

# Machine Learning Applications in Physics

PHYG004 — 2026 Spring

**Prof. Young Woo Choi**

Department of Physics, Sogang University

Wed & Fri, 09:00 – 10:15

3/4/2026

# Instructor

---

<b>Name</b>	Young Woo Choi (최영우)
<b>Email</b>	<a href="mailto:ywchoi02@sogang.ac.kr">ywchoi02@sogang.ac.kr</a>
<b>Office</b>	F303C
<b>Office Hours</b>	Mon, Tue, Thu 10:00–12:00 (by appointment)
<b>Homepage</b>	<a href="https://yw-choi.github.io">https://yw-choi.github.io</a>

---

Please make an appointment by email before visiting.

# Course Overview

- ML from a physicist's perspective: how neural networks can study quantum, statistical, and condensed matter systems
- Topics: basic neural networks, transformers, symmetry-aware models, generative models
- Course format: 50% lectures + 50% student presentations
- No single textbook — lecture notes, coding examples, and selected papers provided throughout
- Hands-on coding with JAX in every lecture

# Course Objectives

1. Understand the basic concepts and architectures of deep learning from a physics perspective
2. Learn how deep learning models are applied to problems in quantum, statistical, and condensed matter physics
3. Gain experience formulating physics problems as machine learning tasks and implementing solutions using JAX
4. *Learn how to effectively use AI agents for programming and best practices for AI-assisted development*

# Grading

Component	Weight	Details
<b>Individual Presentation</b>	30%	Week 7, 3-minute oral presentation
<b>Term Project</b>	50%	Team-based (up to 3), presented in Week 15
<b>Participation</b>	20%	Attendance, in-class engagement

- No midterm or final exams

# Course Schedule (Weeks 1–8)

Week	Topic
1 (03/04)	Why Machine Learning in Physics?
2 (03/11)	Neural Networks Basics
3 (03/18)	Optimization, Generalization, and Inductive Bias
4 (03/25)	Convolutional Neural Networks
5 (04/01)	FNN/CNN Applications in Physics (paper review)
6 (04/08)	Automatic Differentiation & Differentiable Physics
7 (04/15)	<b>Individual Presentations</b>
8 (04/22)	<i>Midterm Week (no exam)</i>

# Course Schedule (Weeks 9–16)

Week	Topic
9 (04/29)	Attention and Transformers
10 (05/06)	Transformer Applications in Physics (paper review)
11 (05/13)	Symmetry in Machine Learning
12 (05/20)	Equivariant Neural Networks
13 (05/27)	Generative Models I (AE, VAE)
14 (06/03)	Generative Models II (Flows, Diffusion)
15 (06/10)	<b>Term Project Presentations</b>
16 (06/17)	<i>Final Exam Week (no exam)</i>

# Individual Presentation (Week 7)

- **3-minute** oral presentation (slides optional)
- **Goal:** Propose an ML approach to a physics problem you care about

## Suggested structure

1. **Problem** — What physics question are you trying to answer?
  2. **Why ML?** — Why is this hard with traditional methods?
  3. **Approach** — What type of model / data would you use?
  4. **Expected outcome** — What would success look like?
- No code or implementation needed — this is about **ideas**
  - Can draw from your own research, coursework, or a paper you've read
  - This presentation can serve as a starting point for your term project

# Term Project (Week 15)

- **Team-based:** up to 3 students per team
- Apply machine learning to a physics problem:
  - Reproduce results from a research paper, **or**
  - Original exploratory study
- Final presentation in Week 15
- More details will be provided later in the semester

# Registration (Shared Google Sheet)

- Register your **GitHub ID** to the shared sheet (due today)
- Register your topic for **individual presentation** (due Week 6)
- **Team formation** for term project (due Week 7)

# Policy on AI Usage

- Use of AI tools (ChatGPT, Copilot, Claude, etc.) is **strongly encouraged**
- Practicing with AI agents (VS Code + Copilot Agents) is a key part of this course
- However, you must **always understand** the output AI generates for you
- Always ask yourself: **how would I verify this result?**

AI amplifies your ability.  $0 \times 100 = 0$

# Development Environment

## Option A: VS Code + GitHub Copilot (Recommended)

- Full-featured IDE with AI-assisted coding
- Free for students via GitHub Education Pack
- Works with local Python / JAX installation

## Option B: Google Colab

- No local setup required
- Free GPU access (limited)
- Great for quick experimentation

Bring your laptop to class — we'll do hands-on exercises!

# GitHub Education Pack

Get **free GitHub Copilot and Copilot Chat** access as a student:

1. Go to [education.github.com](https://education.github.com)
2. Click "**Join GitHub Education**"
3. Verify your student status with your Sogang email
4. Once approved, activate **GitHub Copilot** in your account settings
5. Install the **GitHub Copilot extension** in VS Code

This gives you access to **Copilot Agents** — AI that can write, run, and debug code for you!

# Course Tools

Tool	Purpose
<b>JAX</b>	Primary ML framework (autodiff, JIT, GPU)
<b>NumPy</b>	Array operations, data preparation
<b>Matplotlib</b>	Visualization and plotting
<b>Equinox</b>	Neural network library for JAX
<b>Optax</b>	Gradient-based optimization

All tools are open-source and free. We'll set everything up today!

# Why JAX?

- **Functional programming** style — natural for physicists (think: mathematical functions)
- **Automatic differentiation** — `jax.grad` differentiates any Python function
- **JIT compilation** — `jax.jit` compiles functions for GPU/TPU speed
- **Vectorization** — `jax.vmap` auto-vectorizes over batch dimensions
- **NumPy-compatible** API — easy to learn if you know NumPy
- Widely used in physics ML research

$$\text{Force} = -\nabla V \quad \longrightarrow \quad \text{force} = -\text{jax.grad}(V)$$

# Today: Environment Setup

In the hands-on tutorial, we will:

1. **Install JAX** and verify GPU/CPU backend
2. **Basic operations** — arrays, math, linear algebra
3. **Random numbers** — JAX's explicit PRNG system
4. **NumPy vs JAX** — key differences (immutability)
5. **Automatic differentiation** — computing gradients with `jax.grad`
6. **JIT compilation** — speed comparison

Open the tutorial notebook on Google Colab or your local environment.