

# CS690/SWE699: FORMAL REASONING IN AI AND SOFTWARE ENGINEERING (SWE)

Spring '26

<b>Meetings:</b>	Tues 4:30PM – 7:10PM	<b>Place:</b>	PLANET 224
<b>Instructor:</b>	<a href="#">ThanhVu Nguyen</a>	<b>Email:</b>	<a href="mailto:tvn@gmu.edu">tvn@gmu.edu</a> (Canvas is preferred)
<b>Office Hr:</b>	(email to set appointment)	<b>Place:</b>	Zoom or ENGR 4430
<b>GTA:</b>	TBD	<b>Email:</b>	<a href="mailto:tbd@gmu.edu">tbd@gmu.edu</a>
<b>Office Hr:</b>	TBD	<b>Place:</b>	in-person or Zoom (email for info)

## 1 Description

**Course Overview** This special topic course is a **research seminar** on the application of *Formal Method* (FM) techniques to AI and Software Engineering (SWE) problems, with a focus on *Deep Neural Networks* (DNNs) verification and assurance.. AI, in particular DNNs, have emerged as an effective approach for solving challenging real-world problems. power grid control, fake news detection, drug synthesis and discovery, and COVID-19 detection and diagnosis. However, just like traditional software, DNNs can have “bugs”, e.g., producing unexpected results on inputs that are different from those in training data, and be attacked, e.g., small perturbations to the inputs by a malicious adversary or even sensorial imperfections result in misclassification. These issues, which have been observed in many DNNs and demonstrated in the real world, naturally raise the question of how DNNs should be tested, validated, and ultimately *verified* to meet the requirements of relevant robustness or safety standards.

In this class, we will learn various techniques and tools to verify software, and treat DNNs as special software systems. We will cover topics including the applications of verification, testing, analysis, constraint solving, and abstraction techniques to DNNs such as Feedforward Neural Networks (FNNs), Residual Networks (ResNet), and Convolutional Neural Networks (CNNs). We will focus on scalable and precise techniques that can deal with large, real-world DNNs.

The course will focus on active research areas in formal software and DNN reasoning, but the specific topics will be largely determined by a combination of instructor fiat and the interests of the students.

**Learning Outcomes** By the end of the course, students will gain a solid understanding of FM and their applications, particularly in the context of AI and software engineering.

### 1.1 Learning Outcomes

- **Understanding of AI Verification Techniques and Tools:** Students will gain a deep understanding of FM verification concepts, techniques, and tools. They will learn how to apply these techniques to various types of neural networks, including FNNs, REsNets, CNNs, and RNNs. The student will also learn how to use existing tools to analyze and verify DNNs.
- **In-Depth Understanding:** For the final project, students will delve into DNN verification. They will not only understand the theory but also be able to provide concrete examples and *implement* the technique themselves, gaining a comprehensive understanding of the topic.
- **Strengthen knowledge in Linear Algebra, AI/ML, and Programming:** Students will develop or strengthen their foundational knowledge in linear algebra and AI/ML, making them

well-prepared to tackle advanced topics and real-world problems in AI safety. Programming assignments will require students to implement AI analysis techniques in Python. This will improve their programming skills and their ability to apply theoretical concepts to practical problems.

- **Critical Reading and Evaluation:** Through weekly reading assignments, students will learn to critically evaluate both book chapters and research papers related to AI verification and analysis. They will be able to identify the problem addressed, assess proposed solutions, analyze the strengths and weaknesses of different approaches, and evaluate and compare related techniques.
- **Presentation and Discussion Skills:** Students will have the opportunity to lead group discussions and presentations on assigned readings. This will enhance their presentation and communication skills, as well as their ability to facilitate meaningful discussions among peers.

## 1.2 Prerequisite

- No prerequisite courses. However, basic knowledge in linear algebra and AI/ML, e.g., CS 580, is strongly recommended
- Programming knowledge (Python): we will use Python for assignments and projects.
- *Important:* This class does *not* assume prior knowledge of any AI or deep learning topics, e.g., how NNs are created and trained, as all background material will be provided. It also does not teach these topics and instead focuses on formal analysis of given NNs.

## 1.3 Course Materials

- T. Nguyen and H. Duong, *Engineering a Verifier for Deep Neural Networks* (**Required**, free)
- A. Albarghouthi. *Introduction to Neural Network Verification*. 2021. (**Recommended**, free)
- Other useful resources:
  - [Neural Network Verification Tutorial](#)
  - [Advanced Topics in ML and Formal Methods](#) (Grad level course at UIUC)

## 1.4 Assignment Submission and Communication

We will use **Canvas** for communication and submitting assignments, and to keep track of grades (§2). It's the student's responsibility to ensure that your grade records are correct.

There are two types of assignments in this course: hand-written assignments and code/programming assignments. **DO NOT** submit link to an image, document, file; you will receive a 0 for the assignment if you submit a link.

**Hand-written Assignments** All assignments except programming ones (discussed below). Quizzes will be done in class on paper. Homework assignments will be done on paper (or tablet) and submitted as image or PDF files to Canvas. More details below.

**Code/Programming Assignments** You will submit your `src code` files (Python files) and README file (in PDF or Markdown format) to Canvas. You can submit a single ZIP file containing all files. Instructions are provided in each assignment.

## 2 Grading

Assignments	Percentage
Participation	5%
Homework (HW) Assignments	35%
Quizzes	40%
Programming Assignments (PAs)	20%
<b>Total</b>	100%

### Scales

A+	$\geq 97\%$	A	$\geq 93\%$	A-	$\geq 90\%$
B+	$\geq 87\%$	B	$\geq 83\%$	B-	$\geq 80\%$
C	$\geq 70\%$	D	$\geq 60\%$	F	$< 60\%$

**Groups** For quizzes, you will work individually, but for HW and PAs you can team up with another student (a group of 2). You will need to submit a *single group* solution for each assignment. You also need to include a statement of contributions from each group member. Once you form a group, you cannot change it and will work with your group on all assignments.

### 2.1 Participation

I place great emphasis on peer learning and interactive engagement. The class is structured to leverage interactions to the largest extent possible. Often, there will be in-class exercises for every class. The in-class exercises will be closely related to an upcoming homework assignment.

### 2.2 Homework Assignments

There are weekly *group homework assignments*, which are given in **Schedule** section of the DNN verification book. Homework assignments are due **before class**, i.e., before 4:30 PM on the day we meet. Late submissions are *not* accepted except in truly exceptional circumstances.

- Homeworks to be written by hand (on paper, tablet, etc – but must be handwritten). When done, you submit by taking a picture of your answers and submitting the picture (as a PDF or image file) to Canvas.
- You will be asked present their homework solution in class.
- There are **no make-ups** and **absolutely no late submissions** because we present solutions in class.
- If work in group then everyone in the group gets the same credit.

## 2.3 Quizzes

We will have a short quiz every week. Each quiz is worth 10 pts. The quiz will be based on the material covered in the previous weeks. Each quiz happens during class. *You must be present to take the quiz.*

- Quizzes will be **handwritten on papers, which are then submitted when you're done**, so be sure to bring writing materials to class.
- We will **drop your lowest two** quiz scores.
- **No make-ups** for missed quizzes.

## 2.4 Programming Assignments (PA's)

This course consists of several Programming Assignments (PA's) in Python. These PAs are designed for you to practice and implement the techniques learned in class. *All assignments have similar grading weights.*

Your submissions will be evaluated for correctness, answers to solutions, and documentation. We will not attempt to fix broken submissions that fail to execute properly; only limited partial credit will be given in such situations. Assignments are due at **11:59pm** on the due date.

- **No late submissions**

## 3 GMU Policies

### 3.1 Honor Code

As with all GMU courses, this class governed by the [GMU Honor Code](#). In this course, all assignments carry with them an implicit statement that it is the sole work of the author.

### 3.2 Uses of AI Tools

When explicitly stated by the instructor, Generative AI tools are allowed on the named assignment. Students will be directed if and when citation or statement-of-usage direction is required. Use of these tools on any assignment not specified will be considered a violation of the academic standards policy. All academic standards violations will be reported using the Academic Standards Referral Form.

### 3.3 Learning Disabilities

Disability Services at George Mason University is committed to providing equitable access to learning opportunities for all students by upholding the laws that ensure equal treatment of people with disabilities. If you are seeking accommodations for this class, please first visit <https://ds.gmu.edu/> for detailed information about the Disability Services registration process. Then please discuss your approved accommodations with me. Disability Services is located in Student Union Building I (SUB I), Suite 2500. Email: [ods@gmu.edu](mailto:ods@gmu.edu) — Phone: (703) 993-2474