CS 6170: Computational Topology
Topology Meets Machine Learning
Course Syllabus, Spring 2026

Instructor: Dr. Bei Wang Phillips
beiwang@sci.utah.edu

Course Information

Meeting Time: Tuesdays, Thursdays, 3:40 p.m. - 5:00 p.m.
Classroom: MEB 2325
Textbook: Computational Topology: An Introduction, by Herbert Edelsbrunner and John Harer
Course Public Web Page:
`http://www.sci.utah.edu/~beiwang/teaching/cs6170-spring-2026.html`
Additional Materials for Students Available via Canvas.
Contact Information:

Instructor: Dr. Bei Wang Phillips
Office: WEB 4608
Email: beiwang@sci.utah.edu
Office Hours: See course webpage for details.

1 Course Description

Computational topology lies at the intersection of algebraic topology and computer science and serves as the theoretical foundation for Topological Data Analysis (TDA). TDA is an emerging area within exploratory data analysis and data mining that has attracted growing interest and achieved notable successes across an expanding research community. The application of topological techniques to large-scale and complex data has opened new opportunities in science, engineering, and business intelligence.

The goal of TDA is to understand complex datasets, where complexity arises not only from data scale but also from the richness and structure of underlying features. The objective of this course is to familiarize students with modern methods in computational topology and TDA from theoretical, algorithmic, and applied perspectives.

Successful completion of the course will prepare students to become data practitioners capable of applying TDA pipelines to a wide range of real-world datasets in areas such as materials science, biomedicine, and business intelligence. Students will also be positioned to pursue new research directions in computational topology and TDA, as well as to integrate advanced topological techniques with other areas of data science, including data mining, machine learning, computer graphics, geometric modeling, mesh generation, and data visualization.

For Spring 2026, the course will additionally cover recent advances at the intersection of topology and machine learning, with particular emphasis on topological deep learning.

Suggested Topics: The course materials are organized under three mutually inclusive modules:

- Topological Foundations and Pipeline (FP)
- Topology Meets Machine Learning and Statistics (ML)
- Topological Data Analysis in Data Science (DS)

The course may cover (but is not limited to) the following topics:

- Basic concepts (graphs, connected components, topological space, manifold, point clouds)
- Combinatorial structures on point cloud data (simplicial complexes)
- New techniques in dimension reduction (circular coordinates, etc.)
- Clustering (topology-based data partition, classification)
- Homology and persistent homology
- Topological signatures for classification
- Structural inference and reconstruction from data
- Topological algorithms for massive data
- Multivariate and high-dimensional data analysis
- Topological data analysis for visualization (vector fields, topological structures)
- Practical applications of TDA
- Topological deep learning

2 Prerequisites

There are no formal prerequisites for this class. Students will be expected to have basic knowledge of data structures and algorithmic techniques.

The targeted audience for the class includes PhD students, master students and very-motivated upper level undergraduate students. The students are not required to be majoring in Computer Science, but it is preferable that the students have some background in algorithms and/or other data science related courses. If you are not sure whether you are qualified to take this class, please email the instructor.

3 Course Grading

- 2 assignments in the form of mini-projects (40 points, 40%; each project is worth 20 points). These projects are labeled as **Project 1 and 2** respectively in the course schedule.
- 1 final project (60 points, 60%). This is labeled as the **Final Project** in the course schedule.
  - Final project proposal (15 points, 15%)
  - Final project report (30 points, 30%), including the progress report (10 points) and the final report (20 points)
  - Final project presentation (15 points, 15%)

- Additional 10 bonus points may be available in the form of bonus assignment questions.

Scale for assigning letter grades is as follows (based on points). This scale might be curved based on overall class performance, while ensuring fairness to all.
A 100-93 **A-** 93-90
B+ 90-87 **B** 87-83 **B-** 83-80
C+ 80-77 **C** 77-73 **C-** 73-70
D+ 70-67 **D** 67-63 **D-** 63-60
E 60-0

Assignment Policies:

- Assignments are required to be done individually. Final projects can be done in groups. In general, discussing topics is allowed. However, the copying of each others' work is considered cheating and will result in a failing grade.
- There will be a call for assignments to be submitted (see course schedule for details). Assignments must be turned in at the beginning of class (i.e. at the time requested) on the day in which they are due.
- Most assignments should be submitted via Canvas. Each submission typically requires the following components (see each assignment description for details):
  - A PDF file (for project report)
  - A ZIP file for source code (if the assignment includes programming)
  - A URL that allows access to the deployed online software, or a URL that contains a link to a video that captures the software in action (these URLs are to be included in the submitted PDF file)
- Students are expected to submit completed assignments by the due date and time. To get full credit for an assignment, it must be turned in through Canvas by the start of class. Once the deadline is missed, those turned in late will lose 10% of its total points for each subsequent hour until it is turned in. Therefore, assignments will not be accepted more than 10 hours late, and will be given 0.
- For the portion of an assignment involving programming, if the programs do not execute, no partial points will be given. To demonstrate the execution of the program, a URL that allows access to the deployed online software, or a URL that contains a link to a video that captures the software in action should be provided.
- Please allocate sufficient time for completing the class assignments.
- For assignments (not including the final project), students can have a one-time two-day extension without penalty; please use this exception wisely.[1].

[1]The one-time two-day extension is applicable for all assignments except for final project presentation and final project report.

- For assignments, typesetting (Latex, MS Word, ... even a typewriter if you can find one) is required. Assignments deemed unreadable will be rejected at the time of collection; they can be resubmitted, but with the late penalty applied per the previously mentioned policy.

4 Policies and Guidelines

Please read carefully the School of Computing (SoC) policies and guidelines at:
`https://handbook.cs.utah.edu/`.
Please see also the College of Engineering Semester Guidelines at:
`https://www.price.utah.edu/semester-guidelines`.
These guidelines contain important dates regarding adding, dropping and withdrawing from classes as well as the College Policy regarding repeating courses.

5 Final Project

5.1 Project Description

Your final project may be designed from the perspective of a data practitioner, a developer, or a data theorist.

As a data practitioner, you may work with interesting and nontrivial datasets drawn from real-world application domains (e.g., marketing, scientific simulation, transportation, and business intelligence) and apply emerging and innovative TDA techniques—possibly in combination with data mining and machine learning methods—to extract meaningful insights. The datasets must be of genuine real-world relevance and may not be toy or artificially simplified datasets.

As a developer, you may design new software tools or extend existing ones that integrate TDA with data mining and machine learning techniques.

As a data theorist, you may pursue new theoretical developments, algorithms, or data structures in the field of computational topology and TDA.

You are responsible for pitching a project idea that is appropriate for your background and level of preparation. While you are encouraged to challenge yourself, you should also remain realistic in scope. If you underestimate the difficulty of your project, you must still ensure that a complete and meaningful deliverable is submitted by the due date; selecting an overly ambitious project is not a valid justification for an incomplete submission.

5.2 Project Team

You will work in a team with two members (forming a team with one member, or a team with more than two members will require the permission from the instructor). You can form the team on your own; or the instructor could make some suggestions based on your submitted CV. Please try to form your project team as early as possible.

5.3 Important Dates

There are several milestones for the final project.

- Project team creation, typically in February, 2026.
- Project proposal description, typically in March, 2026.
- Project progress report, typically in March/April, 2026.
- Project presentations, typically in April/May, 2026.
- Project final report, typically in April/May, 2026.

See Canvas for details on the due dates.