free QFT:** no interactions, Gaussian.
- **Perturbative QFT:** distribution is near-Gaussian, compute approximate moments perturbatively.

NN-QFT Correspondence

i) asymptotic neural nets, GPs, and free QFT
ii) finite N neural nets, non-GPs, interacting QFT
iii) Wilsonian renormalization

A way to model NN distributions
using QFT techniques

Asymptotic Neural Networks

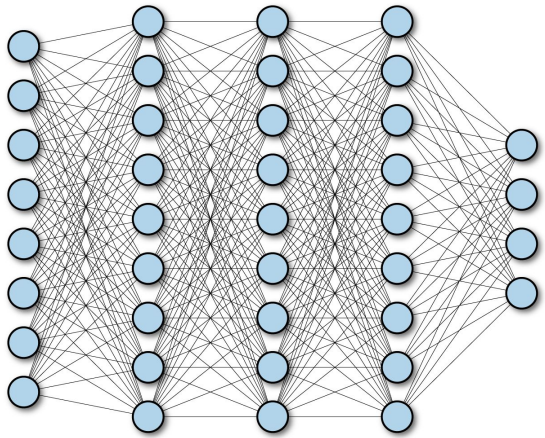
neural network has a discrete hyperparameter N
that enters into its architecture.

asymptotic limit = N $\longrightarrow \infty$ limit

crucial property: want to add infinite number of parameters, which themselves are random variables!

example:
infinite width limit of single-layer or deep feedforward networks

$$f_{\theta,N} : \mathbb{R}^{d_{\text{in}}} \longrightarrow \mathbb{R}^{d_{out}}$$



$$N \longrightarrow \infty$$

Simplest Example: Single-Layer Networks

A single-layer feedforward network is just

$$f_{\theta,N} : \mathbb{R}^{d_{\text{in}}} \xrightarrow{W_0, b_0} \mathbb{R}^N \xrightarrow{\sigma} \mathbb{R}^N \xrightarrow{W_1, b_1} \mathbb{R}^{d_{\text{out}}}$$

$$f(x) = W_1(\sigma(W_0 x + b_0)) + b_1$$

parameters drawn as $b_0, b_1 \sim \mathcal{N}(\mu_b, \sigma_b^2)$

$W_0 \sim \mathcal{N}(\mu_W, \sigma_W^2/d_{\text{in}})$ $W_1 \sim \mathcal{N}(\mu_W, \sigma_W^2/N)$

Limit of interest: infinite width N → ∞.

Then output adds an infinite number of i.i.d. entries from W_1 matrix, so CLT applies, output drawn from Gaussian!
Language: the neural network f is drawn from a ***Gaussian process***, i.e. Gaussian function-space distribution.

NN-GP Correspondence and Central Limit Theorem

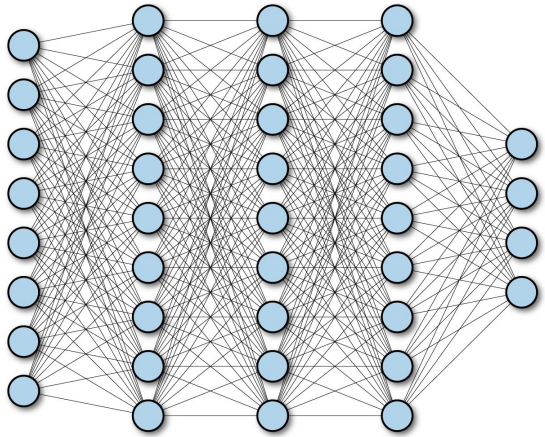
Add N iid random variables,
take N $\rightarrow \infty$,
sum is drawn from a Gaussian distribution.

If some step in a neural net does this,
that step drawn from Gaussian.

e.g., if NN output does, it's drawn from a Gaussian.

then NN is drawn from Gaussian distribution on field space, known as a *Gaussian Process* (GP).

$$f_{\theta,N} : \mathbb{R}^{d_{\text{in}}} \rightarrow \mathbb{R}^{d_{out}}$$



$$N \rightarrow \infty$$

“Most” architectures admit GP limit

Single-layer infinite width feedforward networks are GPs. [Neal], [Williams] 1990’s

Deep infinite width feedforward networks are GPs. [Lee et al., 2017], [Matthews et al., 2018]

Infinite channel CNNs are GPs. [Novak et al., 2018] [Garriga-Alonso et al. 2018]

Tensor programs show any *standard* architecture admits GP limit. [Yang, 2019]

> infinite channel limit [5, 6]. In [7, 8, 9], Yang developed a language for understanding which architectures admit GP limits, which was utilized to demonstrate that any standard architecture admits a GP limit, i.e. any architecture that is a composition of multilayer perceptrons, recurrent neural networks, skip connections [10, 11], convolutions [12, 13, 14, 15, 16] or graph convolutions [17, 18, 19, 20, 21, 22], pooling [15, 16], batch [23] or layer [24] normalization, and / or attention [25, 26]. Furthermore, though these results apply to randomly initialized neural networks, appropriately trained networks are also drawn from GPs [27, 28]. NGPs have been used to model finite neural networks in [29, 30, 31], with some key differences from our work. For these reasons, we believe that an EFT approach to neural networks is possible under a wide variety of circumstances.

tons of examples cited
in our paper admit GP limits

GP property persists under appropriate training. [Jacot et al., 2018] [Lee et al., 2019]

Gaussian Processes and Free Field Theory

Gaussian Process:

distribution: $$P[f] \sim \exp\left[-\frac{1}{2}\int d^{d_{\rm in}}x\, d^{d_{\rm in}}x' f(x)\Xi(x,x')f(x')\right]$$

where: $$\int d^{d_{\rm in}}x'\, K(x,x')\,\Xi(x',x'') = \delta^{(d_{\rm in})}(x-x'')$$

K is the *kernel* of the GP.

log-likelihood: $$S = \frac{1}{2}\int d^{d_{\rm in}}x\, d^{d_{\rm in}}x'\;\; f(x)\Xi(x,x')f(x')$$

n-pt correlation functions: $$G^{(n)}(x_1,\ldots,x_n) = \frac{\int df\;\; f(x_1)\ldots f(x_n)\, e^{-S}}{Z}$$

Crucial note:
P[f] can also have one or zero integrals, "local" and "ultra-local" cases, respectively.

Free Field Theory:

"free" = non-interacting
Feynman path integral:

$$Z = \int D\phi\, e^{-S[\phi]}$$

From P.I. perspective, free theories are Gaussian distributions on field space.

e.g., free scalar field theory

$$S[\phi] = \int d^d x\, \phi(x)(\Box + m^2)\phi(x)$$

GP / asymptotic NN	Free QFT
inputs $(x_1,\ldots,x_k)$	external space or spacetime points
kernel $K(x_1,x_2)$	Feynman propagator
asymptotic NN $f(x)$	free field
log-likelihood	free action $S_{\rm GP}$

GP Predictions for Correlation Functions

if asymptotic NN drawn from GP and GP "=" free QFT, should be able to use Feynman diagrams for correlation functions.

$$G^{(n)}(x_1, \ldots, x_n) = \frac{\int df \; f(x_1) \ldots f(x_n)\, e^{-S}}{Z}$$

Right: analytic and Feynman diagram expressions for n-pt correlations of asymptotic NN outputs.

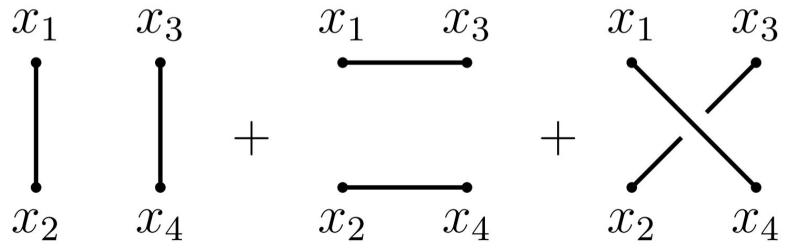
Physics analogy: mean-free GP is totally determined by 2-pt statistics, i.e. the GP kernel.

kernel = propagator, so GP = a QFT where all diagrams rep particles flying past each other.

$$G^{(2)}_{\mathrm{GP}}(x_1, x_2) = K(x_1, x_2)$$

$=$ 

$$G^{(4)}_{\mathrm{GP}}(x_1, x_2, x_3, x_4) = K(x_1, x_2)K(x_3, x_4) + K(x_1, x_3)K(x_2, x_4) + K(x_1, x_4)K(x_2, x_3)$$

$=$ 

What about finite N nets?

Non-Gaussian Processes (NGPs), EFTs, and Interactions

Punchline: finite N networks that admit a GP limit should be drawn from non-Gaussian process. (NGP)

$$S = S_{\mathrm{GP}} + \Delta S$$

where, e.g., could have a **model**:

$$\Delta S = \int d^{d_{\mathrm{in}}}x \ \left[g\, f(x)^3 + \lambda\, f(x)^4 + \alpha\, f(x)^5 + \kappa\, f(x)^6 + \dots \right]$$

such non-Gaussian terms are interactions in QFT.
their coefficients = "couplings."

NGP / finite NN	Interacting QFT
inputs $(x_1, \dots, x_k)$	external space or spacetime points
kernel $K(x_1, x_2)$	free or exact propagator
network output $f(x)$	interacting field
log probability	effective action S

Wilsonian EFT for NGPs:

- Determine the symmetries (or desired symmetries) respected by the system of interest.
- Fix an upper bound k on the dimension of any operator appearing in ΔS.
- Define ΔS to contain all operators of dimension $\leq k$ that respect the symmetries.

determines NGP "effective action" = log likelihood.
Some art in this, but done for decades by physicists.

Experiments below: single-layer finite width networks

$$S = S_{\mathrm{GP}} + \int d^{d_{\mathrm{in}}}x \ \left[\lambda\, f(x)^4 + \kappa\, f(x)^6 \right]$$

odd-pt functions vanish → odd couplings vanish.

κ is 1/N suppressed rel. λ, somes more irrelevant (Wilsonian sense), gives *even simpler NGP distribution.*

NGP Correlation Functions from Feynman Diagrams

Correlation functions defined by NGP distribution:

$$G^{(n)}(x_1, \ldots, x_n) = \frac{\int df \;\; f(x_1) \ldots f(x_n)\, e^{-S}}{Z_0}$$

use usual physics trick

$$= \frac{\int df \;\; f(x_1) \ldots f(x_n) \left[1 - \int d^{d_{\rm in}}x \, g_k f(x)^k + O(g_k^2)\right] \; e^{-S_{\rm GP}}/Z_{\rm GP,0}}{\int df \;\; \left[1 - \int d^{d_{\rm in}}x \, g_k f(x)^k + O(g_k^2)\right] e^{-S_{\rm GP}}/Z_{\rm GP,0}}$$

to compute diagrammatically as Feynman diagrams.

Essentials from QFT reviewed in paper,
e.g. cancellation of "vacuum bubbles" (components with no external points) by expanding the denominator.

Feynman Rules:

1) For each of the n external points x_i, draw 

.

2) For each y_j, draw 

. For each z_k, draw 

.

3) Determine all ways to pair up the loose ends associated to x_i's, y_j's, and z_k's. This will yield some number of topologically distinct diagrams. Draw them with dashed lines.

4) Write a sum over the diagrams with an appropriate combinatoric factor out front, which is the number of ways to form that diagram. Each diagram corresponds to an analytic term in the sum.

4.5) Throw away any diagram that has a component with a λ- or κ correction to the 2-pt function.

5) For each diagram, write $-\int d^{d_{\rm in}}y_j \, \lambda$ for each 

, and $-\int d^{d_{\rm in}}z_k \, \kappa$ for each 

.

6) Write $K(u, v)$ for each 

.

7) Throw away any terms containing vacuum bubbles.

these rules are a picture to analytic expression dictionary.

note: in our experiments, GP kernel happens to be exact all-width 2-pt function.

2-pt, 4-pt, and 6-pt Correlation Functions

point: theory equations that actually enter our NN codes.

$$G^{(2)}(x_1,x_2) = \quad - \lambda\Big[12 \quad x_1 \quad y \quad x_2\Big] - \kappa\Big[90 \quad x_1 \quad z \quad x_2\Big]$$

$$= \quad = K(x_1,x_2), \tag{3.17}$$

$$G^{(4)}(x_1,x_2,x_3,x_4) = 3 \quad - \lambda\Big[72 \quad y \quad + 24\Big]$$

$$- \kappa\Big[540 \quad z \quad + 360 \quad z\Big]$$

$$= 3 \quad - 24\,\lambda \quad y \quad - 360\,\kappa \quad z$$

$$= K(x_1,x_2)K(x_3,x_4) + K(x_1,x_3)K(x_2,x_4) + K(x_1,x_4)K(x_2,x_3)$$

$$- 24 \int d^{d_{\rm in}} y\, \lambda\, K(x_1,y)K(x_2,y)K(x_3,y)K(x_4,y)$$

$$- 360 \int d^{d_{\rm in}} z\, \kappa\, K(x_1,z)K(x_2,z)K(x_3,z)K(x_4,z)K(z,z) \tag{3.18}$$

$$G^{(6)}(x_1,x_2,x_3,x_4,x_5,x_6) = 15 \quad - \lambda\Big[540 \quad y \quad + 360 \quad y\Big]$$

$$- \kappa\Big[720 \quad z \quad + 5400 \quad z \quad + 4050 \quad z\Big]$$

$$= 15 \quad - 360\,\lambda \quad y \quad - \kappa\Big[720 \quad z \quad + 5400 \quad z\Big]$$

$$\begin{aligned}
&= \Big[K_{12}K_{34}K_{56} + K_{12}K_{35}K_{46} + K_{12}K_{36}K_{45} + K_{13}K_{24}K_{56} + K_{13}K_{25}K_{46} + K_{13}K_{26}K_{45} + K_{14}K_{23}K_{56} \\
&+ K_{14}K_{25}K_{36} + K_{14}K_{26}K_{35} + K_{15}K_{23}K_{46} + K_{15}K_{24}K_{36} + K_{15}K_{26}K_{34} + K_{16}K_{23}K_{45} + K_{16}K_{24}K_{35} \\
&+ K_{16}K_{25}K_{34}\Big] - 24 \int d^{d_{\rm in}} y\, \lambda \Big[K_{1y}K_{2y}K_{3y}K_{4y}K_{56} + K_{1y}K_{2y}K_{3y}K_{5y}K_{46} + K_{1y}K_{2y}K_{4y}K_{5y}K_{36} \\
&+ K_{1y}K_{3y}K_{4y}K_{5y}K_{26} + K_{2y}K_{3y}K_{4y}K_{5y}K_{16} + K_{1y}K_{2y}K_{3y}K_{6y}K_{45} + K_{1y}K_{2y}K_{4y}K_{6y}K_{35} \\
&+ K_{1y}K_{3y}K_{4y}K_{6y}K_{25} + K_{2y}K_{3y}K_{4y}K_{6y}K_{15} + K_{1y}K_{2y}K_{5y}K_{6y}K_{34} + K_{1y}K_{3y}K_{5y}K_{6y}K_{24} \\
&+ K_{2y}K_{3y}K_{5y}K_{6y}K_{14} + K_{1y}K_{4y}K_{5y}K_{6y}K_{23} + K_{2y}K_{4y}K_{5y}K_{6y}K_{13} + K_{3y}K_{4y}K_{5y}K_{6y}K_{12}\Big] \\
&- 720 \int d^{d_{\rm in}} z\, \kappa\, K_{1z}K_{2z}K_{3z}K_{4z}K_{5z}K_{6z} - 360 \int d^{d_{\rm in}} z\, \kappa \Big[K_{zz}K_{1z}K_{2z}K_{3z}K_{4z}K_{56} \\
&+ K_{zz}K_{1z}K_{2z}K_{3z}K_{5z}K_{46} + K_{zz}K_{1z}K_{2z}K_{4z}K_{5z}K_{36} + K_{zz}K_{1z}K_{3z}K_{4z}K_{5z}K_{26} \\
&+ K_{zz}K_{2z}K_{3z}K_{4z}K_{5z}K_{16} + K_{zz}K_{1z}K_{2z}K_{3z}K_{6z}K_{45} + K_{zz}K_{1z}K_{2z}K_{4z}K_{6z}K_{35} \\
&+ K_{zz}K_{1z}K_{3z}K_{4z}K_{6z}K_{25} + K_{zz}K_{2z}K_{3z}K_{4z}K_{6z}K_{15} + K_{zz}K_{1z}K_{2z}K_{5z}K_{6z}K_{34} \\
&+ K_{zz}K_{1z}K_{3z}K_{5z}K_{6z}K_{24} + K_{zz}K_{2z}K_{3z}K_{5z}K_{6z}K_{14} + K_{zz}K_{1z}K_{4z}K_{5z}K_{6z}K_{23} \\
&+ K_{zz}K_{2z}K_{4z}K_{5z}K_{6z}K_{13} + K_{zz}K_{3z}K_{4z}K_{5z}K_{6z}K_{12}\Big],
\end{aligned} \tag{3.19}$$

At this point you should object!

(very impressive attention to detail if you actually did.)

Input space integrals often diverge at large input.

QFT prescription: “regularization.”
Various varieties, we use a “hard cutoff” Λ, replace

$$S \to S_\Lambda$$

so any input integral is over a box of size Λ.

Making sense of divergences: Renormalization

Experiments: the central insight in renormalization.

[Zee] for beautiful textbook discussion.

Evaluate set of NNs on inputs

$$\mathcal{S}_{\text{in}} = \{x_1, \ldots, x_{N_{\text{in}}}\}$$
$$|x_i| \ll \Lambda$$

and measure experimental correlation functions,

$$G^{(n)}(x_1, \ldots, x_n) = \frac{1}{n_{\text{nets}}} \sum_{\alpha \in \text{nets}}^{n_{\text{nets}}} f_\alpha(x_1) \ldots f_\alpha(x_n)$$

tl

Goal of theory is to explain them.

Theory: NGP action corrects GP action by

$$\Delta S_\Lambda = \int_{-\Lambda}^{\Lambda} d^{d_{\text{in}}} x \sum_{l \le k} g_{\mathcal{O}_l}(\Lambda)\, \mathcal{O}_l$$

the old S had $\Lambda \to \infty$ and computing n-pt gives divergences. Λ finite regulates those divergences, input is now in a box.

For any Λ sufficiently big, measure couplings, make predictions, verify with experiments.

But there's an infinite number of S_Λ, and only one set of experiments for them to describe!

How does this make sense?

Essence of Renormalization

the infinity of effective actions must make
the same experimental predictions, requiring, e.g.

$$\frac{dG^{(n)}(x_1, \ldots, x_n)}{d\Lambda} = 0$$

Extracting β-functions from theory

NN effective actions (distributions) with different Λ may make the same predictions by absorbing the difference into couplings, ***"running couplings."***

$$\beta(g_{\mathcal{O}_l}) := \frac{d\, g_{\mathcal{O}_l}}{d\, log\Lambda}$$

Encoded in the β-functions, which capture how the couplings vary with the cutoff.

Induces a "flow" in coupling space as Λ varies, ***Wilsonian renormalization group flow***. (RG)

Extract from hitting n-pt functions with derivatives.

$$\frac{\partial G^{(4)}(x_1,x_2,x_3,x_4)}{\partial \log\Lambda} = 0 = \frac{\partial\lambda}{\partial\log\Lambda}\int_{-\Lambda}^{\Lambda} d^{d_{\rm in}}x\,(\gamma_{4,\lambda}+\varrho_{4,\lambda}) + \lambda\frac{\partial(\int_{-\Lambda}^{\Lambda} d^{d_{\rm in}}x\,(\gamma_{4,\lambda}+\varrho_{4,\lambda}))}{\partial\log\Lambda} + \frac{\partial\kappa}{\partial\log\Lambda}\int_{-\Lambda}^{\Lambda} d^{d_{\rm in}}x\,(\gamma_{4,\kappa}+\varrho_{4,\kappa}) + \kappa\frac{\partial(\int_{-\Lambda}^{\Lambda} d^{d_{\rm in}}x\,(\gamma_{4,\kappa}+\varrho_{4,\kappa}))}{\partial\log\Lambda}, \quad (4.13)$$

$$\frac{\partial G^{(6)}(x_1,x_2,x_3,x_4,x_5,x_6)}{\partial \log\Lambda} = 0 = \frac{\partial\lambda}{\partial\log\Lambda}\int_{-\Lambda}^{\Lambda} d^{d_{\rm in}}x\,(\gamma_{6,\lambda}+\varrho_{6,\lambda}) + \lambda\frac{\partial(\int_{-\Lambda}^{\Lambda} d^{d_{\rm in}}x\,(\gamma_{6,\lambda}+\varrho_{6,\lambda}))}{\partial\log\Lambda} + \frac{\partial\kappa}{\partial\log\Lambda}\int_{-\Lambda}^{\Lambda} d^{d_{\rm in}}x\,(\gamma_{6,\kappa}+\varrho_{6,\kappa}) + \kappa\frac{\partial(\int_{-\Lambda}^{\Lambda} d^{d_{\rm in}}x\,(\gamma_{6,\kappa}+\varrho_{6,\kappa}))}{\partial\log\Lambda} \quad (4.14)$$

Our examples:

κ more irrelevant than λ, in sense of Wilson.
Means as Λ gets large, κ goes to zero faster than λ, so you can ignore it.

Extract β-function for λ from deriv. of 4-pt.

A Flash of Some Experimental Results

Erf-net: $\sigma(z) = \mathrm{erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^z dt\, e^{-t^2}$ $\quad K_{\mathrm{Erf}}(x, x') = \sigma_b^2 + \sigma_W^2 \frac{2}{\pi} \arcsin \left[\frac{2(\sigma_b^2 + \frac{\sigma_W^2}{d_{\mathrm{in}}} x x')}{\sqrt{\left(1 + 2(\sigma_b^2 + \frac{\sigma_W^2}{d_{\mathrm{in}}} x^2)\right)\left(1 + 2(\sigma_b^2 + \frac{\sigma_W^2}{d_{\mathrm{in}}} x'^2)\right)}} \right]$

Gauss-net: $\sigma(x) = \frac{\exp(W x + b)}{\sqrt{\exp\left[2(\sigma_b^2 + \frac{\sigma_W^2}{d_{\mathrm{in}}} x^2)\right]}}$ $\quad K_{\mathrm{Gauss}}(x, x') = \sigma_b^2 + \sigma_W^2 \exp\left[-\frac{\sigma_W^2 |x - x'|^2}{2 d_{\mathrm{in}}} \right]$

ReLU-net: $\sigma(z) = \begin{cases} 0 & z < 0 \\ z & z \geq 0 \end{cases}$

$$K_{\mathrm{ReLU}}(x, x') = \sigma_b^2 + \sigma_W^2 \frac{1}{2\pi} \sqrt{(\sigma_b^2 + \frac{\sigma_W^2}{d_{\mathrm{in}}} x \cdot x)(\sigma_b^2 + \frac{\sigma_W^2}{d_{\mathrm{in}}} x' \cdot x')} (\sin\theta + (\pi - \theta)\cos\theta),$$

$$\theta = \arccos\left[\frac{\sigma_b^2 + \frac{\sigma_W^2}{d_{\mathrm{in}}} x \cdot x'}{\sqrt{(\sigma_b^2 + \frac{\sigma_W^2}{d_{\mathrm{in}}} x \cdot x)(\sigma_b^2 + \frac{\sigma_W^2}{d_{\mathrm{in}}} x' \cdot x')}} \right],$$

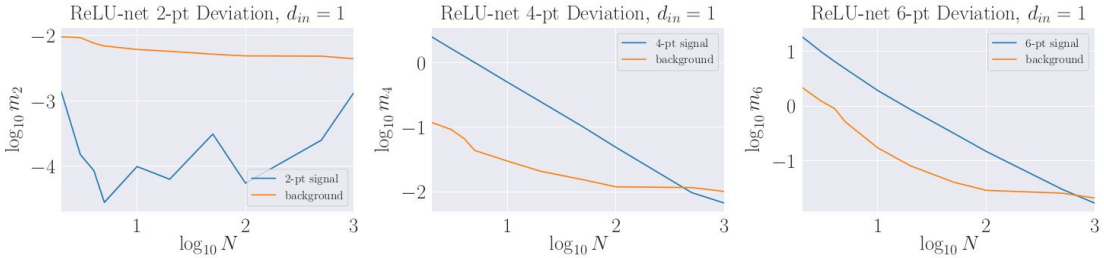
A Flash of Some Experimental Results

Experimental description

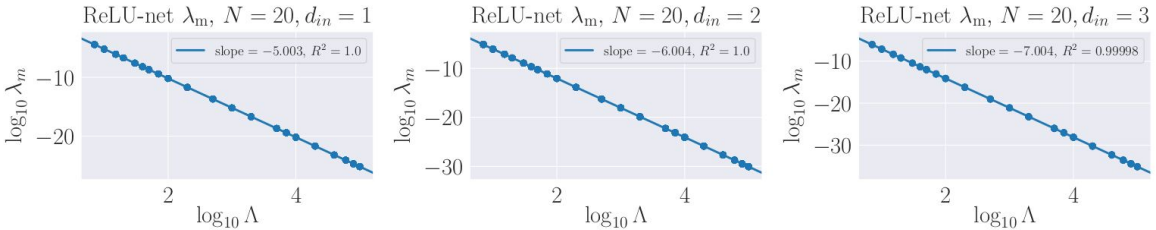
Experiments in three different single-layer networks, with ReLU, Erf, and a custom “GaussNet” activation.

Drew millions of models and evaluated on fixed sets of input to do experiments with correlators and the EFT description of NN distribution.

NGP correlators become GP correlators as N → ∞



Dependence of Quartic Coupling on Cutoff



$$\beta(\lambda) := \frac{\partial \lambda}{\partial \log \Lambda} = -\lambda\, d_{\text{in}}$$

Depends on input dimension.
See quartic is **asymptotically free.**

Verification of EFT Predictions

	$(\lambda_0, \lambda_2, \lambda_{\text{NL}})$	Test (MAPE, MSE)
Gauss M_0	$(0.0, 0.0, 0.0)$	$(100, 0.019)$
Gauss M_1	$(0.0046, 0.0, 0.0)$	$(0.0145, 6.8 \times 10^{-10})$
Gauss M_2	$(0.0043, 0.0011, 0.0)$	$(0.0144, 6.7 \times 10^{-10})$
Gauss M_3	$(0.00062, 0.00016, 0.0015)$	$(0.0156, 7.5 \times 10^{-10})$
ReLU M_0	$(0.0, 0.0, 0.0)$	$(100, 0.003)$
ReLU M_1	$(6.2 \times 10^{-11}, 0.0, 0.0)$	$(0.0035, 7.6 \times 10^{-12})$
ReLU M_2	$(1.2 \times 10^{-18}, 8.7 \times 10^{-15}, 0.0)$	$(0.0013, 1.5 \times 10^{-12})$
ReLU M_3	$(1.2 \times 10^{-18}, 8.7 \times 10^{-15}, 6.8 \times 10^{-17})$	$(0.0012, 1.2 \times 10^{-12})$
Erf M_0	$(0.0, 0.0, 0.0)$	$(100, 0.006)$
Erf M_1	$(0.039, 0.0, 0.0)$	$(0.030, 8.3 \times 10^{-10})$
Erf M_2	$(0.040, -0.00043, 0.0)$	$(0.0042, 1.9 \times 10^{-11})$
Erf M_3	$(0.0019, -0.0054, 0.0063)$	$(0.037, 1.1 \times 10^{-9})$

Test / train split on connected 4-pt function to verify predictions of measured couplings.

Example 2: Discussion and Outlook

summary,
parameter-space / function-space duality,
supervised learning in QFT language

Example 2: Summary

asymptotic NN's "=" Free QFT

GP / asymptotic NN	Free QFT
inputs $(x_1, \ldots, x_k)$	external space or spacetime points
kernel $K(x_1, x_2)$	Feynman propagator
asymptotic NN $f(x)$	free field
log-likelihood	free action S_{GP}

b/c drawn from GPs

NNs "=" QFT

NGP / finite NN	Interacting QFT
inputs $(x_1, \ldots, x_k)$	external space or spacetime points
kernel $K(x_1, x_2)$	free or exact propagator
network output $f(x)$	interacting field
log probability	effective action S

b/c drawn from NGPs

central idea: model NGP / NN distribution using Wilsonian effective field theory. (EFT)

fairly general: any "standard architecture" (Yang) admits a GP limit. persists under some training.

therefore, away from limit, NGP. use EFT to model. import QFT ideas directly into NNs.

EFT treatment of NN distribution yields:

1) output correlation functions as Feynman diagrams.
2) measure some couplings (non-Gaussian coeffs) in experiments, predict, verify in experiments.
3) Wilsonian RG induces flow in couplings, simplifies the model of the NN distribution.

Verified all of this experimentally, single layer networks, indeed QFT gives function-space perspective on NNs.

Example 2: Gains in Perspective

Duality:

In physics, means two perspectives on a single system, where certain things are easier from one.

Parameter-space / function-space duality:
at large N, parameter-space complexity explodes.

but in function-space complexity decreases due to renorm. and 1/N suppression of non-Gaussianities.

Acute example: *single number* in NGP dist. was sufficient to approximate NGP 6-pt corrections, despite losing an ∞ number of params in moving from GP.

Training:

Our formalism only requires being "close" to GP, where measure of closeness determined experimentally and in examples is relatively low N.

Some training preserves GP at large N, in principle allowing QFT treatment of NGP during training.

Supervised learning:
in QFT language, it is just learning the 1-pt function.

in general this will break symmetry of NGP (see paper next week for priors), bring in even more QFT.

Summing up the whole talk

Overarching Conclusions

Example 1: ML → Math

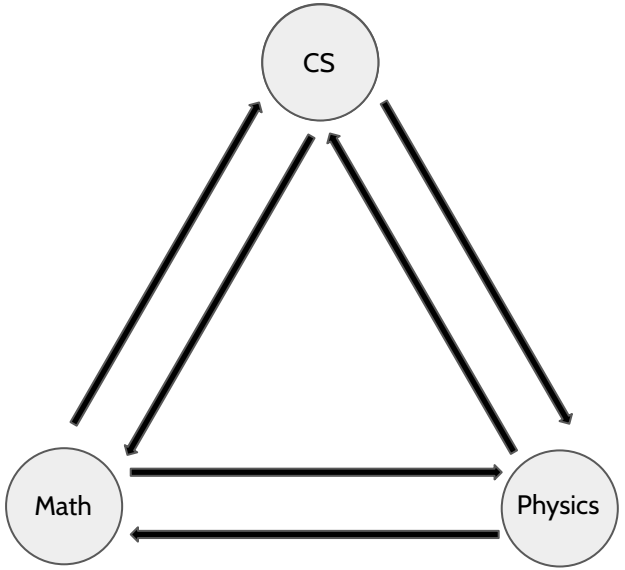
Natural language for *deciding* the unknot.
Reinforcement learning for actually unknotting!
Interpretability via rollouts.

Example 2: Physics → ML

Neural nets are drawn from near-Gaussians.

But near-Gaussian distributions are the backbone of perturbative QFT.

Directly import techniques from QFT to NNs.



Towards a triangle of influence?

Physics ↔ Math for ***millenia***.

CS is in its ***infancy***,
but it is different and powerful.

Will it become the third vertex?

Thanks!

Questions?

And seriously, feel free to get in touch:

e-mail: jhh@neu.edu
Twitter: @jhhalverson
web: www.jhhalverson.com