Archana Warrier

✉ archanarw@gmail.com github.com/archanarw

Education

Technische Universität Kaiserslautern (RPTU) *Apr 2022 – Present*
Master of Science in Computer Science *Current GPA: 1.6/5.0 (German scale)*
- Coursework: Machine Learning I & II, Monte Carlo Algorithms, Automated Reasoning, Functional Programming, 3D Computer Vision, Stochastic Modeling of Cognitive Processes

Birla Institute of Technology, Mesra *June 2018 – June 2021*
Bachelor of Science in Mathematics and Computing *GPA: 9.02/10*

Indian Institute of Technology Madras *Jan 2021 – Aug 2021*
Foundational Course on Data Science *GPA: 9.4/10*

Publications

- **Benchmarking World-Model Learning** (Under review)
  ***A. Warrier**, D. Nguyen, M. Naim, M. Jain, Y. Liang, K. Schroeder, C. Yang, J. Tenenbaum, S. Vollmer, K. Ellis, Z. Tavares*
  - Developed the WorldTest framework and designed the AutumnBench benchmark for evaluating world model building in interactive agents
  - Project website: autumn.basis.ai
- **Had Enough of Experts? Elicitation and Evaluation of Bayesian Priors from LLMs**
  NeurIPS 2024 Workshop | D. Selby, K. Spriestersbach, Y. Iwashita, D. Bappert, **A. Warrier**, S. Mukherjee, K. Kise, S. Vollmer

Research Experience

Research Trainee *Aug 2024 – Present*
Basis, New York | Advisors: Zenna Tavares, Kevin Ellis
- Developing AI systems for scientific reasoning as part of Project MARA (Modeling, Abstraction, and Reasoning Agents)
- Contributing to MARAProtocol and Autumn.cpp frameworks for world model evaluation and learning

Research Assistant *May 2022 – July 2024*
DFKI, Kaiserslautern | Advisor: Prof. Dr. Sebastian Vollmer
- Conducted research on probabilistic inference methods and Bayesian machine learning
- Served as teaching assistant for the “Machine Learning in Julia” course

Research Scholar *May 2023 – Oct 2023*
Columbia University, New York | Advisor: Zenna Tavares
- Developed a framework that automatically constructs simplified model abstractions to balance computational accuracy with resource constraints
- Extended causal abstraction theory to adapt abstractions dynamically based on query requirements and available computational resources
- Implemented proof-of-concept for adaptive abstraction in physics-based simulations using MuJoCo

Research Scholar *Sept 2022 – Nov 2022*
Columbia University, New York | Advisor: Zenna Tavares
- Implemented parametric inversion in JAX — a program transformation technique that inverts non-injective functions by introducing parameters to uniquely determine inputs
- Demonstrated approach on simulation models, including Lotka-Volterra dynamics
- Contributed to Omega.jl probabilistic programming framework

Student Developer, Google Summer of Code *June 2021 – Aug 2021*
The Julia Language Organization *Advisors: Zenna Tavares, Sebastian Vollmer, Moritz Schauer, Jiahao Chen*
- Developed CounterfactualFairness.jl package for counterfactual reasoning methods for algorithmic fairness

- Implemented multiple fairness criteria, including counterfactual fairness and path-specific effects
- Blog post: nextjournal.com/archanarw/counterfactualfairnessjl

Technical Skills

- **Programming Languages:** Julia, Python, C, Haskell, Agda, Isabelle
- **Tools:** LaTeX, Git, MATLAB

Teaching & Service

Teaching Assistant, "Machine Learning in Julia" *May 2022 – Aug 2024*
- DFKI, Kaiserslautern | Assisted students with probabilistic programming concepts and Julia implementation

Content and Publicity Head, IEEE Student Branch *2020 – 2021*
- Birla Institute of Technology | Conducted workshops and organized technical events

Mentor, MTTS Program *2020*
- Supported students in the Level 0 advanced mathematics program

Selected Honors

- Top 300, National Programming and Data Science Qualifier, IIT Madras (95% score) *2020*
- Received a letter of appreciation from the Union Minister of India, Smriti Irani *2015*
- Bronze Medal, International Olympiad of Mathematics *2013*

Open Source Contributions

- **Active Contributor:** MARAProtocol, Autumn.cpp, Omega.jl, Jaxy.jl
- **Maintainer:** CounterfactualFairness.jl, NormalizingFlows.jl