Session K Abstracts

Exploration of the Learning From Hints Paradigm
Paul Huang
Mentor: Yaser S. Abu-Mostafa

In machine learning, the goal is often to approximate some unknown target function by utilizing learning algorithms to exploit the information contained in the training data distribution. While raw training data serves as a foundational input, there is potential in utilizing transformed versions of this data. Such transformations convey human knowledge about the target function, acting as "hints" for the learning algorithms. Although the concept of using hints in learning algorithms has been widely studied in machine learning literature, most works focused on data augmentation, also known as duplicate examples. This narrowed focus has ignored the potential of another effective hinting technique - the deployment of virtual examples. Virtual examples offer the advantage of guiding the model to learn hints in a broader input domain, as opposed to the limited scope of duplicate examples confined to the training data distribution. Our research focuses on the utilization of virtual examples as a hint-providing mechanism, testing their efficacy against using duplicate examples as hint-providers. We evaluate these techniques based on the resulting model's performance and robustness on both data highly correlated to the training set and out-of-distribution data that models have not seen in the training stage.

A Type Checker for Mathematical Proofs Written in LaTeX
Zachary Huang
Mentor: Adam Blank

The primary goal of this project is to answer the question: how can we best apply modern programming language technologies in order to help students learn proof-based discrete mathematics? A tool which could analyze student proofs written in LaTeX and identify common mistakes such as type mismatches and circular logic would be invaluable to the learning process and help students improve the rigor of their mathematical proofs. The objective is to implement such a tool and test its effectiveness in CS 13, a new course on discrete mathematics intended for CS majors. LaTeX, however, is quite difficult to analyze statically, but if we ignore the advanced features of LaTeX, we can parse out the overall textual and mathematical content of proofs using a context-free grammar. We can then fit a Hindley-Milner-like type system onto the extracted mathematical content to perform type inference/checking in a similar manner to functional programming languages. We cover the main “data types” of discrete math, including numbers (naturals, integers, rationals, and reals), sets, and sequences. In addition, the system can process arithmetic operations, inequalities, set operations, and more. Then, pattern-matching can be used to perform ad-hoc proof recognition and analysis.

Blowup Analysis for the Weak Convection Hou-Li Model
Xiang Qin
Mentor: Thomas Y.Hou

We study the a-parameterized weak advection version of a 1D model proposed by Hou-Li. This model is based on an approximation of the axisymmetric Navier-Stokes equations in the r direction. By considering the fixed-point problem of some nonlinear map, we show that the weak convection model admits exact self-similar finite-time blowup solutions with smooth profiles for 2/3<a<1. We also verify and visualize our theoretical results using a direct numerical iterative method. Our work reproduces the previous results in a purely analytic approach, and obtain more properties of the profiles.

Investigating Potential Finite-Time Blowup in Models for 3D Euler Singularity Using Fixed-Point Method
Xiuyuan Wang
Mentor: Thomas Y. Hou

The fundamental question on the global regularity of the 3D Euler and Navier-Stokes equations with smooth initial data remains one of the most challenging open problems in fluid dynamics. Huang et.al. have developed a fixed-point method to prove the existence of self-similar finite time blowup in 1D generalized Constantin-Lax-Majda model, which is a simplified model for vorticity formulation of the 3D incompressible Euler equations. Our current work aims to further advance this line of research by extending this method to multi-dimensional cases and applying it to investigate possible finite-time singularities in several simplified models proposed to study 3D Euler singularities, both numerically and theoretically.

Creating an Intuitive XR Space for Efficient 3D Data Inspection and Direct Manipulation Supported by 2D and 3D Interfaces
Sam Johnson-Lacoss
Mentors: Santiago Lombeyda and S. George Djorgovski
There are certain phenomena that in order to be fully understood, one must first grasp their three-dimensional spatial relations. For these situations, two dimensional screens and a passive point-of-view fail to allow a viewer to fully utilize the extent of our cognitive faculties to understand three dimensional structures and retain those intricacies. As such, extended reality environments often prioritize the third dimension in representations of data, in the attempt to provide clearer insight into these phenomena. However, this priority often neglects data that is more effectively conveyed by both two-dimensional and complex connected three-dimensional representations, rendering such data difficult to navigate in an efficient or meaningful way. This project defines an intuitive and efficient extended reality space for object-focused visualization that combines two- and three-dimensional interaction techniques as well as representations, to create a tool for more efficient and meaningful data analysis.

Curating a Retrospective Art Exhibit for the Caltech/JPL Data to Discovery Program
Ethan McFarlin
Mentors: Santiago Lombeyda and Hillary Mushkin

Melding the spirit of scientific inquiry with foundational research in human-computer interaction and visual perception, Caltech’s Data to Discovery (D2D) Initiative aims to demystify complex data sets through the design of interactive visualization systems. Past projects launched by the initiative vary precipitously not only in their means of visualization but also their underlying scientific focus— from tracking the movement of the Mars’ Exploration Rover to simulating glacier flow in Antarctica. The wide breadth of projects undertaken combined with the 10 year anniversary of the program engenders the opportunity to revisit unexpected parallels and insights uncovered by D2D. This paper describes the development of a comprehensive retrospective to revisit and communicate those visual and scientific insights in a manner that is widely digestible to those outside the field. The process of creating the retrospective involves curating raw material for an interactive art exhibit and thematically analyzing connections between projects.

Scalable Gaussian Processes for Non-ergodic Earthquake Models
Stephen Huan
Mentors: Houman Owhadi and Grigorios Lavrentiadis

Non-ergodic ground-motion models attempt to estimate the probability an earthquake will exceed a certain intensity and are used to assess the seismic hazard at nuclear power plant locations, among other applications in civil engineering. However, current codes scale cubically with the number of data points, severely limiting the size of processable datasets. This work applies recent state of the art results in sparse approximate Cholesky factors to significantly accelerate the computations necessary in statistical inference. Our method enjoys a number of advantageous properties, including near-linear time and space efficiency, optimality in Kullback–Leibler divergence, and embarrassingly parallel computation. We develop two code repositories: SparseKolesky.jl, a user-friendly, well-documented library for sparse factorization of generic covariance matrices and EarthquakeGPs.jl, specializing our methodology to earthquake models. We write our codes in Julia, the high-performance scientific programming language with the intention of targeting supercomputing clusters. The project website and additional resources may be found at https://kolesky.cgdct.moe/.

Verifying Web Browser Security With Cachet, a JIT Compiler DSL for Expressive Static Assertions
June Woodward
Mentors: Deian Stefan and Richard M. Murray

Implementations of dynamically-typed programming languages often perform inline caching, which takes advantage of observed regularities in an executing program and speeds up the “fast path” by replacing a function call with the type-specific method that it delegates to. However, since a dynamically-typed program is not required to maintain these regularities, implementations must also preface each inline cache with a guard, ensuring that a method is not called on the wrong type. SpiderMonkey’s just-in-time (JIT) compiler has experienced multiple bugs and vulnerabilities from subtle mismatches between these assumptions and checks. The PLSysSec team at UCSD has made Cachet, a domain-specific language (DSL) where these invariants can be expressed and statically verified, and partially rewritten SpiderMonkey’s CacheIR-to-MASM compilation step in Cachet. In my SURF project I continued to apply Cachet to SpiderMonkey, and improved the build system and documentation.