Project Portfolio by Sebastian Jost

- Visual summaries of personal projects
- most important **goals**, **methods** & **results**
- **Time** spent working on a project includes all aspects of it: planning, implementation & documentation
- Projects are sorted by invested time
- Please request access if interested in private GitHub repositories

[images/ screenshots of program UI or results]

[*links to project files and demo videos*]

[timespan where I worked on the project]

Solving General Twisty Puzzles with RL

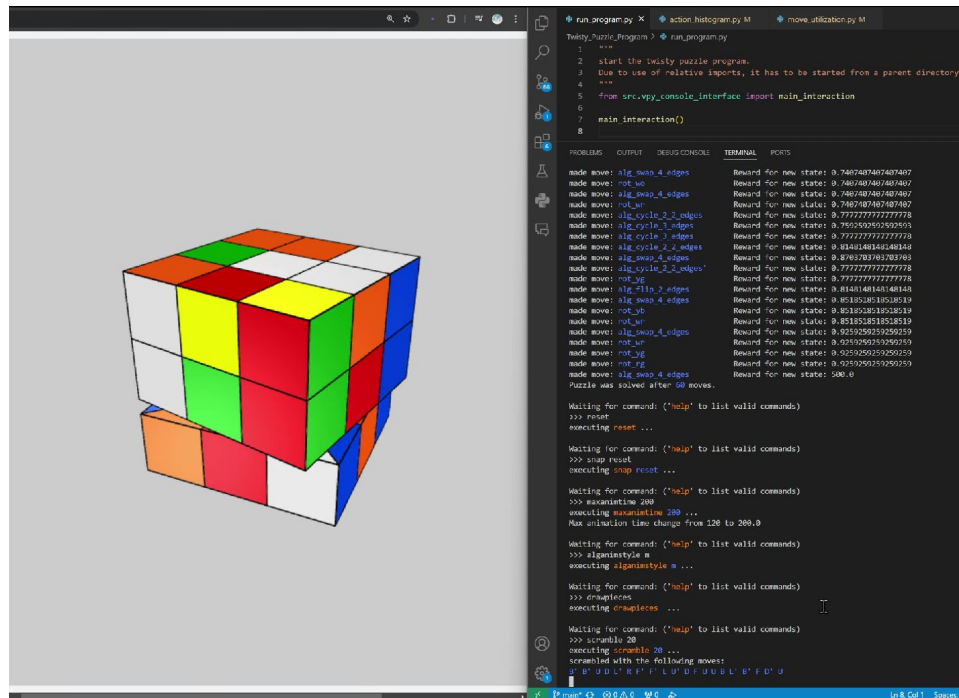**Goal:** Compute interpretable solutions on consumer hardware

Methods:
- Automatic action space augmentation
- Dense, intuitive rewards

Result:
- ~100x faster training than previous RL-approaches
- AI solutions similar to human ones

Time: ~600 hours incl. thesis

Aug 2020 - Dec 2024

Code on GitHub   *master's thesis*   *video explanation*

more demo videos

Custom Ticket to Ride Map Creation Tools

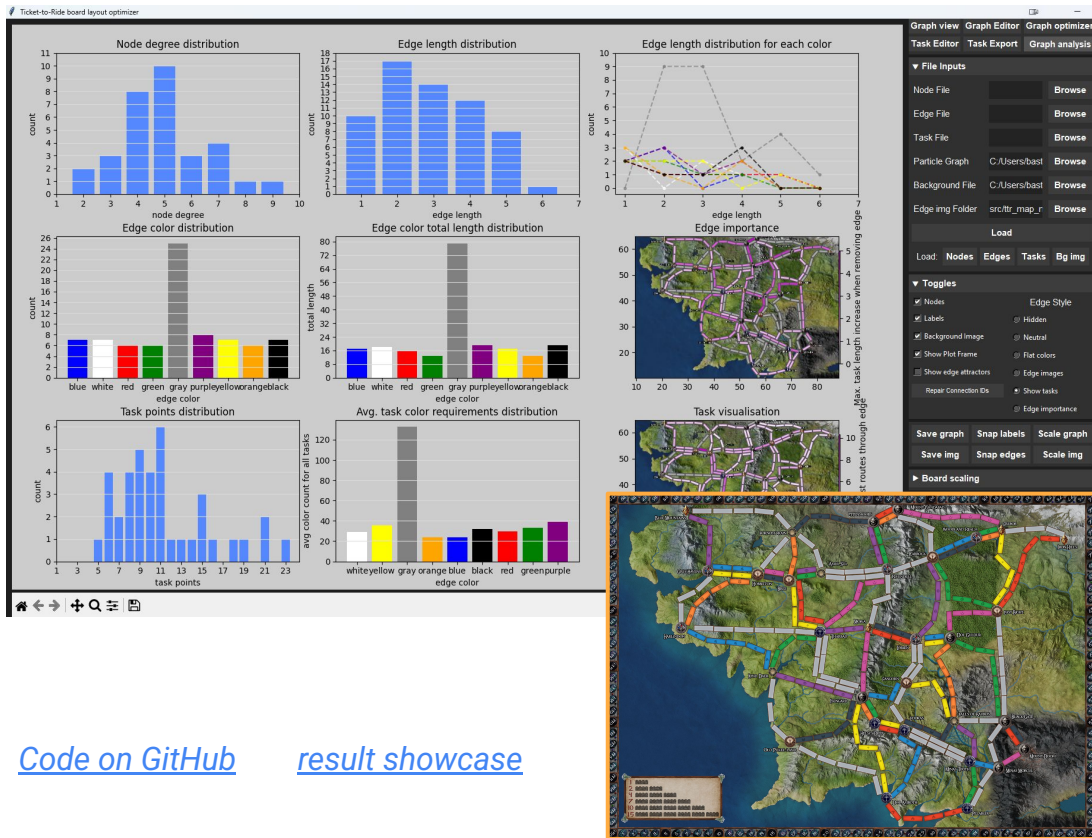**Goal:** Efficiently create custom Ticket to Ride versions

Methods:
- Analyze creation processes and bottlenecks
- Build tools to support creation at multiple stages

Result:
- Faster creation of variants
- High-quality custom games

Time: ~300 hours

Dec 2022 - Dec 2023

Code on GitHub   *result showcase*

Simulating real-world traffic sensor noise for more robust RL-Traffic Signal Control

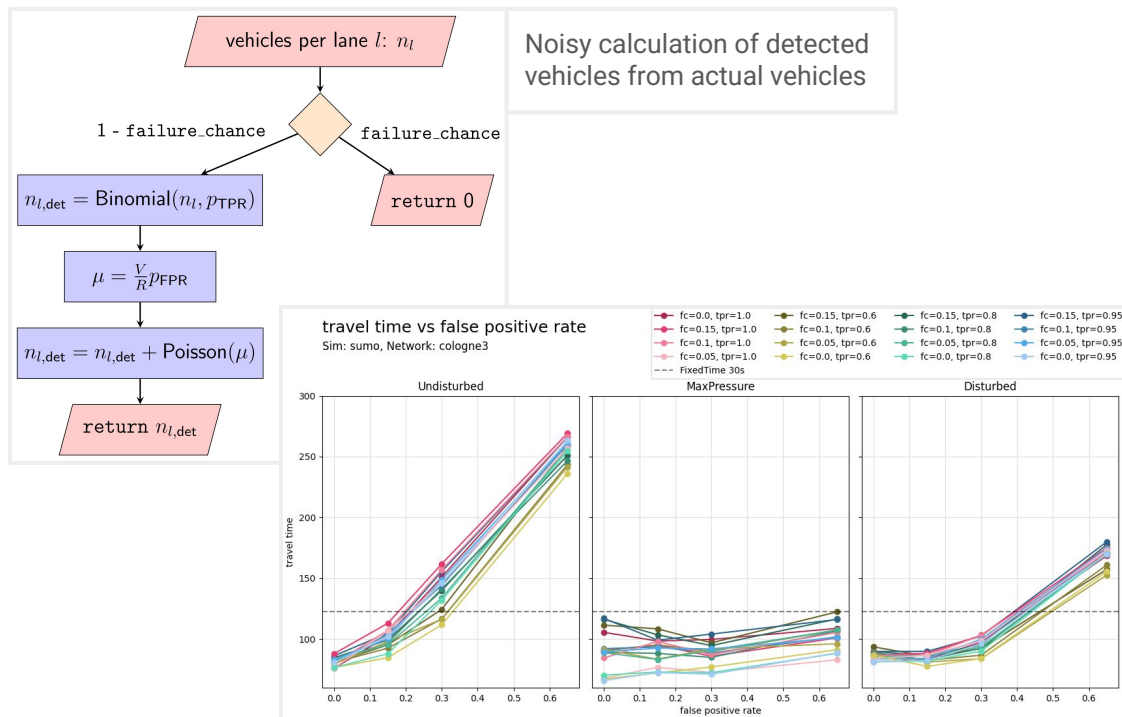**Goal:** Improve robustness of RL-TSC agents against noisy data

Methods:
- Simulate realistic sensor noise
- >600,000 simulated hours to test parameter settings and analyze their impact

Result:
- Understanding impact of different noise types on tested RL and non-RL agents
- Actionable sensor type deployment recommendations

Time: ~200 hours incl. writing paper

Noisy calculation of detected vehicles from actual vehicles

vehicles per lane l: n_l

$1 - \texttt{failure\_chance}$ $\texttt{failure\_chance}$

$n_{l,\text{det}} = \text{Binomial}(n_l, p_{\text{TPR}})$

$\texttt{return } 0$

$\mu = \frac{V}{R} p_{\text{FPR}}$

$n_{l,\text{det}} = n_{l,\text{det}} + \text{Poisson}(\mu)$

$\texttt{return } n_{l,\text{det}}$

travel time vs false positive rate
Sim: sumo, Network: cologne3

[Code on GitHub](#)   *publication planned*

Northgard Board Game with Web-UI

Goals:
- Recreate the board game Northgard digitally
- Write web-UI without prior JS/CSS/html knowledge

Methods:
- AI tools for writing frontend
- Python backend
- Custom tools for asset creation

Result:
- Prototype with responsive web-UI

Time: ~100 hours incl. asset creation

Dec 2024 - Sep 2025

[Code on GitHub](#) [demo video](#) [game info](#)

[Assets on GitHub](#)

Wizard Game with multiple AI players

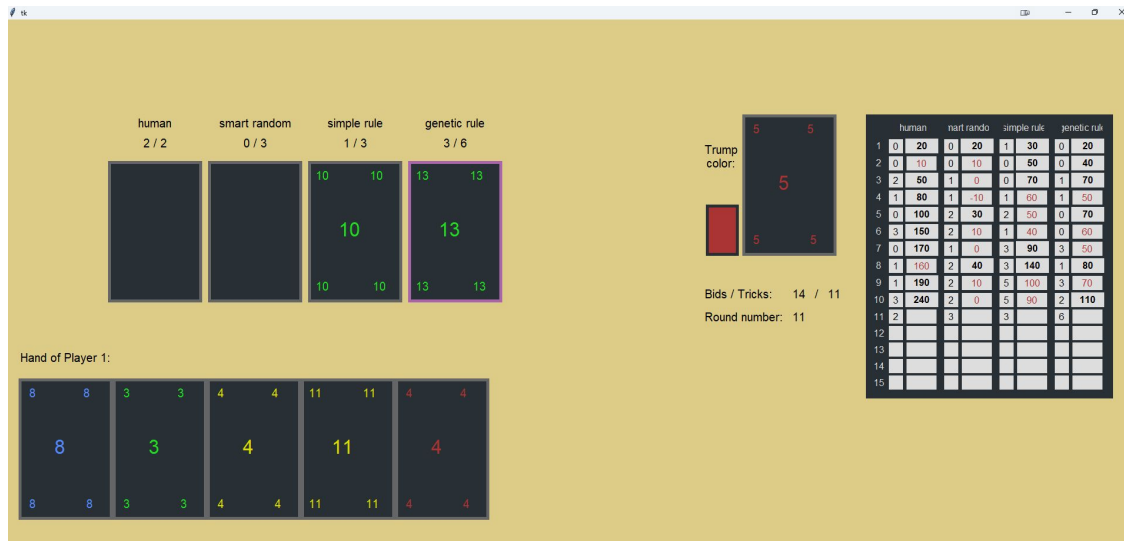**Goal:** Compare different AI techniques for creating wizard game agents

Methods:
- Rule-based AI, optimized via genetic evolution
- tkinter GUI design

Result:
- Major capability differences between AI agents

Time: ~60 hours

[Code on GitHub](#)

Simplified Saboteur Game with RL players

Goal:

- Investigate emergence of deceptive AI behaviour

Methods:

- LLM-accelerated coding

Result:

- RL-agents successfully play simplified game
- *unfinished*

Time: ~20 hours

Code on GitHub

demo video

Feb 2025

Maze Generator & Solver using Q-Learning

Goals:
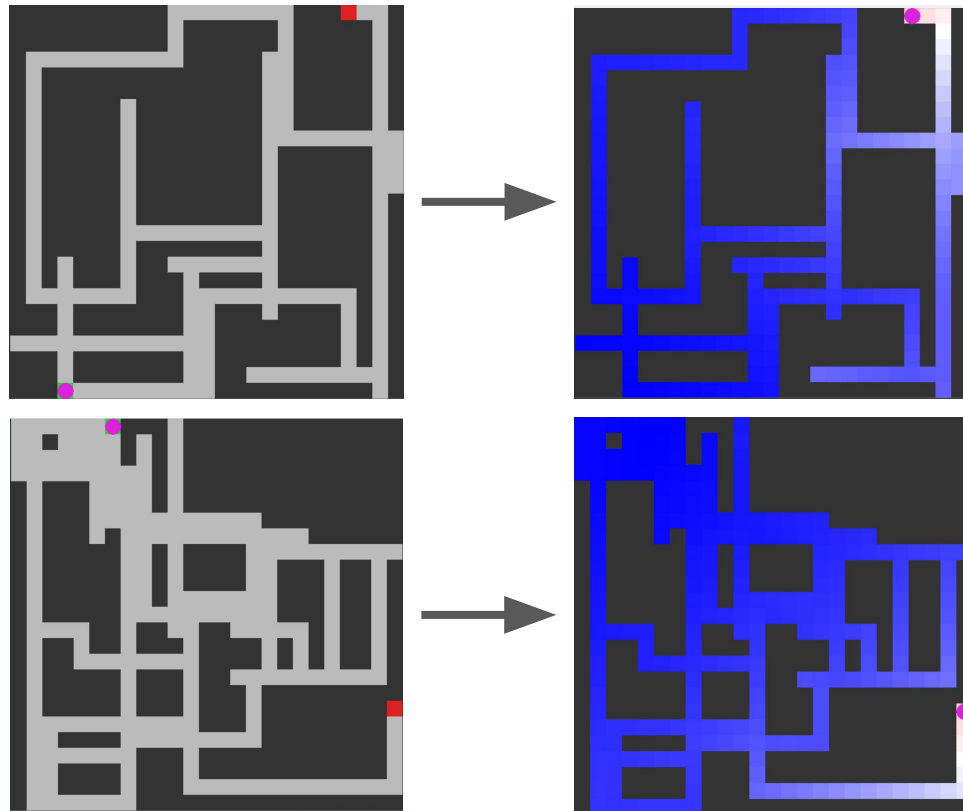- Automatic maze generation
- Q-learning based solver

Methods:
- Customized random walk
- Q-learning
- tkinter GUI design

Result:
- Reliable, efficient solver for small mazes

Time: 3 days ≈ 20 hours

Learning Methods in Humans, Animals and AI

Goal: Compare learning techniques in different organisms to inspire future research

Methods:
- Offered Workshop at VANconference25
- Years of following current research

Result:
- Overview helps predict industry & research direction

Time: <10 hours

Learning method	Animals	Humans	Computer (Machine Learning)
Evolution	genetic evolution	genetic evolution	genetic algorithms (e.g. NEAT)
Trial & Error	Conditioning	Conditioning	Reinforcement learning
Imitation	Orcas teaching hunting methods & language	Learning language	Supervised learning, Imitation learning
Abstract Instructions	Bee dance to describe location of food	Most of school / higher education, baking recipes	In-context learning in LLMs
Mental Simulation	Birds cracking nuts using cars, tool use	Dreams, pondering other's reactions	World models (e.g. DreamerV3)

GitHub page