PHY 591 — Applications of AI Methods Across STEM Disciplines

AAI4STEM · Spring 2027

Instructor and class information

Instructor	Michael Pürrer — mpuerrer@uri.edu
Meetings	Tuesday and Thursday, 11:00 AM–12:15 PM, Swan Hall 211
Office hours	Tuesday and Thursday 1:00–2:00 PM, or by appointment
Credits	3
Enrollment	capped at 20
First / last class	Tue Jan 26 / Thu Apr 29, 2027 · Spring break Mar 22–28 · Final report due Tue May 11
Course site	Brightspace (readings, submissions, announcements) · course repository (all code and notebooks)

Course description

This graduate course, open to master's and PhD students in any STEM field, explores how modern artificial intelligence (deep learning and its generative and physics-informed extensions) is changing research across the STEM disciplines. It is a *methods* course taught through *applications*: every technique is introduced from its mathematical foundations, implemented from scratch in PyTorch, and then put to work on a real scientific problem where a classical method already exists and has to be beaten, or honestly not beaten.

Part I (weeks 1–4) gives the background the method modules need, following Bishop & Bishop, *Deep Learning: Foundations and Concepts* (2024). **Weeks 1–2 are a refresher of the probabilistic toolkit**: probability, the Gaussian, linear regression and the bias–variance trade-off, in the notation used for the rest of the course. They are a review, not a substitute for a course in statistics or multivariate statistical learning. **Weeks 3–4 are a compressed primer on deep networks**: multilayer networks and their error functions, gradient descent, backpropagation and regularization. For more depth, we point to advanced CS&S courses such as CSC 561 (Neural Networks and Deep Learning) and CSC 492 (Large Language Models).

Part II (weeks 5–12) is a sequence of *method modules*, each pairing an architecture with a scientific domain and, where possible, a guest domain expert:

week	module	method	domain
5	MP1	Graph neural networks	molecular property prediction (QM9)
6	MP2	Physics-informed neural networks	coastal ocean dynamics (shallow-water equations)
7	MP3	Neural ODEs & universal differential equations	epidemics; measles in London 1944–64
8–9	MP4	Convolutional networks and classification; transformers	genomic sequences (splice-site recognition)
10	MP5	Normalizing flows & simulation-based inference	gravitational-wave parameter estimation (black-hole ringdown)
11	MP6	Diffusion models	satellite sea-surface temperature; cloud gap-filling
12	MP7	Reinforcement learning	building energy control under time-of-use tariffs

Week 13 is devoted to team presentations of the capstone projects.

Learning outcomes

By the end of the semester you will be able to

1. explain the core concepts of deep learning — gradient descent, backpropagation, regularization, inductive bias — and compare architectures (MLPs, CNNs, GNNs, transformers) in terms of what each assumes about the data;
2. implement, train and evaluate neural networks in PyTorch, on laptop / URI's Unity cluster;
3. apply advanced methods (GNNs, PINNs, neural ODEs, transformers, normalizing flows, diffusion models and RL) to data from scientific and engineering domains;
4. evaluate a model the way a scientist must: against a named classical baseline, against the physics (conservation laws, symmetries, known consensus, information-theoretic bounds), and with honest uncertainty and multiple seeds;
5. tell, with evidence, when a machine-learning method is *not* the right tool;
6. work in an interdisciplinary team to design, execute and present a small research project.

Prerequisites

Graduate standing (master's or PhD) in a STEM discipline, or permission of the instructor. You are expected to have

- **mathematics:** undergraduate linear algebra (vectors, matrices, eigenvalues), multivariable calculus (gradients, partial derivatives), basic probability and statistics (distributions, expectation, Bayes' rule);
- **programming:** working Python (functions, NumPy, Matplotlib, Jupyter);
- **recommended, not required:** prior exposure to supervised learning or basic neural networks. Students with CSC 461 or CSC 561 will find weeks 1–4 a quick review and can go straight to the modules.

If you have not used PyTorch, plan to work through the official *60-minute blitz* before week 2.

Texts and resources

- **Primary text:** Bishop & Bishop, *Deep Learning: Foundations and Concepts*, Springer 2024 — free online at bishopbook.com. Part I follows Ch. 1–4 and 6–9; Part II modules cite it where it covers the method (GNNs Ch. 13, classification Ch. 5, CNNs Ch. 10, transformers Ch. 12, flows Ch. 18, diffusion Ch. 20).
- **Supplementary readings** for methods not in Bishop (PINNs, neural ODEs, reinforcement learning): one foundational paper and one review per module, posted on Brightspace.
- **Software:** Python 3 and PyTorch. Every module ships a from-scratch starter notebook and a small library; you may use higher-level libraries (PyTorch Geometric, DeepXDE, torchdiffeq, Hugging Face, Stable-Baselines3, …) in projects **provided you cite them and can explain what they do**.
- **Computing:** you will receive an account on URI's **Unity** HPC/AI cluster (GPU partitions and the `phy591` environment). Several modules run in minutes on a laptop; the module handouts say which.

How the course works

Part I: lectures, labs, homework (weeks 1–4)

Tuesday sessions are lectures on the week's Bishop chapters; Thursdays are hands-on labs in PyTorch (bring a laptop). Three problem sets combine Bishop exercises with short coding tasks.

Part II: method modules (weeks 5–12)

Each week: **Tuesday — method seminar** (the instructor develops the method from Bishop or the anchor paper; when a domain expert visits, they present the science and their own use of the method); **Thursday — lab** working through that module's starter notebook. Every starter notebook builds the method from scratch, applies it to a real scientific dataset, compares it against a classical baseline, and checks it against the physics of the problem.

Projects: teams, one method each, two tiers

In week 3 the class forms **seven teams of 2–3** and each team adopts **one module's project** from the project bank (`projects/`). The team's project has two tiers:

- **Base tier — the mini-project** (graded, due about ten days after the module's lab week): reproduce the starter, carry out two of the handout's modifications with proper measurement, and evaluate against the baseline and the physics. This is the team's major deliverable for Part II.
- **Stretch tier — the capstone**: the same problem taken to research scope (real data, a published number reproduced, a method extended), presented in week 13 and written up as a 5-page paper.

Every student also completes a **short lab exercise** for each of the *other* modules (a few hours each, individual, graded lightly; the two-week MP4 has one per week), so everyone works with every method.

Team projects can start before their method is taught: each project dossier separates the groundwork that needs no method (data, classical baseline, evaluation metric) from the parts that do. Teams whose method comes late in the term (diffusion, RL) receive an early primer.

Assignments and grading

component	weight	what it consists of
Homework and labs	25 %	three Part I problem sets (HW1–3) and the Part II lab exercises for the other modules
Mini-project (base tier)	40 %	the team's module project: notebook + 2-page write-up, graded on the module rubric
Capstone (stretch tier)	25 %	proposal (pass/revise), checkpoint (5 %), presentation (5 %), 5-page paper + repository (15 %)
Participation and discussion	10 %	seminar and lab engagement, peer questions at presentations, project stand-ups

Team submissions receive a team grade; a short individual contribution statement accompanies each team deliverable, and grades may be adjusted for documented disparities.

Final grades: A 94–100 · A– 90–93 · B+ 87–89 · B 83–86 · B– 80–82 · C+ 77–79 · C 73–76 · C– 70–72 · D+ 65–69 · D 60–64 · F below 60.

Schedule — Spring 2027

Dates from the official URI 2026–27 academic calendar. Classes begin Tue Jan 26; spring break Mar 22–28; last day of classes Mon May 3; final examinations May 6–7 and 10–12. Every teaching week has both a Tuesday and a Thursday session. Chapter numbers refer to Bishop & Bishop (2024).

wk	Tue · Thu	topic	reading	due (Sundays 23:59 unless noted)
1	Jan 26 · 28	*Refresher:* course overview; the deep learning revolution; probability	Ch. 1–2	Lab 0: environment + Unity smoke test (Thu Jan 28)
2	Feb 2 · 4	*Refresher:* the Gaussian; linear regression; bias–variance	Ch. 3–4	**HW1** (Ch. 1–4), Sun Feb 7
3	Feb 9 · 11	*DL primer:* deep neural networks, error functions, output layers; gradient descent & SGD	Ch. 6 (+ Ch. 7 start)	**Team formation & project choice**, Tue Feb 9

wk	Tue · Thu	topic	reading	due (Sundays 23:59 unless noted)
4	Feb 16 · 18	*DL primer:* optimizers, backpropagation & autodiff, regularization	Ch. 7–9	**HW2** (Ch. 4, 6), Sun Feb 21 · **HW3** (Ch. 7–9), Sun Feb 28
5	Feb 23 · 25	**MP1 · Graph neural networks** — molecules	Ch. 13; Gilmer+ 2017	Lab 1 · MP1 base tier Sun Mar 7
6	Mar 2 · 4	**MP2 · Physics-informed NNs** — coastal waves	Raissi+ 2019; Karniadakis+ 2021	Lab 2 · MP2 base tier Sun Mar 14
7	Mar 9 · 11	**MP3 · Neural ODEs** — measles epidemics	§18.3; Chen+ 2018	Lab 3 · MP3 base tier Sun Mar 21
8	Mar 16 · 18	**MP4a · Convolutional networks & classification** — splice sites	Ch. 10; §5.2, §5.4	Lab 4a · **Capstone proposal, Thu Mar 18**
—	Mar 22–28	*Spring break — no classes*		
9	Mar 30 · Apr 1	**MP4b · Transformers** — genomic sequences	Ch. 12; Vaswani+ 2017	Lab 4b · MP4 base tier Sun Apr 11
10	Apr 6 · 8	**MP5 · Normalizing flows & SBI** — gravitational waves	Ch. 18; Cranmer+ 2020	Lab 5 · MP5 base tier Sun Apr 18
11	Apr 13 · 15	**MP6 · Diffusion models** — sea-surface temperature	Ch. 20 (+ Ch. 14); Ho+ 2020	Lab 6 · **Capstone checkpoint stand-ups, Thu Apr 15** · MP6 base tier Sun Apr 25
12	Apr 20 · 22	**MP7 · Reinforcement learning** — building control	Sutton & Barto Ch. 1, 3, 6, 13; Schulman+ 2017	Lab 7 · MP7 base tier Sun May 2
13	Apr 27 · 29	**Capstone presentations** (teams 1–4 Tue, 5–7 Thu); wrap-up	—	Slides, Mon Apr 26
—	Exam period	*No class meetings*		**Capstone paper + repository, Tue May 11, 23:59**

Guest domain experts are scheduled for Tuesday seminars in Part II; the module topics stand regardless of the guest schedule, which will be announced on Brightspace.

Capstone project

Each team's stretch tier. Deliverables and dates:

when	deliverable	weight
Thu Mar 18	**Proposal** (1 page): problem, data, method, classical baseline, success criterion, compute estimate	pass / revise
Thu Apr 15	**Checkpoint**: data in hand, baseline running, 5-minute stand-up	5 %
Apr 27 / 29	**Presentation**: 10 minutes + 4 minutes of questions	5 %
Tue May 11	**Paper** (5 pages, conference style) + notebook or repo	15 %

Example stretch projects: amortized inference on a real gravitational-wave event; a physics-informed network on a real coastal bathymetry; a universal differential equation for a biological kinetic model with symbolic recovery; a pretrained DNA language model fine-tuned on a regulatory-genomics benchmark; conditional diffusion for satellite gap-filling with real cloud masks; a reinforcement learning controller on the CityLearn benchmark. Every paper must state the classical baseline it was measured against and cite all external code, models and data.

Policies

Collaboration. Discussion is encouraged. Homework and lab exercises are individual: you may discuss approaches but must write your own code and text. Team deliverables are shared work within the team; do not share code or write-ups across teams before the deadline.

Use of AI coding assistants and language models. This is a course about AI, and you will use AI tools; the rules are the same as for any external help. (1) Cite them: name the tool and how you used it in a short statement at the end of each submission. (2) You must be able to explain every line you submit — I may ask you to, in office hours or at your presentation, and "the model wrote it" is not an explanation. (3) Never paste a problem set question into a tool and submit its answer; the point of the exercise is your understanding. (4) Verify everything: cited papers must exist, quoted numbers must be reproduced by your own code. Undisclosed use, or submitted work you cannot explain, is treated as academic dishonesty.

Late work. Up to 48 hours late: −10 % per day. Beyond 48 hours: no credit unless arranged in advance. Team deadlines are firm; one free 48-hour extension per team for a base-tier deliverable may be claimed by email before the deadline.

Academic integrity. URI's Student Code of Conduct applies. Plagiarism, fabrication of results, or unauthorized sharing of code will be referred to the Office of Student Conduct.

Reproducibility. Every submitted result must come with the code and the seeds that produce it. Claims about stochastic methods (RL, generative models) require multiple seeds with spread reported; the module rubrics deduct points for single-seed claims.

Accessibility. Students with documented disabilities should contact me as early as possible so that reasonable accommodations can be arranged, and should be in touch with Disability, Access, and Inclusion (302 Memorial Union, 401-874-2098).

Computing resources. You will have an account on URI's Unity cluster. Store data and outputs in the course project space, not in your home directory; do not run training jobs on login nodes; and use the cpu partition for modules that do not need a GPU (the handouts say which).

Syllabus version of 8 October 2026; dates follow the official URI academic calendar.