Oxford Lectures on Physics for AI

Jim Halverson

Department of Physics, Northeastern University, Boston, MA 02115, USA

*The NSF Institute for Artificial Intelligence
and Fundamental Interactions*

jhh@neu.edu

Abstract

These notes were used for lectures at a Physics for AI meeting at the University of Oxford, March 2025. The first lecture focuses on applications of field theory techniques in ML, including why field theory language is unavoidable; the statistics of neural networks, relation to generalized free fields, and non-Gaussianities from $1/N$-corrections; and the dynamics of neural networks, including the neural tangent kernel and feature learning. The second lecture focuses on a neural network approach to field theory, including when and why cherished principles from field theory arise, including interactions, conformal symmetry, and unitarity. I will explain why neural networks provide a universal language for quantum mechanical systems and demonstrate some classic results, such as Heisenberg uncertainty.

Contents

1 Introduction

It is a remarkable time in computer science, forged by numerous advances in machine learning. Classical examples include the use of reinforcement learning to train world-class agents at Go and Chess [1], the use of diffusion models to generate high quality images [2, 3], and the use of large language models to generate exceptional code and text, e.g. [4]. Some think we have seen sparks [5] of AGI or human-level intelligence. In total, these advances have led to a trillion dollar industry, many breakthroughs in the natural sciences, and Nobel prizes in Chemistry and Physics.

Notably, the 2024 Nobel Prize in Physic was not for the application of machine learning to cutting edge physics problems, but instead of applications of physics principles within ML. This “Physics of Learning” is a growing field comprised of physicists and computer scientists from many different subfields, with a common goal to utilize the tools of physics to understand the principles of ML.

Why should physics and ML have anything to do with each other? A field theoretic lens on the question is the subject of these lectures, which were given at a Physics for AI Workshop at the University of Oxford in March 2025. The lectures focus on the emergence of field theory in machine learning, and the use of neural networks as a way to define and understand field theories. Much of the material is adapted and updated from my 2024 TASI lectures [6] on Physics for Machine Learning, but material related to network expressivity is omitted, including universal approximation theorem, Kolomogorov-Arnold representiation theorem, and associated network architctures.

For me, the interplay between field theory and ML cuts both ways. Accordingly, one lecture will be for ML, and the other for physics.

Acknowledgements: I would like to thank Ard Louis, Andre Lukas, Shivaji Sondhi, and especially Andrei Constantin for organizing this meeting, the participants and speakers for a lively meeting, and many friends whose work inspired these lectures. I am supported by the National Science Foundation under CAREER grant PHY-1848089 and Cooperative Agreement PHY-2019786 (The NSF AI Institute for Artificial Intelligence and Fundamental Interactions).

2 Field Theory for Machine Learning

In this lecture I'll explain how field theoretic concepts emerge naturally in ML, and present a number of classic associated results on the statistics and dynamics of neural networks.

2.1 Why Field Theory?

Understanding ML at the very least means understanding neural networks. A neural network is a function

$$\phi_\theta : \mathbb{R}^d \to \mathbb{R} \tag{1}$$

with parameters θ. We've chosen outputs in $\mathbb{R}$ because, channeling Coleman, scalars already exhibit the essentials. We'll use the lingo

$$\begin{aligned} \text{Input:} &\quad x \in \mathbb{R}^d && (2)\\ \text{Output:} &\quad \phi_\theta(x) \in \mathbb{R} && (3)\\ \text{Network:} &\quad \phi_\theta \in \text{Maps}(\mathbb{R}^d, \mathbb{R}) && (4)\\ \text{Data:} &\quad \mathcal{D}, && (5) \end{aligned}$$

where the data $\mathcal{D}$ depends on the problem, but involves at least a subset of $\mathbb{R}^d$, potentially paired with labels $y \in \mathbb{R}$.

With this minimal background, let's ask our central question:

Question: What does a NN predict?

For any fixed value of θ, the answer is clear: $\phi_\theta(x)$. However, the answer is complicated by issues of both dynamics and statistics.

First, **dynamics**. In ML, parameters are updated to solve problems and we really have **trajectories** in

$$\begin{aligned} \text{Parameter Space:} &\quad \theta(t) \in \mathbb{R}^{|\theta|} && (6)\\ \text{Output Space:} &\quad \phi_{\theta(t)}(x) \in \mathbb{R} && (7)\\ \text{Function Space:} &\quad \phi_{\theta(t)} \in \text{Maps}(\mathbb{R}^d, \mathbb{R}). && (8) \end{aligned}$$

governed by some learning dynamics determined by the optimization algorithm and the nature of the learning problem. For instance, in supervised learning we have data

$$\mathcal{D} = \{(x_\alpha, y_\alpha) \in \mathbb{R}^d \times \mathbb{R}\}_{\alpha=1}^{|\mathcal{D}|}, \tag{9}$$

and a loss function

$$\mathcal{L}[\phi_\theta] = \sum_{\alpha=1}^{|\mathcal{D}|} \ell(\phi_\theta(x_\alpha), y_\alpha), \tag{10}$$

where ℓ is a loss function such as $\ell_{\text{MSE}} = (\phi_\theta(x_\alpha) - y_\alpha)^2$. One may optimize θ by gradient descent

$$\frac{d\theta_i}{dt} = -\nabla_{\theta_i} \mathcal{L}[\phi_\theta], \tag{11}$$

or other algorithms, e.g., classics like stochastic gradient descent (SGD) [7, 8] or Adam [9], or a more recent technique such as Energy Conserving Descent [10, 11]. Throughout, t is training time of the learning algorithm unless otherwise noted.

Second, **statistics**. When a NN is initialized on your computer, the parameters θ are initialized as draws

$$\theta \sim P(\theta) \tag{12}$$

from a distribution $P(\theta)$, where $\sim$ means "drawn from" in this context. Different draws of θ will give different functions ϕ_θ, and a priori we have no reason to prefer one over another. The prediction $\phi_\theta(x)$ therefore can't be fundamental! Instead, what is fundamental is the average prediction and second moment or variance:

$$\mathbb{E}[\phi_\theta(x)] = \int d\theta P(\theta)\, \phi_\theta(x) \tag{13}$$

$$\mathbb{E}[\phi_\theta(x)\phi_\theta(y)] = \int d\theta P(\theta)\, \phi_\theta(x)\phi_\theta(y), \tag{14}$$

as well as the higher moments. Expectations are across different initializations. Since we're physicists, we henceforth replace $\mathbb{E}[\cdot] = \langle \cdot \rangle$ and we remember this is a statistical expectation value. It's useful to put this in our language:

$$G^{(1)}(x) = \langle \phi_\theta(x) \rangle \tag{15}$$

$$G^{(2)}(x, y) = \langle \phi_\theta(x)\phi_\theta(y) \rangle, \tag{16}$$

the mean prediction and second moment are just the one-point and two-point correlation functions of the statistical ensemble of neural networks. Apparently ML has something to do with field theory.

Putting the dynamics and statistics together, we have an ensemble of initial θ-values, each of which is the starting point of a trajectory $\theta(t)$, and therefore we have an ensemble of trajectories. We choose to think of $\theta(t)$ drawn as

$$\theta(t) \sim P(\theta(t)), \tag{17}$$

a density on parameters that depends on the training time and yields time-dependent correlators

$$G_t^{(1)}(x) = \langle \phi_\theta(x) \rangle_t \tag{18}$$

$$G_t^{(2)}(x, y) = \langle \phi_\theta(x)\phi_\theta(y) \rangle_t, \tag{19}$$

where the subscript t indicates time-dependence and the expectation is with respect to $P(\theta(t))$. Of course, assuming that learning is helping, we wish to take $t \to \infty$ and are interested in

$$G_\infty^{(1)}(x) = \text{mean prediction of } \infty\text{-number of NNs as } t \to \infty.$$

Remarkably, we will see that in a certain supervised setting there is an exact analytic solution for this quantity.

Having set the basic groundwork, in the remainder of this lecture we will take a deeper dive into the statistics and dynamics of neural networks, and the emergence of field theoretic concepts.

2.2 Statistics of Neural Networks

Let's try to understand neural networks at initialization. For this, firing up your computer once is not enough, since one initialization give you one draw $\theta \sim P(\theta)$ and therefore one random neural network. Instead, we want to understand the *statistics* of the networks:

Question: What characterizes the stats of the NN ensemble?

One aspect of this is encoded in the moments, or n-pt functions,

$$G^{(n)}(x_1, \dots, x_n) = \langle \phi(x_1) \dots \phi(x_n) \rangle, \tag{20}$$

which may be obtained from a partition function as

$$Z[J] = \langle e^{\int d^dx J(x)\phi(x)} \rangle \tag{21}$$

$$G^{(n)}(x_1, \dots, x_n) = \left(\frac{\delta}{\delta J(x_1)} \dots \frac{\delta}{\delta J(x_n)} Z[J] \right) \bigg|_{J=0}, \tag{22}$$

where $J(x)$ is a source. This expectation $\langle \cdot \rangle$ is intentionally not specified here to allow for flexibility. For instance, using the expectation in the introduction we have

$$Z[J] = \int d\theta P(\theta) e^{\int d^dx J(x)\phi(x)}, \tag{23}$$

reminding the reader that the NN $\phi(x)$ depends on θ. The partition function integrates over the density of network parameters. But as physicists we're much more familiar with function space densities according to

$$Z[J] = \int \mathcal{D}\phi \, e^{-S[\phi]} e^{\int J(x)\phi(x)}, \tag{24}$$

the Feynman path integral that determines the correlators from an action $S[\phi]$ that defines a density on functions.

Since starting a neural network requires specifying the data $(\phi, P(\theta))$, the *parameter space partition function* (23) and associated parameter space calculation of correlators is always available to us. Given that mathematical data, one might ask

Question: What is the action $S[\phi]$ associated to $(\phi, P(\theta))$?

When this question can be answered, it opens a second way of studying or understanding the theory. The parameter-space and function-space descriptions should be thought of as a **duality**.

2.2.1 NNGP Correspondence

Having raised the question of the action $S[\phi]$ associated to the network data $(\phi, P(\theta))$, we can turn to a classic result of Neal [12].

For simplicity, we again consider a single-layer fully connected network of width N, with the so-called biases turned off for simplicity:

$$\phi(x) = \frac{1}{\sqrt{N}} \sum_{i=1}^{N} \sum_{j=1}^{d} w_i^{(1)} \sigma(w_{ij}^{(0)} x_j), \tag{25}$$

where the set of network parameters is $\theta = \{w_{ij}^{(0)}, w_i^{(1)}\}$ independently and identically distributed (i.i.d.).

$$w_{ij}^{(0)} \sim P(w^{(0)}) \qquad w_i^{(1)} \sim P(w^{(1)}). \tag{26}$$

Under this assumption, we see

Observation: The network is a sum of N i.i.d. functions.

This is a function version of the Central Limit Theorem, generalizing the review in Appendix A, and gives us the Neural Network / Gaussian Process (NNGP) correspondence,

NNGP Correspondence: in the $N \to \infty$ limit, ϕ is drawn from a Gaussian Process (GP),

$$\lim_{N\to\infty} \phi(x) \sim \mathcal{N}\left(\mu(x), K(x,y)\right), \tag{27}$$

with mean and covariance (or kernel) $\mu(x)$ and $K(x,y)$.

By the CLT, $\exp(-S[\phi])$ is Gaussian and therefore $S[\phi]$ is quadratic in networks. Now this really feels like physics, since the infinite neural network is drawn from a Gaussian density on functions, which defines a generalized free field theory.

We will address generality of the NNGP correspondence momentarily, but let's first get a feel for how to do computations. To facilitate then, we take $P(w^{(1)})$ to have zero mean and finite variance,

$$\langle w^{(1)} \rangle = 0 \qquad \langle w^{(1)} w^{(1)} \rangle = \mu_2, \tag{28}$$

which causes the one-point function to vanish $G^{(1)}(x) = 0$. Following Williams [13], we compute the two-point function in parameter space (with Einstein summation)

$$G^{(2)}(x, y) = \frac{1}{N}\langle w_i^{(1)}\sigma(w_{ij}^{(0)}x_j)\; w_k^{(1)}\sigma(w_{kl}^{(0)}y_l)\rangle \tag{29}$$

$$= \frac{1}{N}\,\langle w_i^{(1)}w_k^{(1)}\rangle\langle\sigma(w_{ij}^{(0)}x_j)\sigma(w_{kl}^{(0)}y_l)\rangle \tag{30}$$

$$= \frac{\mu_2}{N}\langle\sigma(w_{ij}^{(0)}x_j)\,\sigma(w_{il}^{(0)}y_l)\rangle, \tag{31}$$

where the last equality follows from the ones being i.i.d., $\langle w_i^{(1)}w_k^{(1)}\rangle = \mu_2\delta_{ik}$. The sum over i gives us N copies of the same function, leaving us with

$$G^{(2)}(x, y) = \mu_2\;\langle\sigma(w_{ij}^{(0)}x_j)\,\sigma(w_{il}^{(0)}y_l)\rangle, \tag{32}$$

where we emphasize there is now *no summation on* i. This is an exact-in-N two-point function that now requires only on the computation of the quantity in bra-kets. One may try to evaluate it exactly by doing the integral over $w^{(0)}$. If it can't be done, Monte Carlo estimates may be obtained from M samples of $w^{(0)} \sim P(w^{(0)})$ as

$$G^{(2)}(x, y) \simeq \frac{\mu_2}{M}\sum_{\text{samples}}^{M}\sigma(w_{ij}^{(0)}x_j)\,\sigma(w_{il}^{(0)}y_l). \tag{33}$$

In typical NN settings, parameter densities are easy to sample for convenience, allowing for easy computation of the estimate. If the density is more complicated, one may always resort to Markov chains, e.g. as in lattice field theory.

With this computation in hand, we have the defining data of this NNGP,

$$\lim_{N\to\infty}\;\phi(x) \sim \mathcal{N}\left(0, G^{(2)}(x, y)\right). \tag{34}$$

The associated action is

$$S[\phi] = \int d^dx d^dy\,\phi(x)\,G^{(2)}(x, y)^{-1}\,\phi(y), \tag{35}$$

where

$$\int d^dy\,G^{(2)}(x, y)^{-1}G^{(2)}(y, z) = \delta^{(d)}(x - z). \tag{36}$$

defines the inverse two-point function. In fact, this allows us to determine the action of any NNGP with $\mu(x) = G^{(1)}(x) = 0$, by computing the $G^{(2)}$ in parameter space and inverting it.

So certain large neural networks are function draws from generalized free field theories. But at this point you might be asking yourself

Question: How general is the NNGP correspondence?

Neal's result — that infinite-width single-layer feedforward NNs are drawn from GP — stood on its own for many years, perhaps (I am guessing) due to focus on non-NN ML techniques in the 90's and early 2000's during a so-called AI Winter. As NNs succeeded on many tasks in the 2010's after AlexNet [14], however, many asked whether architecture X has a hyperparameter N such that the network is drawn from a Gaussian Process as $N \to \infty$. Before listing such X's, let's rhetorically ask

Question: Didn't Neal's result essentially follow from summing N i.i.d. random functions? Maybe NNs do this all the time?

In fact, that is the case. Architectures admitting an NNGP limit include

- **Deep Fully Connected Networks,** $N =$ width,
- **Convolutional Neural Networks,** $N =$ channels,
- **Attention Networks,** $N =$ heads,

and many more. See, e.g., [15] and references therein.

2.2.2 Non-Gaussian Processes

If the GP limit exists due to the CLT, then violating any of the assumptions of the CLT should introduce non-Gaussianities, which are interactions in field theory. From Appendix A, we see that the CLT is violated by finite-N corrections and breaking statistical independence. See [16] for a systematic treatment of independence breaking, and derivation of NN actions from correlators.

We wish to see the N-dependence of the connected 4-pt function. William's technique for computing $G^{(2)}$ extends to any correlator. To avoid a proliferation of indices, we will compute it using the notation

$$\phi(x) = \sum_i w_i \varphi_i(x) \tag{37}$$

where w_i is distributed as $w^{(1)}$ was in the single layer case, and $\varphi_i(x)$ are i.i.d. neurons of *any* architecture. The four-point function is

$$\begin{aligned} G^{(4)} &= \langle \phi(x)\phi(y)\phi(z)\phi(w)\rangle && (38)\\ &= \sum_{i,j,k,l} \langle w_i w_j w_k w_l\rangle \langle \varphi_i(x)\varphi_j(y)\varphi_k(z)\varphi_l(w)\rangle && (39)\\ &= \sum_i \langle w_i^4\rangle \langle \varphi_i(x)\varphi_i(y)\varphi_i(z)\varphi_i(w)\rangle && (40)\\ &\quad + \sum_{i\neq j} \langle w_i^2\rangle\langle w_j^2\rangle \langle \varphi_i(x)\varphi_i(y)\varphi_j(z)\varphi_j(w) + \text{perms}\rangle. && (41)\end{aligned}$$

One can see that you have to be careful with indices. The connected 4-pt function is [17]

$$G_c^{(4)}(x,y,z,w) = G^{(4)}(x,y,z,w) - \left(G^{(2)}(x,y)G^{(2)}(z,w) + \text{perms}\right), \tag{42}$$

and watching indices carefully we obtain

$$\begin{aligned} G_c^{(4)}(x,y,z,w) = \frac{1}{N}\Big(&\mu_4 \left\langle \varphi_i(x)\varphi_i(y)\varphi_i(z)\varphi_i(w)\right\rangle && (43)\\ &-\mu_2^2\left(\big\langle \varphi_i(x)\varphi_i(y)\big\rangle\langle \varphi_i(z)\varphi_i(w)\rangle + \text{perms}\right)\Big), && (44)\end{aligned}$$

with no Einstein summation on i. We see that the connected 4-pt function is non-zero at finite-N, signalling interactions. We will see that in some examples $G_c^{(4)}$ can be computed exactly.

2.3 Dynamics of Neural Networks

Having covered expressivity and statistics, we turn to dynamics. Focusing on the most elementary NN dynamics, we ask

Question: How does a NN evolve under gradient descent?

First we will study a simplification known as the neural tangent kernel (NTK), and then will use it in the case of MSE loss to solve a model exactly. We'll discuss drawbacks of the NTK, and then improve upon them with a scaling analysis that ensures feature learning.

We will study the dynamics of supervised learning with gradient descent, with data

$$\mathcal{D} = \{(x_\alpha, y_\alpha)\}_{\alpha=1}^{|\mathcal{D}|}, \tag{45}$$

and loss function

$$\mathcal{L}[\phi] = \frac{1}{|\mathcal{D}|}\sum_{\alpha=1}^{|\mathcal{D}|} \ell(\phi(x_\alpha), y_\alpha), \tag{46}$$

We optimize the network parameters θ by gradient descent,

$$\frac{d\theta_i}{dt} = -\eta \nabla_{\theta_i}\mathcal{L}[\phi]. \tag{47}$$

It is also convenient to define

$$\Delta(x) = -\frac{\delta \ell(\phi(x), y)}{\delta\phi(x)}, \tag{48}$$

where y is to be understood as the label associated to x, which yields

$$\frac{d\theta_i}{dt} = \frac{\eta}{|\mathcal{D}|}\sum_{\alpha=1}^{|\mathcal{D}|} \Delta(x_\alpha)\frac{\partial\phi(x_\alpha)}{\partial\theta_i}. \tag{49}$$

as another form of the gradient descent equation, by chain rule. $\Delta(x)$ is the natural object of gradient descent in function space.

We use Einstein summation throughout this section unless stated otherwise (which will happen).

2.3.1 Neural Tangent Kernel

We now arrive at a classic dynamical result in ML theory, the neural tangent kernel [18]. We train the network by gradient descent

$$\begin{aligned} \frac{d\phi(x)}{dt} &= \frac{\partial \phi(x)}{\partial \theta_i}\frac{d\theta_i}{dt} \qquad (50) \\ &= \frac{\eta}{|\mathcal{D}|}\sum_{\alpha=1}^{|\mathcal{D}|} \Delta(x_\alpha)\,\Theta(x,x_\alpha), \qquad (51) \end{aligned}$$

where

$$\Theta(x,x_\alpha) = \frac{\partial \phi(x)}{\partial \theta_i}\frac{\partial \phi(x_\alpha)}{\partial \theta_i} \qquad (52)$$

is the *neural tangent kernel* (NTK).

Given the short derivation, this is clearly a fundamental object, but it seems terrible to work with, because it is:

- **Parameter-dependent.** For modern NNs, there are billions of parameters to sum over.
- **Time-dependent.** Of course, the learning trajectory is $\theta(t)$, and therefore the NTK time evolves.
- **Stochastic.** Since $\theta(t)$ begins at $\theta(0)$ sampled at initialization, the NTK inherits the randomness.

It's also non-local, communicating information about the loss at train points x_α to the test point x. In summary, the NTK is unwieldy.

The reason it is a classic result, however, is that it simplifies in the $N \to \infty$ limit. In this limit, neural network training is in the so-called

$$\text{Lazy regime:} \qquad |\theta(t) - \theta(0)| \ll 1, \qquad (53)$$

i.e. the number of parameters is large but the evolution keeps them in a local neighborhood. In such a regime the network is approximately a linear-in-parameters model [18, 19]

$$\lim_{N\to\infty} \phi(x) \simeq \phi_{\text{lin}}(x) := \phi_{\theta_0}(x) + (\theta - \theta_0)_i \frac{\partial \phi(x)}{\partial \theta_i}\bigg|_{\theta_0}, \qquad (54)$$

and we have

$$\lim_{N\to\infty} \Theta(x,x') \simeq \Theta(x,x')\big|_{\theta_0}. \qquad (55)$$

That is, the infinite-width NTK is the NTK at initialization, provided that the network evolves as a linear model. Furthermore, in the same limit the law of large numbers often allows a sum to be replaced by an expectation value, e.g.,

$$\lim_{N\to\infty} \Theta(x,x')|_{\theta_0} = \langle \beta_\theta(x,x')\rangle =: \bar{\Theta}(x,x'), \qquad (56)$$

for computable $\beta(x,x')$, yielding network dynamics governed by

$$\frac{d\phi(x)}{dt} = -\frac{\eta}{|\mathcal{D}|}\sum_{\alpha=1}^{|\mathcal{D}|} \frac{\delta l(\phi(x_\alpha),y_\alpha)}{\delta \phi(x_\alpha)}\ \bar{\Theta}(x,x_\alpha), \qquad (57)$$

where $\bar{\Theta}$ is the so-called *frozen NTK*, a kernel that may be computed at initialization and fixed once-and-for-all. This is a dramatic simplification of the dynamics.

However, you should also complain.

> **Complaint:** The dynamics in (57) simply interpolates between information at train points x_α and test point x,

according to a fixed function $\bar{\Theta}$. This isn't "learning" in the usual NN sense, and there are *zero* parameters. In particular, since the NN only affects (57) through $\bar{\Theta}$, which is fixed, nothing happening dynamically in the NN is affecting the evolution. We say that in this limit the NN **does not learn features** in the hidden dimensions (intermediate layers), since their non-trivial evolution would cause the NTK to evolve.

Example. Let's compute the frozen NTK for a single-layer network, to get the idea. The architecture is

$$\phi(x) = \frac{1}{\sqrt{N}}\sum_{i=1}^{N}\sum_{j=1}^{d} w_i^{(1)}\sigma(w_{ij}^{(0)}x_j). \qquad (58)$$

We make the sums explicit here because one is very important. The NTK is

$$\Theta(x,x') = \sum_i \frac{\partial\phi(x)}{\partial w_i^{(1)}}\frac{\partial\phi(x')}{\partial w_i^{(1)}} + \sum_{ij}\frac{\partial\phi(x)}{\partial w_{ij}^{(0)}}\frac{\partial\phi(x')}{\partial w_{ij}^{(0)}} \tag{59}$$

$$= \frac{1}{N}\sum_{i=1}^{N}\Bigg(\sum_{j,l=1}^{d}\sigma(w_{ij}^{(0)}x_j)\sigma(w_{il}^{(0)}x_l') \tag{60}$$

$$+\sum_{j=1}^{d} x_j x_j'\, w_i^{(1)} w_i^{(1)}\, \sigma'(w_{ij}^{(0)}x_j)\sigma'(w_{ij}^{(0)}x_j')\Bigg) \tag{61}$$

$$=: \frac{1}{N}\sum_i \beta_i(x,x'). \tag{62}$$

If you squint a little, you'll see that the i-sum is a sum over the same type of object, $\beta_i(x,x')$, whose i dependence comes from all these i.i.d. parameter draws in the i-direction. By the law of large numbers, we have that in the $N\to\infty$ limit

$$\bar\Theta(x,x') = \langle\beta_i(x,x')\rangle, \tag{63}$$

with no sum on i. We emphasize

> **Observation:** The NTK in the $N\to\infty$ limit is deterministic (parameter-independent), depending only on $P(\theta)$.

Sometimes, the expectation may be computed exactly, and one knows the NTK that governs the dynamics once and for all.

2.3.2 An Exactly Solvable Model

Let us consider a special case of frozen-NTK dynamics with MSE loss,

$$\ell(\phi(x),y) = \frac{1}{2}(\phi(x)-y)^2. \tag{64}$$

Then the dynamics (57) becomes

$$\frac{d\phi(x)}{dt} = -\frac{\eta}{|\mathcal{D}|}\sum_{\alpha=1}^{|\mathcal{D}|}(\phi(x_\alpha)-y_\alpha)\bar\Theta(x,x_\alpha). \tag{65}$$

The solution to this ODE is

$$\phi_t(x) = \phi_0(x)+\frac{1}{|\mathcal{D}|}\bar\Theta(x,x_\alpha)\bar\Theta(x_\alpha,x_\beta)^{-1}\left(\mathbb{1}-e^{-\eta\bar\Theta t}\right)_{\beta\gamma}(y_\gamma-\phi_0(x_\gamma)), \tag{66}$$

where computational difficulty is that $\bar\Theta(x,x_\alpha)$ is a $|\mathcal{D}|\times|\mathcal{D}|$ matrix and takes $O(|\mathcal{D}|^3)$ time to invert. The solution defines a trajectory through function space from ϕ_0 to ϕ_∞. The converged network is

$$\phi_\infty(x) = \phi_0(x)+\bar\Theta(x,x_\alpha)\bar\Theta(x_\alpha,x_\beta)^{-1}(y_\beta-\phi_0(x_\beta)). \tag{67}$$

This is known as **kernel regression**, a classic technique in ML. In general kernel regression, one chooses the kernel. In our case, gradient descent training in the $N\to\infty$ limit *is* kernel regression, with respect to a specific kernel determined by the NN, the NTK $\bar\Theta$.

On train points we have **memorization**

$$\phi_\infty(x_\alpha) = y_\alpha \qquad \forall\alpha. \tag{68}$$

On test points x, the converged network is performing an interpolation, communicating residuals R_β on train points β through a fixed kernel $\bar\Theta$ to test points x. The prediction depends on ϕ_0, but may be averaged over to obtain

$$\mu_\infty(x) := \langle\phi_\infty(x)\rangle = \bar\Theta(x,x_\alpha)\bar\Theta(x_\alpha,x_\beta)^{-1}y_\beta, \tag{69}$$

provided that $\langle\phi_0\rangle = 0$, as in many initializations for the parameters. Let's put some English on the **remarkable facts**,

- $\mu_\infty(x)$ is the mean prediction of an ∞ number of ∞-wide NNs trained to ∞ time.
- If ϕ_0 is drawn from a GP, then ϕ_∞ is as well. The mean is precisely $\mu_\infty(x)$, see [19] for the two-point function and covariance.

2.3.3 Feature Learning

The frozen NTK is a tractable toy model, but it has a major drawback: it does not learn features. In this section we briefly present principles that ensure feature learning, building on lecture notes of Pehlevan and Bordelon [20] that I also utilized in detail in my TASI lectures; see those lectures for more details, [21, 22] for original literature from those lectures that I utilize, and [23, 24] as well as Boris Hanin's lectures at this workshop for related work on feature learning. The **feature learning principles** are

- **Finite Initialization Pre-activations.** $z^{(\ell)} \sim O_N(1)\ \forall l.$
- **Learning in Finite Time.** $d\phi(x)/dt \sim O_N(1)$.
- **Feature Learning in Finite Time.** $dz^{(\ell)}/dt \sim O_N(1)\ \forall l.$

Where the words are the general idea, and the equations their implementation in a specific case: a deep feedforward network with L layers and width N, which in all is a map

$$\phi : \mathbb{R}^D \to \mathbb{R} \tag{70}$$

(note the input dimension D; d is reserved for below) defined recursively as

$$\phi(x) = \frac{1}{\gamma_0 N^d} z^{(L)}(x) \tag{71}$$
$$z^{(L)}(x) = \frac{1}{N^{a_L}} w_i^{(L)}\, \sigma(z_i^{(L-1)}(x)) \tag{72}$$
$$z_i^{(\ell)}(x) = \frac{1}{N^{a_\ell}} W_{ij}^{(\ell)} \sigma(z_j^{(\ell-1)}(x)) \tag{73}$$
$$z_i^{(1)}(x) = \frac{1}{N^{a_1}\sqrt{D}} W_{ij}^{(1)} x_j \tag{74}$$

where Einstein summation is implied throughout this section (unless stated otherwise) and all Latin indices run from $\{1, \dots, N\}$ *except in the j-index in the first layer*, when they are $\{1, \dots, D\}$. The parameters are drawn

$$w_i^{(\ell)} \sim \mathcal{N}\left(0, \frac{1}{N^{b_L}}\right) \qquad W_{ij}^{(\ell)} \sim \mathcal{N}\left(0, \frac{1}{N^{b_\ell}}\right). \tag{75}$$

We scale the learning rate as

$$\eta = \eta_0 \gamma_0^2 N^{2d-c} \tag{76}$$

with γ_0, η_0 $O(1)$ constants, where d has already been introduced but c is a new parameter. The z's are known as the pre-activations, as they are the inputs to the activation functions σ.

We have a standard MLP but have parameterized our ignorance of N-scaling, governed by parameters (a_ℓ, b_ℓ, c, d). Demanding that our principles hold, a few page calculation yields

> **Result:** There is a one-parameter family of solutions that ensure feature learning, according to the principles above.

Furthermore, it is completely fixed if one also demands that η is $O(1)$ in N. This is known as the maximal update parameterization [21].

3 Machine Learning for Field Theory

In this lecture we will turn the table around, asking instead whether ML can do something for field theory.

3.1 NN-FT Correspondence

Understanding the statistics and dynamics of NNs has led us naturally to objects that we are used to from field theory. The idea has been to understand ML theory, but one can also ask the converse, whether ML theory gives new insights into field theory. With that in mind, we ask

> **Question:** What is a field theory?

At the very least, a field theory needs

- **Fields**, functions from an appropriate function space, or sections of an appropriate bundle, more generally.

- **Correlation Functions** of fields, here expressed as scalars

$$G^{(n)}(x_1, \ldots, x_n) = \langle \phi(x_1) \ldots \phi(x_n) \rangle. \tag{77}$$

You might already be wanting to add more beyond these minimal requirements – we'll discuss that in a second. For now, we have

> **Answer:** a FT is an ensemble of functions with a way to compute their correlators.

In the Euclidean case, when the expectation is a statistical expectation, one my say

> **Euclidean Answer:** a FT is a statistical ensemble of functions.

Our minimal requirements get us a partition function

$$Z[J] = \langle e^{\int d^dx J(x)\phi(x)} \rangle \tag{78}$$

that we can use to compute correlators, where at this stage we are agnostic about the definition of $\langle \cdot \rangle$. In normal field theory, the $\langle \cdot \rangle$ is defined by the Feynman path integral

$$Z[J] = \int \mathcal{D}\phi \, e^{-S[\phi] + \int d^dx J(x)\phi(x)}, \tag{79}$$

which requires specifying an action $S[\phi]$ that determines a density on functions $\exp(-S[\phi])$. But that's not the data we specify when we specify a NN. The NN data $(\phi_\theta, P(\theta))$ instead defines

$$Z[J] = \int d\theta P(\theta) e^{\int d^dx J(x)\phi_\theta(x)}. \tag{80}$$

These are two different ways of defining a field theory, and indeed given $(\phi_\theta, P(\theta))$ one can try to work out the associated action, in which case we have dual description of the same field theory, as in the NNGP correspondence. The parameter space description is already quite useful, though, as it enables the computation of correlation functions even if the action isn't known. In certain cases it enables the computation of exact correlators in interacting theories.

Okay, you get it, this is a different way to do field theory. Now I'll let you complain about my definition. You're asking

> **Question:** Shouldn't my definition of field theory include X?

I'm writing this before I give the lecture, and my guess is you already asked about a set of X's, e.g.

$$X \in \{\text{Quantum, Lagrangian, Symmetries, Locality}, \ldots\}. \tag{81}$$

The problem is that with any such X, there's usually some community of physicists that doesn't care. **QFT Types / Communities**:

1. Perturbative Lagrangian,
2. Lattice,
3. Geometric Engineering,
4. Conformal Bootstrap,
5. Statistical Field Theory,
6. Constructive QFT,
7. Algebraic QFT,
8. ...

For instance, not all statistical field theories are Wick rotations of quantum theories; not all field theories have a known Lagrangian; not all field theories have symmetry; not all field theories are local. So I'm going to stick with my definition, because at a minimum I want fields and correlators.

Instead, if your X isn't included in the definition of field theory, then **X is an engineering problem**. Whether you're defining your specific theory by $S[\phi]$, $(\phi_\theta, P(\theta))$, or something else, you can ask

Question: Can I engineer my defining data to get FT + X?

For $X =$ Symmetries you've already seen ways to do this at the level of actions in QFT1 and at the level of $(\phi_\theta, P(\theta))$ in these lectures.

For a current account of recent progress in NN-FT, see the rather long Introduction of [16] and associated references, as well as results.

3.2 When is a NN-FT Quantum?

We've been on Euclidean space the whole time[1], so it's natural to wonder in what sense these field theories are *quantum*. In a course on field theory, we first learn to canonically quantize and then at some later point learn about Wick rotation, and how it can define Euclidean correlators. The theory is manifestly quantum.

But given a Euclidean theory, can it be continued to a well-behaved quantum theory in Lorentzian signature, e.g. with unitary time evolution and a Hilbert space without negative norm states? If we have a nice-enough local action, it's possible, but what if we don't have an action? We ask:

Question: Given Euclidean correlators, can the theory be continued to a well-behaved Lorentzian quantum theory?

This is a central question in axiomatic quantum field theory, and the answer is that it depends on the properties of the correlators. The **Osterwalder-Schrader (OS) Axioms** [26] gives a set of conditions on the Euclidean correlators that ensure that the theory can be continued to a unitary Lorentzian theory that satisfies the Wightman axioms. The conditions of the theorem include

- **Euclidean Invariance**. The correlators are invariant under the Euclidean group, which after continuation to Lorentzian signature becomes the Poincaré group.
- **Permutation Invariance** of the correlators $G^{(n)}(x_1, \ldots, x_n)$ under any permutation of the $x_1, \ldots, x_n$.
- **Reflection Positivity**. Having time in Lorentzian signature requires picking a Euclidean time direction τ. Let $R(x)$ be the reflection of x in the $\tau = 0$ plane. Then reflection positivity requires that
$$G^{(2n)}(x_1, \ldots, x_n, R(x_1), \ldots, R(x_n)) \geq 0. \tag{82}$$
Technically, this is necessary but not sufficient. An accessible elaboration can be found in notes [27] from a previous TASI.
- **Cluster Decomposition** occurs when the connected correlators vanish when any points are infinitely far apart.

If all of these are satisfied, then the pair $(\phi_\theta, P(\theta))$ that defines the NN-FT actually defines a neural network *quantum* field theory [28]. In NN-FT, permutation invariance is essentially automatic and Euclidean invariance may be engineered [28]. Cluster decomposition and reflection positivity hold in some examples [28, 16], but systematizing their construction is an important direction for future work.

There is at least one straightforward way to obtain an interacting NN-QFT. Notably, if $(\phi_\theta, P(\theta))$ is a NNGP that satisfies the OS axioms (this is much easier [28]) with Gaussian partition function

$$Z_G[J] = \int d\theta P(\theta) e^{\int d^dx J(x)\phi_\theta(x)}, \tag{83}$$

then one may insert an operator associated to any local potential $V(\phi)$, which deforms the action in the expected way and the NN-FT to

$$\begin{aligned} Z[J] &= \int d\theta P(\theta) e^{\int d^dx V(\phi_\theta(x))} e^{\int d^dx J(x)\phi_\theta(x)} && (84) \\ &=: \int d\theta \tilde{P}(\theta) e^{\int d^dx J(x)\phi_\theta(x)}, && (85) \end{aligned}$$

where the architecture equation $\phi_\theta(x)$ lets us sub out the abstract expression for a concrete function of parameters, defining a new density on parameters $\tilde{P}(\theta)$ in the process. The interactions in $V(\phi)$ break Gaussianity of the NNGP that was ensured by a CLT. This means

[1] This can be relaxed, see e.g. for a recent paper defining an equivariant network in Lorentzian signature [25].

a CLT assumption must be violated: it is the breaking of statistical independence in $\tilde{P}(\theta)$. The theory $Z[J]$ defined by $(\phi_\theta, \tilde{P}(\theta))$ is an interacting NN-QFT, since local potentials that deformed Gaussian QFTs still satisfy reflection positivity and cluster decomposition.

With this discussion of operator insertions, it's clear now how to get ϕ^4 theory. We just insert the operator

$$e^{\int d^dx \phi_\theta(x)^4} \tag{86}$$

into the partition function associated to the free scalar; this operator with the architecture (100) technically requires an IR cutoff, though other architectures realizing the free scalar may not. The operator insertion deforms the parameter densities in (101) and breaks their statistical independence, explaining the origin of interactions in the NN-QFT. See [16] for a thorough presentation.

3.3 Features of Field Theories in NN-FT

If we are to view field theories as theories of fields and associated correlation functions, which might be adorned with extra features in X, it is natural to ask which features may be engineered in NN-FT. This is a developing subject, but I will summarize some results related to symmetries, interactions, and conformal fields, and then present some essential results on quantum mechanics.

3.3.1 Symmetries

It natural at this point to ask:

Question: Are there global symmetries in NN-FT?

By this I mean symmetries that the ensemble of NNs realizes that an individual network might not see, i.e. the individual network might not be invariant or even equivariant, but the ensemble is invariant.

To allow for symmetries at both input and output, in this section we consider networks

$$\phi : \mathbb{R}^d \to \mathbb{R}^D. \tag{87}$$

with D-dimensional output. We're interested in symmetries that arise in *ensembles* of neural networks, which leave the statistical ensemble invariant. In field theory, we call them **global symmetries**. Let the network transform under a group action as

$$\phi \mapsto \phi_g, \qquad g \in G. \tag{88}$$

We say that ensemble of networks has a global symmetry group G if the partition function is invariant,

$$Z_g[J] = Z[J], \qquad \forall g \in G. \tag{89}$$

At the level of expectations, this is

$$\langle e^{\int d^dx J(x)\phi_g(x)} \rangle = \langle e^{\int d^dx J(x)\phi(x)} \rangle \qquad \forall g \in G, \tag{90}$$

where one can put indices on ϕ and J as required by D. By a field redefinition, this may be cast as having a symmetry if $\langle \cdot \rangle$ is invariant. In the usual path integral this is the statement of invariant action $S[\phi]$ and measure $\mathcal{D}\phi$. In parameter space, the redefinition may be instituted [29] by absorbing g into a redefinition of parameters as $\theta \mapsto \theta_g$, with symmetry arising when

$$\int d\theta_g P(\theta_g) e^{\int d^dx J(x)\phi_\theta(x)} = \int d\theta P(\theta) e^{\int d^dx J(x)\phi_\theta(x)}, \tag{91}$$

i.e. the parameter density and measure must be invariant. We will give a simple example realizing this mechanism in a moment.

It is most natural to transform the input or output of the network. Our mechanism allows for symmetries of both types, which in normal language are

NN Ensembles	**Field Theory**
Input Symmetries	Spacetime Symmtries
Output Symmetries	Internal Symmetries

It may also be interesting to study symmetries of intermediate layers, if one wishes to impose symmetries on learned representations. Equivariance fits into this picture because it turns a transformation at input into a transformation at output. The ensemble of equivariant NNs is invariant under ρ_d action on the input if the partition function is invariant under the induced ρ_D action on the output.

Example. Theories that are Wick rotations of Lorentz-invariant QFTs enjoy Euclidean symmetry (rotations and translations). To see how it might arise, consider the neurons

$$\ell_i(x) = F(\mathbf{w^{(0)}}_i) \cos\left(\sum_j w^{(0)}_{ij} x_j + b^{(0)}_i\right), \quad i \in 1, \ldots, N, \tag{92}$$

where the sum has been made explicit since i's are not summed over, with

$$w^{(0)}_{ij} \sim P(w^{(0)}_{ij}) \qquad b^{(0)}_i \sim \text{Unif}[-\pi, \pi]. \tag{93}$$

These neurons are in Euclidean invariant ensembles, and enjoy other properties

- **Larger Euclidean Nets.** Any network that builds on ℓ, e.g. $f(\ell(x))$, without reusing its parameters is Euclidean-invariant.
- **Spectrum Shaping.** In computing $G^{(2)}(p)$, $\mathbf{b}_i$ gets evaluated on p, and F may be chosen to shape the power spectrum (momentum space propagator) arbitrarily.
- **Periodic Functions.** One may replace cos by any periodic function with period 2π, (or otherwise, if one also changes the endpoints of the Unif distribution).

We refer the reader to [28] for general calculations of ℓ-correlators.

We obtain the free scalar by spectrum shaping with the choice

$$F(\mathbf{w^{(0)}}_i) = \frac{1}{\mathbf{w^{(0)}}_i \cdot \mathbf{w^{(0)}}_i + m^2}. \tag{94}$$

Then a linear output layer on the ℓ's gives a network

$$\phi(x) = \frac{1}{\sqrt{N}} \sum a_i \ell_i(x) \tag{95}$$

chosen with output weights i.i.d. and $\langle a^2 \rangle \sim N^0$, $\langle a \rangle = 0$. A short computation gives

$$G^{(2)}(p) = \frac{1}{p^2 + m^2} \tag{96}$$

up to a computable normalization factor for all N, which is the only non-trivial correlator as $N \to \infty$, realizing the free scalar field.

3.3.2 Interactions and ϕ^4 Theory

In my first lecture, we learned that non-interacting (i.e. Gaussian) NN-FTs are in general Gaussian for a reason: the Central Limit Theorem (CLT); see Appendix A for a brief review. Of course, the vanilla statement is that a random variable

$$\phi = \frac{1}{\sqrt{N}} \sum_{i=1}^{N} X_i, \tag{97}$$

obtained as a sum of N i.i.d. random variables X_i with finite variance, is Gaussian in the limit $N \to \infty$. This makes two essential **assumptions:**

1. $N \to \infty$.
2. $X_i \sim P(X_i)$ are i.i.d.

Therefore, one may introduce interactions (non-Gaussianities) by violating these assumptions. That is:

Origin of Interactions: $1/N$ or independence breaking.

The $1/N$ non-Gaussianities are captured by the connected correlation functions, where for a NN obtainted from N neurals we have

$$\phi(x) = \frac{1}{N} \sum_i a_i \varphi_i(x), \tag{98}$$

where a_i and φ_i are output weights and neurons drawn i.i.d., and the connected correlators scale as

$$G^{(2n)}_c(x_1, \ldots, x_n) \sim \frac{1}{N^{n-1}}, \tag{99}$$

so that higher order correlators vanish as $N \to \infty$, precisely as expected by the CLT. We saw this same result in the first lecture when we discussed non-Gaussian processes.

One may also exhibit interactions by independence breaking, even in the $N \to \infty$ limit, which we now exemplify.

Example: ϕ^4 Theory. Here is one final example you might like,

$$\phi(x) = \sqrt{\frac{2\text{vol}(B^d_\Lambda)}{(2\pi)^d \sigma^2_{w_1}}} \frac{1}{\sqrt{\mathbf{w^{(0)}}_i^2 + m^2}} \, w^{(1)}_i \cos\left(w^{(0)}_{ij} x_j + b^{(0)}_i\right), \tag{100}$$

and specific parameter densities

$$w^{(1)} \sim \mathcal{N}\left(0, \frac{\sigma^2_{w_1}}{N}\right) \qquad w^{(0)} \sim \text{Unif}(B^d_\Lambda) \qquad b^{(0)} \sim \text{Unif}[-\pi, \pi], \tag{101}$$

where B^d_Λ is a d-ball of radius Λ. The theory is translation invariant by construction, and so we compute the power spectrum of the two-point function $G^{(2)}(x-y)$ to be

$$G^{(2)}(p) = \frac{1}{p^2 + m^2}. \tag{102}$$

We see that we have a realization of the free scalar field theory in d Euclidean dimensions.

Given the free scalar, we know how to obtain ϕ^4 theory. We just insert the operator

$$e^{\int d^dx \phi_\theta(x)^4} \tag{103}$$

into the partition function associated to the free scalar; this operator with the architecture (100) technically requires an IR cutoff, though other architectures realizing the free scalar may not. Writing the parameters $\theta = \{w^{(1)}, w^{(0)}, b^{(0)}\}$ and the free scalar partition function as

$$Z[J] = \int d\theta P(\theta) \, e^{\int d^dx J(x) \phi_\theta(x)} \tag{104}$$

we see the operator insertion deforms the parameter densities

$$P(\theta) \mapsto P(\theta)_{\phi^4} = P(\theta) \, e^{\int d^dx \phi_\theta(x)^4} \tag{105}$$

where the RHS depends on parameters through the use of the architecture equation, explaining the origin of interactions by the breaking of statistical independence. See [16] for a thorough presentation.

Lorentz Generator	**Conformal Transformation**
L_{ij}	Rotation
L_{+-}	Scaling
L_{i+}	Translation
L_{i-}	Special Conformal

Table 1: Lorentz generators in $(D+2)$-dimensions and the conformal transformation they induce on the Poincaré section $\mathbb{R}^D$, written in the light-cone coordinates instead of Minkowski, where $i = 1, \cdots, D$ and $+/-$ are the light-cone indices.

3.3.3 Conformal Fields

Another central topic in FT is

> **CFT:** describes RG fixed points and phase transitions.

Minimally, they are field theories with correlators that respect the conformal group, the group of transformations that preserve angles, which in d Euclidean dimensions dimensions is $SO(d+1,1)$. This leads to

> **Dirac's Observation:** $SO(d+1,1)$ is the Lorentz group in $d+2$ dimensions and the conformal group in d dimensions.

This observation is utilized in the so-called *embedding formalism* in the CFT community, which obtains the space $\mathbb{R}^d$ of a CFT from $\mathbb{R}^{d+1,1}$ by passing to the null cone

$$\text{NC} := \{x \cdot x + X^2_{d+1} - X^2_0 = x \cdot x - X_+ X_- = 0\}, \tag{106}$$

where $X_\mu = (X_0, x, X_{d+1}) \in \mathbb{R}^{d+1,1}$, the lightcone coordinates are $X_\pm = X_0 \pm X_{d+1}$, and $x \in \mathbb{R}^d$. The projective null cone is defined to be

$$\mathbb{P}\mathrm{NC} = \frac{\mathrm{NC} \setminus 0}{\mathbb{R} \setminus 0}, \tag{107}$$

and one obtains a copy of $\mathbb{R}^d$ by passing to the Poincaré section

$$X_\mu = (X_+, x, X_-) = (1, x, x^2) \in \mathbb{R}^d \subsetneq \mathbb{R}^{d+1,1}. \tag{108}$$

Then, Lorentz transformations in $\mathbb{R}^{d+1,1}$ induce conformal transformations in $\mathbb{R}^d$ according to Table 1.

Neural networks may be used to define conformal fields [30], where a conformal field on the Poincaré section is obtained from a Lorentzian theory associated to a homogeneous neural network architecture. There are **three properties** to ensure:

1. **Homogeneity** arising from the choice of architecture.

2. **Lorentz-invariance** in $D+2$ dimensions from an appropriately chosen $P(\Theta)$.

3. **Finiteness.** The correlators must be well-defined.

Homogeneity and formal Lorentz-invariance is often straightforward, requiring only a careful choice of architecture and $P(\Theta)$, but ensuring that the correlators are well-defined is a non-trivial task. This construction ensures that the conformal fields have conformally invawriant correlators, a minimal definition for a CFT. Crossing symmetry is automatic in the four-point function, and induces non-trivial constraints on the conformal block decomposition when the theory has an operator product expansion. Interestingly, the Lorentzian theory *need not be translation invariant*, as translation-invariance of the conformal field emerges from the Lorentz symmetry itself.

Pedagogical Example. Here's a simple non-unitary example with $\Delta = -1$ that proves the point. Let the architecture be

$$\Phi(X) = \Theta \cdot X \tag{109}$$

where $\Theta \sim P(\Theta)$ is rotationally invariant in Euclidean (D+2)-dimensions, giving a Lorentz-invariant theory upon Wick rotation to $\mathbb{R}^{D+1,1}$. The two-point and four-point functions of Φ are given by

$$G^{(2)}(x_1, x_2) = x_{12}^2 \tag{110}$$

$$G^{(4)}(x_1, x_2, x_3, x_4) = \frac{\mu_4}{3}\left[x_{12}^2 x_{34}^2 + x_{13}^2 x_{24}^2 + x_{14}^2 x_{23}^2\right], \tag{111}$$

where $x_{ij}^2 := (x_i - x_j)^2 = -2X_i \cdot X_j$ in the embedding space and $\mu_4 := \mu_{4,i,i,i,i}$, the diagonal part of the moment tensor of $P(\Theta)$. In the s-channel the four-point function is

$$G^{(4)}(x_1, x_2, x_3, x_4) = g(u, v)\, x_{12}^2 x_{34}^2, \tag{112}$$

where

$$g(u, v) = \frac{\mu_4}{3}\left(1 + \frac{1}{u} + \frac{v}{u}\right), \qquad u = \frac{x_{12}^2 x_{34}^2}{x_{13}^2 x_{24}^2}, \qquad v = \frac{x_{14}^2 x_{23}^2}{x_{13}^2 x_{24}^2}. \tag{113}$$

We emphasize that this is an *exact* expression for the conformal four-point function of Φ. Using coordinates where $u = z\bar{z}, v = (1 - z)(1 - \bar{z})$, as conventional in the bootstrap community, it may be decomposed into $4D$ conformal blocks (CB) $g_{\Delta,\ell}$ as

$$g(z, \bar{z}) = \frac{\mu_4}{3}\left(2g_{-2,0}(z, \bar{z}) + \frac{4}{3}g_{0,0}(z, \bar{z})\right), \tag{114}$$

which tells us that the $\Phi \times \Phi$ OPE has a $(-2, 0)$ operator and a $(0, 0)$ operator. An essential element of bootstrap-ology is that the OPE relates coefficients of CBs to coefficients of three-point and two-point functions. For instance, postulating that the $(-2, 0)$ operator is precisely Φ^2, we compute

$$\begin{aligned} G_{\Phi\Phi\Phi^2}(X, Y, Z) &= \mathbb{E}[\Theta_i\Theta_j\Theta_{k_1}\Theta_{k_2}]X^iY^jZ^{k_1}Z^{k_2} = \frac{2\mu_4}{3}(X \cdot Z)(Y \cdot Z), \\ G_{\Phi^2\Phi^2}(X, Y) &= \mathbb{E}[\Theta_{i_1}\Theta_{j_1}\Theta_{i_2}\Theta_{j_2}]X^{i_1}X^{j_1}Y^{i_2}Y^{j_2} = \frac{2\mu_4}{3}(X \cdot Y)^2 \end{aligned} \tag{115}$$

and the CBD coefficient is the three-point coefficient squared over the two-point coefficient. Similarly, postulating that the $(0,0)$ operator is $\Theta \cdot \Theta =: \mathcal{O}_{1,0}$, we have

$$G_{\Phi\Phi\mathcal{O}_{1,0}}(X,Y,Z) = \delta^{i_1 j_1}\mathbb{E}[\Theta_{i_1}\Theta_{j_1}\Theta_k\Theta_j]X^k Y^j = \frac{\mu_4}{3}((D+2)+2)X\cdot Y,$$
$$G_{\mathcal{O}_{1,0}\mathcal{O}_{1,0}}(X,Y) = \delta^{i_1 j_1}\delta^{i_2 j_2}\mathbb{E}[\Theta_{i_1}\Theta_{j_1}\Theta_{i_2}\Theta_{j_2}] = \frac{\mu_4}{3}((D+2)^2+2(D+2)), \tag{116}$$

and the CBD coefficient is recovered precisely in the case $D = 4$, which agrees with the fact that we did a $4D$ conformal block decomposition. The prescription also hold for $G^{(4)}_{\Phi^2}$, which involves the computation of a (simple) four-loop effect, yielding a quartic polynomial in D that must be restricted to $D = 4$ to obtain the match.

3.3.4 NN-QM

In Section 3.2 we gave one definition of when a NN-FT is a QFT: when it satisfies the Osterwalder-Schrader axioms, which guarantee a Lorentzian continuation satisfying the Wightman axioms. Since one of the Wightman axioms is Lorentz invariance, this is clearly too strong for the general case, we would like to back up and ask for more general notions of when a NN-FT is quantum.

The simplest place to carry out this exercise is in $d = 1$. The following discussion is based on work to appear soon with C. Ferko. We change notation such that

Rename: $\phi_\theta(t) \mapsto x_\theta(t)$, Feynman's paths in QM.

We wish to understand circumstances under with NNs yield $x(t)$'s exhibit features of (Euclidean) quantum mechanics.

Let us begin with a general stochastic process (SP) $x(t)$ and ask

Question: Are there **minimal requirements** that a SP $x(t)$ must sastify to be Euclidean QM?

Of course, there are many essential notions in QM. Some **minimal requirements**[2] (that are not sufficient) are:

- **Path Continuity:** $x(t)$ is continuous, no jumps!
- **Finite Two-point Function:** $G^{(2)}(t,s)$ is finite for all t,s. This follows from the Källén-Lehmann spectral representation.

These are sufficient to ensure the assumptions of the Kosambi-Karhunen-Loève theorem (KKL), which ensures that x_t admits a decomposition

$$x_t = \sum_{k=1}^{\infty} \theta^k e_k(t)\,, \tag{117}$$

where e_k is a set of continuous, orthogonal real-valued functions on $[a,b]$ and θ^k are pairwise uncorrelated. This is a neural networks with continuous, orthogonal neurons e_k and weights θ^k.

Universality: A Euclidean QM theory can be written as a NN.

More generally, since the minimal requirements are not sufficient for QM, there are SPs that satisfy the conditions — and therefore admit a NN description — that are not quantum mechanical. NNs we consider in practice are much different from the KKL decomposition, but an appropriate single-layer network is always sufficient.

Of course, the minimal requirements are not sufficient, and one might ask

Question: What are the sufficient conditions for QM?

There are weaker conditions, but certainly one version is **d=1 OS Axioms**, which include

[2] Technically, the first requirement is mean-square continuity, which is stronger and ensures continuity of paths and the two-point function.

1. **Reflection Positivity.** For all $t_i > 0$ and bounded F

$$\langle F(x(t_1), \ldots, x(t_n)) \, (F(x(-t_1), \ldots, x(-t_n)))^* \rangle \geq 0 \qquad (118)$$

2. **Tranlation Invariance.** $x_t = x_{t+a}$ in distribution.

3. **Symmetric.** $x_t = x_{-t}$ in distribution,

as well as some technical or trivial conditions suppressed for brevity. We call such theories OS-QM.

The relationship between the different notions of SP is presented in Figure 1. Within the space of all stochastic processes (SP), a subset (NN-SP) admit a representation as a neural network. SPs satisfying the minimal requirements admited a NN description and are denoted NN-SP; those that are actually quantum mechanical are denoted MQM. and therefore MQM $\subset$ NN-SP. One might impose additional restrictions upon minimal quantum models, such as the Osterwalder-Schrader axioms (OS-QM) or another set of conditions defining a notion of quantum mechanics of one's choosing (QM'), which carve out different subsets of MQM.

After establishing universality of the NN description, the focus of our paper is on RP and OS-QM. We have two primary **RP Mechanisms:**

1. **"Parameter Splitting".** The NN architecture has some parameters that control fluctuations on different sides of $t = 0$. Together with a condition of the parameter density, this ensures RP by a perfect square integrand. This is exemplified.

2. **Markov property.** It is well known that Markov → RP. A Markov $x(t)$ may be obtained by traditional means or by an injective NN with Markovian weights $\theta(t)$.

Other important results include:

1. **Deep NN-QM:** A NN acting on any Markov process is RP, and therefore sastifies a crucial condition for QM.

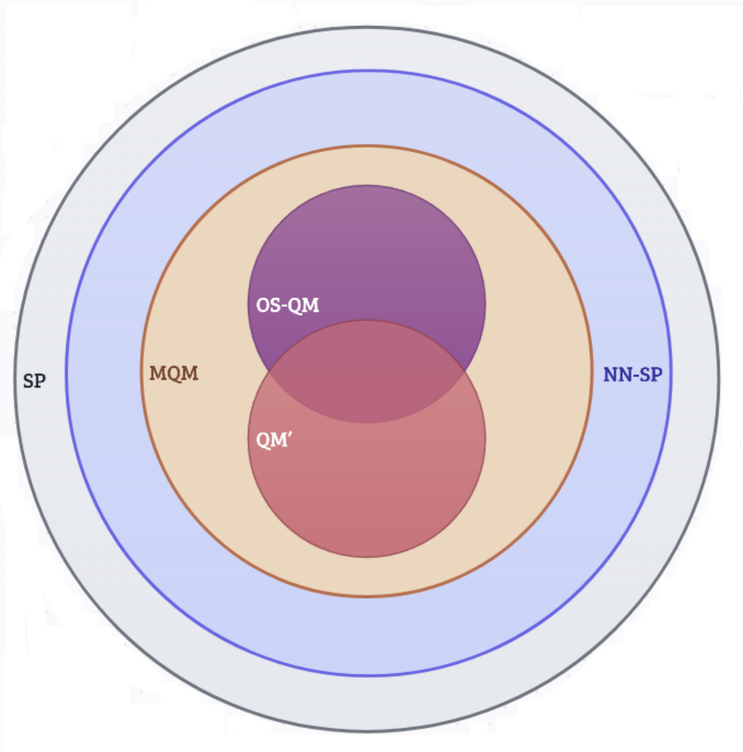

Figure 1: A Venn diagram illustrating the various relations between different SPs and QM.

2. **Numerical Examples** in several cases require classic QM properties, e.g. the spectrum, non-trivial commutators, and Heisenberg uncertainty.

We hope to have this work out within a week of this meeting.

4 Recap and Outlook

In these lectures I discussed one angle on the physics of learning, the relationship between field theory and neural networks. I will recap them briefly, only an hour before the second lecture.

In the first lecture on Physics-for-ML, I argued that very elementary considerations in machine learning with neural networks inevitably leads to field theoretic concepts. Specifically, since neural network predictions depend on a random draw of parameters at initialization, one should integrate over all initializations to compute the average prediction (one-point function) and covariance (connected two-point function). Since learning depends on t, one should track how these correlators evolve. I then turned to details of the statistics, beginning with the NNGP correspondence, which shows that many neural network architectures admit a large-N limit in which the networks are draws from a Gaussian process. This means they are described by a generalized free field theory, with statistics completely determined by the one-point and two-point functions. Non-Gaussianities, which correspond to interactions, arise by violating assumptions of the CLT. I then turned to the dynamics of neural networks, which are governed by the Neural Tangent Kernel (NTK). With certain scaling in the large-N limit, the NTK becomes frozen and the frozen NTK at initialization governs the dynamics for all time. However, nothing is being learned, and in particular late-time features in the hidden dimensions are in a local neighborhood of their initial values. I showed how a detailed N-scaling analysis allows one to demand that network features and predictions update non-trivially, leading to richer learning regimes known as dynamical mean field theory or the maximal update parameterization.

In the second lecture, I discussed the NN-FT correspondence, which provides a neural network approach to field theory. In this framework, a field theory is defined by a neural network architecture ϕ_θ and a probability density $P(\theta)$ on its parameters. This allows one to compute correlation functions directly in parameter space, without requiring an explicit action. I explained how Gaussian processes arise naturally in the large-N, corresponding to generalized free field theories. Interactions can be introduced by breaking statistical independence or considering finite-width corrections, leading to non-Gaussian processes. I also discussed how cherished features of field theories, such as symmetries, interactions, and conformal invariance, can be engineered in NN-FT. For example, I showed how Euclidean symmetry can be built into the architecture and parameter distribution, and how ϕ^4 theory can be realized by deforming the parameter density of a free scalar NN-FT. Additionally, I introduced the construction of conformal fields using neural networks, leveraging the embedding formalism to ensure conformal invariance of the correlators. Finally, I explored the connection between neural networks and quantum mechanics (NN-QM). I demonstrated that any Euclidean-time quantum mechanical theory can be represented as a neural network, satisfying minimal requirements such as path continuity and finite two-point functions. Reflection positivity, a key feature of quantum mechanics, can be ensured through mechanisms like parameter splitting or Markov processes. Classic QM results may be recovered in numerical simulations.

A Central Limit Theorem

Let us recall a simple derivation of the Central Limit Theorem (CLT), in order to better understand the statistics of neural networks. Consider a sum of random variables

$$\phi = \frac{1}{\sqrt{N}} \sum_{i=1}^{N} X_i, \tag{119}$$

with $\langle X_i \rangle = 0$. The moments μ_r and cumulants κ_r are determined by the moment generating function (partition function) $Z[J] = \langle e^{J\phi} \rangle$ and cumulant generating function $W[J] = \log Z[J]$, respectively, as

$$\mu_r = \left(\frac{d}{dJ}\right)^r Z[J]\bigg|_{J=0} \tag{120}$$

$$\kappa_r = \left(\frac{d}{dJ}\right)^r W[J]\bigg|_{J=0}. \tag{121}$$

If the X_i are independent random variables, then the partition function factorizes $Z_{\sum_i X_i}[J] = \prod_i Z_{X_i}[J]$, and the cumulant generating function of the sum is the sum of the cumulant generating functions, yielding

$$W_{\sum_i X_i}[J] = \sum_i W_{X_i}[J] \tag{122}$$

$$\kappa_r^{\sum X_i} = \sum_i \kappa_r^{X_i}. \tag{123}$$

If the X_i are identically distributed, then the cumulants $\kappa_r^{X_i}$ are the same for all i and we account for the $1/\sqrt{N}$ appropriately, we obtain

$$\kappa_r^{\phi} = \frac{\kappa_r^{X_i}}{N^{r/2-1}}. \tag{124}$$

This yields

$$\lim_{N\to\infty} \kappa_{r>2}^{\phi} = 0, \tag{125}$$

which is sufficient to show that ϕ is Gaussian in the large-N limit. In physics language, cumulants are connected correlators, and (125) means that Gaussian (free) theories have no connected correlators.

In neural networks we will be interested in studying certain Gaussian limits. From this CLT derivation, we see two potential origins of non-Gaussianity:

- **$1/N$-corrections** from appearance in κ_r^{ϕ}.
- **Independence breaking** since the proof relied on (122).

References

[1] D. Silver, J. Schrittwieser, *et al.*, "Mastering the game of go without human knowledge," *Nature* **550** no. 7676, (Oct, 2017) 354–359. `https://doi.org/10.1038/nature24270`.

[2] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *International conference on machine learning*, pp. 2256–2265, PMLR. 2015.

[3] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," *arXiv preprint arXiv:2011.13456* (2020) .

[4] T. Brown, B. Mann, *et al.*, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, eds., vol. 33, pp. 1877–1901. Curran Associates, Inc., 2020. `https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf`.

[5] S. Bubeck, V. Chandrasekaran, *et al.*, "Sparks of artificial general intelligence: Early experiments with gpt-4," *arXiv preprint arXiv:2303.12712* (2023) .

[6] J. Halverson, "Tasi lectures on physics for machine learning." 2024. `https://arxiv.org/abs/2408.00082`.

[7] H. Robbins and S. Monro, "A Stochastic Approximation Method," *The Annals of Mathematical Statistics* **22** no. 3, (1951) 400 – 407. `https://doi.org/10.1214/aoms/1177729586`.

[8] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain." *Psychological review* **65 6** (1958) 386–408. `https://api.semanticscholar.org/CorpusID:12781225`.

[9] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization." 2017. `https://arxiv.org/abs/1412.6980`.

[10] G. B. De Luca and E. Silverstein, "Born-infeld (bi) for ai: Energy-conserving descent (ecd) for optimization," in *International Conference on Machine Learning*, pp. 4918–4936, PMLR. 2022.

[11] G. B. De Luca, A. Gatti, and E. Silverstein, “Improving energy conserving descent for machine learning: Theory and practice,” *arXiv preprint arXiv:2306.00352* (2023) .

[12] R. M. Neal, “Bayesian learning for neural networks,” *Lecture Notes in Statistics* **118** (1996) .

[13] C. K. Williams, “Computing with infinite networks,” *Advances in neural information processing systems* (1997) .

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, eds., vol. 25. Curran Associates, Inc., 2012. `https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf`.

[15] G. Yang, “Wide feedforward or recurrent neural networks of any architecture are gaussian processes,” *Advances in Neural Information Processing Systems* **32** (2019) .

[16] M. Demirtas, J. Halverson, A. Maiti, M. D. Schwartz, and K. Stoner, “Neural network field theories: non-Gaussianity, actions, and locality,” *Mach. Learn. Sci. Tech.* **5** no. 1, (2024) 015002, `arXiv:2307.03223 [hep-th]`.

[17] S. Yaida, “Non-gaussian processes and neural networks at finite widths,” in *Mathematical and Scientific Machine Learning*, pp. 165–192, PMLR. 2020.

[18] A. Jacot, F. Gabriel, and C. Hongler, “Neural tangent kernel: Convergence and generalization in neural networks,” in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, eds., vol. 31. Curran Associates, Inc., 2018. `https://proceedings.neurips.cc/paper_files/paper/2018/file/5a4be1fa34e62bb8a6ec6b91d2462f5a-Paper.pdf`.

[19] J. Lee, L. Xiao, S. Schoenholz, Y. Bahri, R. Novak, J. Sohl-Dickstein, and J. Pennington, “Wide neural networks of any depth evolve as linear models under gradient descent,” *Advances in neural information processing systems* **32** (2019) .

[20] C. Pehlevan and B. Bordelon, “Lecture notes on infinite-width limits of neural networks.” August, 2024. `https://mlschool.princeton.edu/events/2023/pehlevan`. Princeton Machine Learning Theory Summer School, August 6 - 15, 2024.

[21] G. Yang and E. J. Hu, “Feature learning in infinite-width neural networks,” *arXiv preprint arXiv:2011.14522* (2020) .

[22] B. Bordelon and C. Pehlevan, “Self-consistent dynamical field theory of kernel evolution in wide neural networks,” *Advances in Neural Information Processing Systems* **35** (2022) 32240–32256.

[23] D. A. Roberts, S. Yaida, and B. Hanin, *The principles of deep learning theory*, vol. 46. Cambridge University Press Cambridge, MA, USA, 2022.

[24] S. Yaida, “Meta-principled family of hyperparameter scaling strategies,” *arXiv preprint arXiv:2210.04909* (2022) .

[25] M. Zhdanov, D. Ruhe, M. Weiler, A. Lucic, J. Brandstetter, and P. Forré, “Clifford-steerable convolutional neural networks,” *arXiv preprint arXiv:2402.14730* (2024) .

[26] K. Osterwalder and R. Schrader, “AXIOMS FOR EUCLIDEAN GREEN’S FUNCTIONS,” *Commun. Math. Phys.* **31** (1973) 83–112.

[27] D. Simmons-Duffin, “The Conformal Bootstrap,” in *Theoretical Advanced Study Institute in Elementary Particle Physics: New Frontiers in Fields and Strings*, pp. 1–74. 2017. `arXiv:1602.07982 [hep-th]`.

[28] J. Halverson, "Building Quantum Field Theories Out of Neurons," arXiv:2112.04527 [hep-th].

[29] A. Maiti, K. Stoner, and J. Halverson, "Symmetry-via-Duality: Invariant Neural Network Densities from Parameter-Space Correlators," arXiv:2106.00694 [cs.LG].

[30] J. Halverson, J. Naskar, and J. Tian, "Conformal Fields from Neural Networks," arXiv:2409.12222 [hep-th].