Why Machine Learning in Physics?

Lecture 02 — PHYG004, 2026 Spring

Prof. Young Woo Choi

Department of Physics, Sogang University

3/6/2026

The AI Moment We're In

The Hype

- "AGI is near" headlines
- AI investment bubble ($300B+ in 2024–25)
- Generated content flooding the internet
- Exaggerated benchmark competitions

The Real Progress

- Code generation that actually works (Copilot, Claude Code)
- Agentic systems: autonomous experiment design
- Scientific breakthroughs (next slide)

> We need to separate **signal from noise** — especially as physicists.

AI for Science: Breakthroughs That Changed the Game

AlphaFold (2020, 2024)

Protein structure prediction — solved a 50-year grand challenge

200M+ structures predicted

Structural Biology

GNoME (2023)

Graph NN trained on 89M DFT calculations

2.2M new stable crystals

Materials Science

FermiNet / Psiformer (2020–24)

NN wavefunctions surpassing gold-standard quantum chemistry

Beyond CCSD(T) accuracy

Quantum Chemistry

> These are not incremental improvements — they are **paradigm shifts**.

Three Paradigms of AI for Science

🔍

1. Pattern Recognition

Data → Pattern → Classification

🔄

2. Surrogate Model

Expensive calc → Fast NN proxy

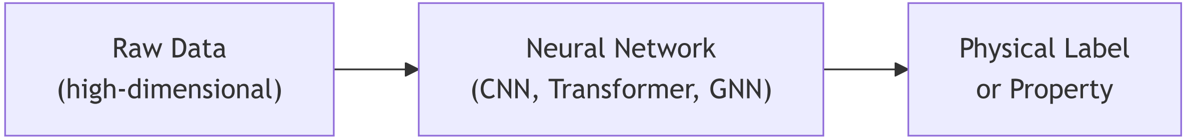
🤖

3. Agent

Observe → Decide → Act

> This course focuses on **1 and 2**. Paradigm 3 is emerging rapidly.

Paradigm 1: Pattern Recognition



Input		**Output**
Spin configurations	→	Phase: ordered / disordered
Detector signals	→	Event: signal / background
Spectra	→	Material class

> **Key idea**: The NN extracts patterns that may be invisible to human analysis or lack a known order parameter.

Pattern Recognition in Action

Example 1: Phase Transition Classification

- Carrasquilla & Melko, *Nature Physics* (2017)
- Input: 2D Ising MC spin configurations (32×32)
- Model: Simple CNN
- Output: Ordered ($T < T_c$) vs Disordered ($T > T_c$)

```
T < Tc: ■■■■■■■■
         ■■■■■■■■
         ■■■■□■■■
T > Tc: ■□■□□■□■
         □■□■■□■□
         ■■□□■□□■
```

NN finds T_c accurately — even for topological phases!

Example 2: Gravitational Wave Detection

- George & Huerta, *Phys. Lett. B* (2018)
- Input: LIGO time-series strain data
- Model: Deep CNN
- Output: Signal present / absent

Matched filtering sensitivity, but **real-time** processing

Both examples use CNN — we build the first one in **Week 4**.

Paradigm 2: Surrogate Model

Traditional

```
      Atomic structure
              │
              ▼
        ┌──────────────┐
│  DFT / QMC   │ ← hours ~ days
        │  (ab initio) │
        └──────┬───────┘
               ▼
      Energy, Forces
```

ML Surrogate

```
      Atomic structure
              │
              ▼
        ┌──────────────┐
│  Neural Net  │ ← milliseconds
        │  (trained)    │
        └──────┬───────┘
               ▼
      Energy, Forces
```

Speed: 10^3–$10^6\times$ faster | **Accuracy**: DFT-level or beyond | **Scale**: millions of atoms

Surrogate Models in Action

1. Machine Learning Interatomic Potentials (MLIP)

Wk 11–12

- NequIP, MACE: E(3)-equivariant GNN maps atomic structure → energy/forces
- DFT accuracy + MD speed → million-atom simulations

2. Neural Network Wavefunctions

Wk 5, 9

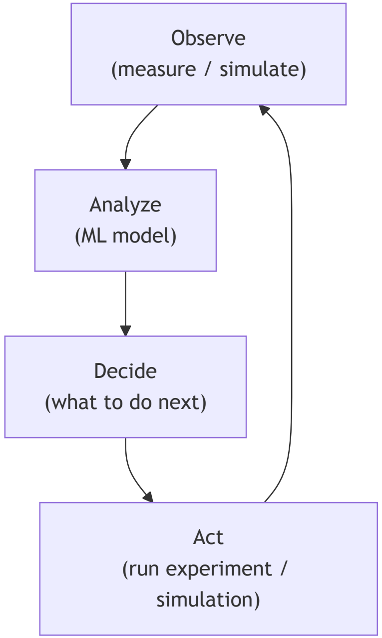
- FermiNet, Psiformer: NN represents $\Psi(\mathbf{r}_1, \ldots, \mathbf{r}_N)$
- Variational Monte Carlo training → beyond CCSD(T)

3. Learning the Hamiltonian Directly

Wk 11–12

- DeepH: atomic structure → DFT Hamiltonian matrix
- Electronic structure of 10,000+ atoms without SCF iteration

Paradigm 3: Agent



- ML decides **what experiment/calculation to run next**
- Active learning: explore regions of highest uncertainty
- Bayesian optimization: minimize objective with fewest evaluations

Agent Paradigm: Examples

1. Active Learning for Training Data

- Problem: MLIP training needs DFT data, but DFT is expensive
- Solution: auto-select structures where NN uncertainty is highest → run DFT → add to training set
- Result: **10× fewer DFT calculations** needed

2. Self-Driving Laboratories

- Acceleration Consortium (U of T), A-Lab (LBNL)
- ML proposes synthesis conditions → robot synthesizes → characterize → ML updates
- 24/7 autonomous operation for accelerated materials discovery

> In this course, we focus on Paradigms 1 & 2. But many term projects can incorporate active learning ideas.

Course Roadmap

Week	Topic	Paradigm Connection
1–4	NN Basics, CNN	Foundation
5	FNN/CNN in Physics (papers)	Pattern Recognition, Surrogate
6	Automatic Differentiation	Foundation (enables all)
7	Individual Presentations	—
8	(Midterm week, no exam)	—
9–10	Transformers	Pattern Recognition, Surrogate
11–12	Equivariant NNs	Surrogate (MLIP, NQS)
13–14	Generative Models (VAE/Flow/Diff)	Sampling

Part 2

Physics Modeling vs ML Modeling

How Physicists Build Models

Fundamental principles (Newton, Schrödinger, Maxwell)

↓ symmetry → conservation laws

Mathematical structure (Hamiltonian, Lagrangian)

↓ approximation → mean-field, perturbation

Tractable equations (Hartree-Fock, Landau theory)

↓ dimensional reduction → order parameter

Predictions (phase diagram, spectrum)

Symmetry

Approximation

Dim. Reduction

The Physicist's Toolkit

Symmetry → Reduce the space

- Rotational symmetry → angular momentum quantum numbers ℓ, m → block-diagonalize Hilbert space
- Translation symmetry → Bloch's theorem, reduction to k-space

Approximation → Make it solvable

- Mean-field: many-body → effective single-body (e.g., Hartree-Fock)
- Perturbation theory: expand in small parameter (e.g., $e^2/\hbar c \approx 1/137$)
- Variational: trial wavefunction + energy minimization (e.g., VMC)

Dimensional Reduction → Capture the essence

- Order parameter: 10^{23} spins → magnetization m
- Effective theory: integrate out high-energy DOF → low-energy EFT
- Renormalization group: remove irrelevant DOF scale by scale

Where Physics Modeling Breaks Down

$$\Psi(\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_N) \in \mathbb{C}^{3N\text{-dimensional space}}$$

N (electrons)	Dimension ($3N$)	Grid points (10/axis)	Feasibility
2	6	10^6	✅ tractable
10	30	10^{30}	❌
50	150	10^{150}	❌ universe has $\sim 10^{80}$ atoms
100	300	10^{300}	❌❌❌

> **This is why we need ML.** Traditional basis expansions fail exponentially. Neural networks offer a way out.

How ML Builds Models: Function Approximation

$$\text{Given data } \{(x_i, y_i)\}_{i=1}^{N}, \quad \text{find } f_\theta \text{ such that } f_\theta(x_i) \approx y_i$$

- f_θ: function parameterized by θ (neural network)
- Objective: minimize loss $\mathcal{L}(\theta) = \frac{1}{N}\sum_i |f_\theta(x_i) - y_i|^2$
- Update rule: $\theta_{t+1} = \theta_t - \eta \nabla_\theta \mathcal{L}$

Universal Approximation Theorem (Cybenko 1989, Hornik 1991): A single hidden layer NN with sufficient width can approximate any continuous function on a compact domain to arbitrary accuracy.

But this theorem says nothing about **how many parameters** or **how much data** you need.

The Curse of Dimensionality

d-dimensional unit cube $[0, 1]^d$, grid spacing ϵ:

$$N_{\text{grid}} = \left(\frac{1}{\epsilon}\right)^d$$

With $\epsilon = 0.1$ (10 points per axis):

d	N_{grid}	
3	10^3	✅ Easy
10	10^{10}	⚠️ Expensive
30	10^{30}	❌ Impossible
100	10^{100}	❌❌ Absurd

```
1D: • • • • • (10)

2D: • • •
• • •
• • • (100)

3D: [cube of 1000 points]

↓

continues exponentially...
```

Why ML Works: Data Lives on Low-Dimensional Manifolds

> Physical data occupies a **tiny, structured subset** of the full high-dimensional space.

Ising Model Example:

- Configuration space: $\{-1, +1\}^{L \times L}$ → for $L = 32$: 2^{1024} possible states
- Thermodynamically relevant: only those weighted by $p(\sigma) \propto e^{-\beta H(\sigma)}$
- Effective dimension: (T, m) — a **2-dimensional manifold**

```
Full space (2^1024 states)

┌──────────────────────┐
│                      │
│                      │
│     ┌────────┐       │
│   / ●●●●●●● \ Data   │
│  |  ●●●●●●●● |lives│
│   \ ●●●●●●● / here   │
│     └────────┘       │
│                      │
│                      │
└──────────────────────┘
```

> Neural networks are **implicitly learning this manifold structure**.

ML Training ≡ Physical Optimization

Physics (Variational Principle)

$$E[\Psi_\theta] = \frac{\langle \Psi_\theta | \hat{H} | \Psi_\theta \rangle}{\langle \Psi_\theta | \Psi_\theta \rangle} \geq E_0$$

- Trial wavefunction Ψ_θ
- Minimize energy expectation
- $\theta \leftarrow \theta - \eta \nabla_\theta E[\Psi_\theta]$

ML (Loss Minimization)

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \ell\big(f_\theta(x_i),\, y_i\big)$$

- Neural network f_θ
- Minimize loss function
- $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}(\theta)$

Identical mathematical structure: parameter space + objective function + gradient descent.

Inductive Bias: The Key to Good ML

> **Inductive bias**: The set of assumptions a learning algorithm uses to predict outputs for unseen inputs. *"What shape of function do you expect before seeing any data?"*

```
    ●              ●              ●
   /              /|\            ~~~
●  ——  ●        ● | ●          ● ● ●
```

Linear	**Polynomial**	**Periodic**
"straight line"	"smooth curve"	"repeating pattern"

- Same data, different assumptions → completely different predictions
- No Free Lunch Theorem: **no learning without assumptions**
- **The better the physics, the stronger the inductive bias**

> For physicists: inductive bias = **symmetry, conservation laws, locality**

Physics Priors → Neural Network Architectures

Physical Prior	Math	NN Architecture	Course
Translation symmetry	$f(x+a) = f(x)$	**CNN** (shared weights)	Wk 4
Permutation symmetry	$f(P\mathbf{x}) = f(\mathbf{x})$	**GNN** (message passing)	Wk 11
Rotation/reflection	$f(R\mathbf{x}) = D(R)f(\mathbf{x})$	**E(3)-equivariant NN**	Wk 12
Energy conservation	$dE/dt = 0$	**Hamiltonian NN**	Wk 6
Fermion antisymmetry	$\Psi(\ldots i \ldots j \ldots) = -\Psi(\ldots j \ldots i \ldots)$	**Determinant layer**	Wk 5
Probability normalization	$\int p(x)dx = 1$	**Normalizing flow**	Wk 14

This table is the backbone of this course. We will return to it every week.

Example: CNN Encodes Translation Symmetry

MLP (no symmetry)

- Input: flatten 32×32 spin lattice to 1D vector
- Same domain pattern at different positions → different input → different output

```
[██□□□□...] → MLP → "ordered"
[□□██□□...] → MLP → "???" ← different input!
```

"MLP treats domains at (3,5) and (20,15) as completely different patterns"

CNN (translation equivariant)

- Same filter scans all positions → position-independent features

```
[██□□□□...] → CNN → "ordered"
[□□██□□...] → CNN → "ordered" ✓
```

"CNN detects the domain regardless of where it appears"

> CNN's weight sharing = translation invariance **built into** the architecture.

The Power of Encoding Symmetry: 1000× Less Data

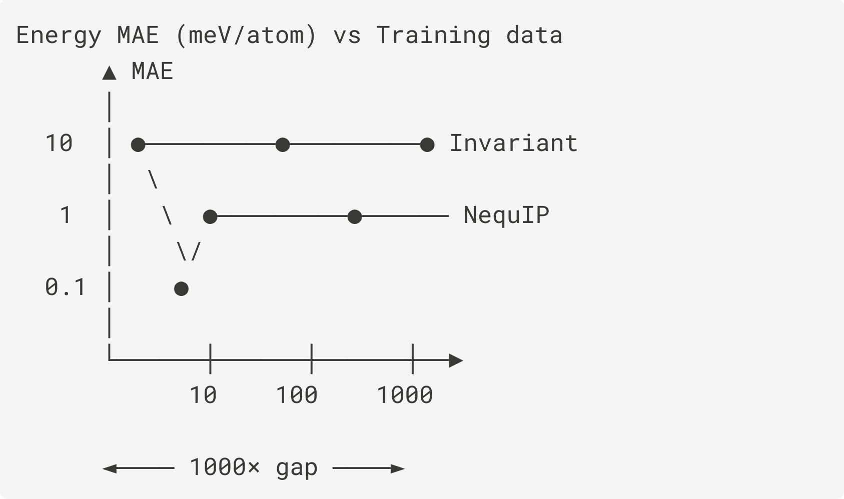
Molecular/crystal energy is invariant under rotation, reflection, translation:

$$E(R\mathbf{x} + \mathbf{t}) = E(\mathbf{x})$$

Forces are equivariant under rotation:

$$\mathbf{F}(R\mathbf{x}) = R \cdot \mathbf{F}(\mathbf{x})$$

NequIP: encodes this symmetry exactly in GNN architecture



Same accuracy with 1000× less training data by encoding rotation symmetry into the architecture.

Two Ways to Handle Symmetry

Data Augmentation (brute force)

- Method: duplicate training data with rotations/reflections
- ✅ Simple to implement, works with any model
- ❌ Training data $\times|G|$, training time $\times|G|$
- ❌ Symmetry learned only **approximately**
- ❌ Continuous symmetries (SO(3)) impossible to cover exactly

Equivariant Architecture (by design)

- Method: each NN layer commutes with symmetry transforms
- ✅ No additional data needed
- ✅ Symmetry guaranteed **exactly**
- ✅ Generalization is automatic
- ❌ Architecture design is more complex

Lesson: When you know the symmetry, encode it in the architecture. Data augmentation is a fallback, not a solution.

Translating Physics Problems into ML Tasks

Given a physics problem, **what kind of ML task is it?**

1. **Approximation** → "Represent a complex function"
2. **Inference** → "Infer hidden variables from observations"
3. **Sampling** → "Generate samples from a distribution"

> Every physics-ML project starts by identifying which task type applies.

Task Type 1: Approximation — Represent Complex Functions

> Learn a parametric function $f_\theta : \mathcal{X} \to \mathcal{Y}$ from data or by optimization.

In physics: represent complex functions where traditional basis expansions fail

Physics Problem	Input $\mathcal{X}$	Output $\mathcal{Y}$	Model
Interatomic potential	Atomic positions $\{\mathbf{r}_i\}$	Energy E, Forces $\mathbf{F}$	NequIP, MACE
Quantum wavefunction	Electron coords $\mathbf{r}_1, \ldots, \mathbf{r}_N$	$\Psi(\mathbf{r})$	FermiNet
DFT Hamiltonian	Crystal structure	H_{ij} matrix elements	DeepH

> **Bottleneck solved**: Basis expansion fails exponentially; NNs provide compact, flexible representations.

Task Type 2: Inference — From Observations to Hidden Variables

Given observations $\mathbf{d}$, estimate hidden parameters $\boldsymbol{\phi}$:

$$P(\boldsymbol{\phi} \,|\, \mathbf{d}) \propto P(\mathbf{d} \,|\, \boldsymbol{\phi})\, P(\boldsymbol{\phi})$$

Physics Problem	Observations $\mathbf{d}$	Hidden $\boldsymbol{\phi}$	ML Method
Quantum state tomography	Measurement outcomes	Density matrix ρ	Generative NN
Phase classification	Spin configurations	Phase label, T_c	CNN classifier
Simulation-based inference	Detector data	Theory parameters	Neural posterior

Bottleneck solved: Likelihood-free inference when $P(\mathbf{d} \mid \boldsymbol{\phi})$ is intractable.

Task Type 3: Sampling — Generate Configurations from Distributions

> Generate independent samples $\mathbf{x} \sim p(\mathbf{x})$ where $p(\mathbf{x}) = \frac{1}{Z}e^{-\beta E(\mathbf{x})}$

Traditional: MCMC (Metropolis, HMC) → **autocorrelation**, **critical slowing down** near T_c

ML solution:

- **Normalizing flow**: simple distribution → complex distribution, **one-shot** independent samples
- **Diffusion model**: start from noise → gradually generate structure

Examples:

- Lattice ϕ^4 theory (Albergo et al. 2019) → critical slowing down eliminated
- Boltzmann generator (Noe et al. 2019) → one-shot protein conformations

Three ML Task Types: Summary

ML Task	Physics Question	What NN Does	Key Example	Course
Approximation	Represent complex function	$f_\theta(\mathbf{x}) \approx y$	MLIP, NQS	Wk 2–6, 11–12
Inference	Infer hidden variables	$P(\boldsymbol{\phi} \mid \mathbf{d})$	QST, SBI	Wk 4–5, 9–10
Sampling	Generate from distribution	$\mathbf{x} \sim p(\mathbf{x})$	Flow, Diffusion	Wk 13–14

> Many real problems **combine** these tasks. E.g., Neural network quantum states = Approximation + Sampling (VMC).

Summary & What's Next

Today's Key Messages:

1. **AI for Science** works through three paradigms: pattern recognition, surrogate modeling, agents
2. **Physics modeling vs ML**: both solve optimization in parameter space — identical mathematical structure
3. **Inductive bias** is the bridge: encoding symmetry into NN architectures yields dramatic improvements
4. **Three ML task types** — approximation, inference, sampling — framework for any physics → ML translation

Next:

- **Now**: Hands-on — Gradient descent on potential energy surfaces (JAX notebook)
- **Week 2** (3/11, 3/13): Neural Networks Basics — linear models → MLP → backpropagation

References

- Carrasquilla, J. & Melko, R. G. Machine learning phases of matter. *Nature Physics* **13**, 431 (2017)
- George, D. & Huerta, E. A. Deep neural networks to enable real-time multimessenger astrophysics. *Phys. Rev. D* **97**, 044039 (2018)
- Batzner, S. et al. E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature Communications* **13**, 2453 (2022)
- Pfau, D. et al. Ab initio solution of the many-electron Schrodinger equation with deep neural networks. *Phys. Rev. Research* **2**, 033429 (2020)
- Li, H. et al. Deep-learning density functional theory Hamiltonian for efficient ab initio electronic-structure calculation. *Nature Comp. Sci.* **2**, 367 (2022)
- Albergo, M. S. et al. Flow-based generative models for Markov chain Monte Carlo in lattice field theory. *Phys. Rev. D* **100**, 034515 (2019)
- Noe, F. et al. Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning. *Science* **365**, eaaw1147 (2019)
- Cybenko, G. Approximation by superpositions of a sigmoidal function. *Math. Control Signal Syst.* **2**, 303 (1989)