

Jaykumar Bhagiya

Robotics & Machine Learning

📧 jayBhagiya | 🌐 jay-bhagiya | ✉ jaykumarbhagiya@gmail.com

EDUCATION

- **Universität des Saarlandes** Saarbrücken, Germany
MSc. in Computer Science Oct 2023 - Oct 2026
 - **Courses:** Reinforcement Learning, High-Level Computer Vision, Image Processing and Computer Vision, Neural Networks: Theory and Implementation, Optimization for Machine Learning
 - **Seminars:** Machine Learning for Natural Language Processing, Explainable Machine Learning

SKILLS SUMMARY

Programming Languages	Python, C / C++, Java
Frameworks	PyTorch, Transformers, PyTorch Geometric, scikit-learn, OpenCV, Open3D, ROS / ROS2, PyQt5, MoveIt, OMPL
Tools	Bash / Zsh, Git / GitHub, Linux, Docker, CMake, LaTeX, Simulators (Gazebo, RedMax, Isaac Sim)
Hardware Platforms	Robots (UR10e, MiR100, Turtlebot), Depth cameras (Intel RealSense D405, Microsoft Azure Kinect), LIDAR, PhaseSpace motion capture, Jetson Nano, Raspberry Pi
Spoken Languages	English (C1), German (A1), Hindi, Gujarati
Other Skills	Version Control, Debugging & Profiling, Software Development, Documentation, Technical Writing

WORK EXPERIENCE

- **ZeMA gGmbH** Saarbrücken, Germany
Research Assistant April 2024 - Present
 - Developed **FAST**, a **Graph Transformer** that predicts the physical feasibility of disassembly actions for **automated assembly-sequence planning**, removing per-candidate physics simulation from the planning loop.
 - Trained FAST on a large simulated dataset of feasible and infeasible actions labelled by a physics-based **assemblability and stability** checker, so the planner learns directly from failed attempts.
 - Contributed to a **point-cloud** perception and **collision-free trajectory-planning** pipeline for picking parts from cluttered, reconfigurable bins, with live replanning when the scene changes.
 - Integrated a **digital twin** to verify planned motions before execution and authored the planning code connecting **perception, planning, and control** into an end-to-end pipeline on a real robot arm.
- **Robert Bosch Centre for Cyber-Physical Systems, IISc** Bangalore, India
Research Assistant June 2022 - April 2023
 - Developed a **safe autonomous-exploration** framework for ground robots, pairing target detection with an **abstraction-based (symbolic) controller** for provably safe navigation in unknown environments.
 - Built **distributed controller synthesis** for multi-agent systems, combining symbolic control with **barrier certificates** to enforce local temporal-logic goals alongside global safety.
 - Parallelized expensive **symbolic-control synthesis** to cut controller-synthesis time, and took graduate coursework in **parallel computing** to implement it.
- **Robert Bosch Centre for Cyber-Physical Systems, IISc** Bangalore, India
Project Intern Jan 2022 - May 2022
 - Modeled unknown nonlinear dynamics with **Gaussian processes** on small datasets via combined kernels, then solved the reachability problem with a **funnel-based controller**.
 - Built **ROS** support for a **PhaseSpace** motion-capture system for ground-truth localization, and implemented control algorithms for robots under complex specifications.

PUBLICATIONS

- **FAST: Action-aware Feasibility in Assembly Sequence Planning using Graph Transformer** [↗](#)
IEEE International Conference on Intelligent Robots and Systems (IROS 2026), accepted
- **Collision-Free Trajectory Planning in Cluttered Environments for Efficient Bin Picking** [↗](#)
IEEE International Conference on Intelligent Robots and Systems (IROS 2025)
- **Automated Assembly Picking System for Cluttered Environments with Live Collision-Free Trajectory Planning** [↗](#)
IEEE Conference on Emerging Technologies and Factory Automation (ETFA 2025)
- **Scalable Distributed Controller Synthesis for Multi-Agent Systems using Barrier Functions and Symbolic Control** [↗](#)
IEEE Conference on Decision and Control (CDC 2023)
- **Autonomous Exploration using Ground Robots with Safety Guarantees** [↗](#)
IEEE International Conference on Intelligent Robots and Systems (IROS 2023)
- **Funnel-based Reachability Control of Unknown Nonlinear Systems using Gaussian Processes** [↗](#)
IEEE Indian Control Conference (ICC 2022)

PROJECTS

- **Learning Control: from Rules to Deep Q-Networks** [↗](#) Mar 2026
 - Rebuilt seven reinforcement-learning tasks into one reproducible progression, from fixed policies and **Monte Carlo control** to tabular **Q-learning** and a stabilized **Deep Q-Network** with experience replay and a target network.
 - Showed concretely where each method breaks: on Racetrack only replay plus a target network learned reliably across seeds, while online DQN and replay-alone collapsed (Python, PyTorch, Gymnasium).
- **Classical Image Processing in Modern C++** [↗](#) Jul 2025
 - Implemented 20+ classical operators from scratch, from **Fourier/DCT** analysis to **Canny** edges, the **Hough** transform, morphology, optical flow, and stereo.
 - Packaged them as a dependency-free toolkit with interactive, side-by-side comparisons that make each parameter's effect visible (C++20, CMake).
- **From Simulated Roads to Real-looking Street Scenes** [↗](#) Jul 2024
 - Benchmarked four image-translation families for turning synthetic driving scenes realistic, **U-Net**, **Pix2Pix**, a transformer bottleneck, and **CycleGAN-Turbo**.
 - Built a strict scene-held-out Virtual KITTI to KITTI split so test scenes never appear in training, and evaluated at the pixel, perceptual, and distribution level (Python, PyTorch, GANs, Diffusion).
- **Efficient XGLM Adaptation for Ayacucho Quechua** [↗](#) Mar 2024
 - Inspected the multilingual **XGLM** model's representation geometry layer by layer to see where languages align.
 - Adapted it to low-resource Ayacucho Quechua, comparing full fine-tuning against **LoRA**, **BitFit**, and **IA³** parameter-efficient methods (Python, PyTorch, Transformers, PEFT).
- **Graph Transformers on PATTERN** [↗](#) Mar 2025
 - Reimplemented a **Graph Transformer** in **PyTorch Geometric** and ran a controlled PATTERN ablation of sparse attention, **Laplacian positional encoding**, and normalization under matched settings (Python, PyTorch, PyTorch Geometric).
- **Integrated Gradients across Vision, Text, and Graphs** [↗](#) Aug 2024
 - Implemented **Integrated Gradients** from first principles with completeness checks and applied it across image, sentiment-text, and molecular-graph classifiers (Python, PyTorch, Explainable AI).
- **AdaGrad and Adam with Gradient Differences** [↗](#) Sep 2024
 - Modified **AdaGrad** and **Adam** so their adaptive denominators track successive gradient differences, evaluated from convex problems to **CIFAR-10 (ResNet-34)** and neural style transfer (Python, PyTorch).