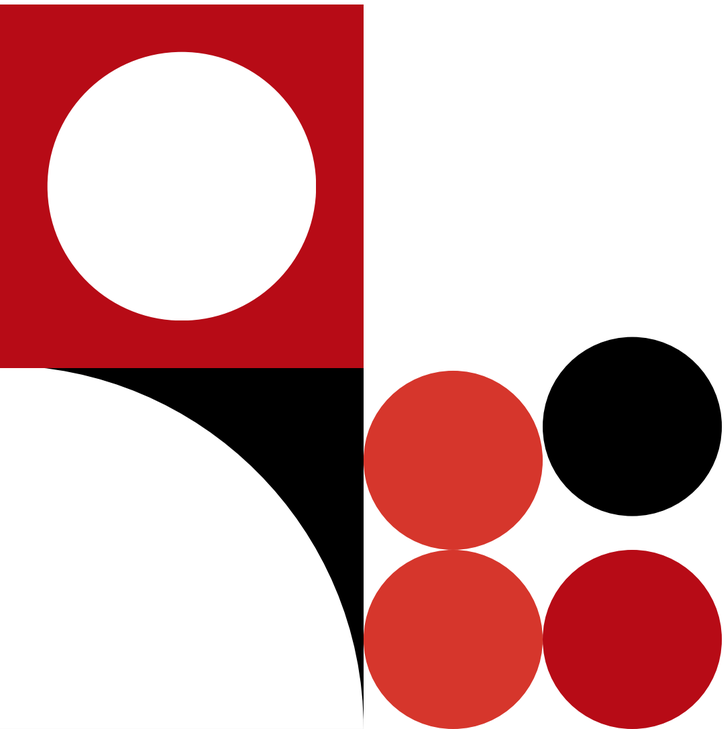
Honda Research Institute | LABS

SPRING 2026 UNIVERSITY INNOVATION IMPACT REPORT

CONTENTS



ACKNOWLEDGMENTS

We want to extend our heartfelt thanks to everyone at 99P Labs, Honda Research Institute, and all of our university partners. Your dedication and collaborative spirit have been instrumental in making these student innovation projects possible.

Thank you all for your hard work, creativity, and commitment to bridging industry and academia. Your contributions have made a real impact on our mission and will continue to inspire future projects. We're excited about what we're building together.

INTRODUCTION

Welcome to 99P Labs' Spring 2026 University Innovation Impact Report. This document is a window into the university engagements we joined during the spring 2026 semester.

This semester we also published a two-part blog series, "Rethinking the Industry–Academia Relationship" (Part 1, Part 2), on the thinking behind this work. The short version: the old vehicles for industry–academia collaboration still create value, but the world got faster and they mostly did not. The fundamentals universities teach are still sound. What changed is the practice layer on top, and AI is compressing those cycles fast. We think the answer is not one big new program, but a portfolio of small bets that are cheap to start, quick to learn from, and flexible enough to evolve as the work does. The engagements in this report are early bets in that spirit.

The report begins with the Engagement Metrics Snapshot. This section pulls metrics from our Medium blog and LinkedIn activity, plus the engagements themselves. It gives a visual read on our outreach and impact, so you can track progress and effectiveness straight from the report.

Next comes a snapshot of the technologies we explored and applied across projects. It shows the range of tools and approaches we investigated, and gives a quick sense of the technical landscape shaping our work this semester.

After that, the report lays out a full table of all the engagements. Each row gives a glimpse of one engagement: the problem statement, the domain, the university or program, key numbers, and links to related assets. Those assets include blogs, videos, posters, and other materials, and they offer a deeper dive into each program.
We then present each engagement in greater detail later in the report, so you can see the challenges we are working to address.

This report is both a record and an invitation. It reflects our commitment to academic collaboration and reaches out to anyone interested in this work. Doing this well takes more than one company or one university. We are looking for partners in industry, in academia, and at the edges of both who want to think with us. The goal is shared understanding first, not a shared plan. The next step is not a launch. It is a conversation.

For further information, queries, or feedback, please feel free to contact Rajeev Chhajer at rajeev_chhajer@honda-ri.com and/or Ryan Lingo at ryan_lingo@honda-ri.com. We are eager to connect with you and explore opportunities for collaboration

Thank you for your interest in our work. We look forward to hearing from you. You can also connect with our community by checkout out our website, following us on Medium and LinkedIn.

ENGAGEMENT METRICS SNAPSHOT FOR SPRING 2026

Engagements

Student Engagement Projects: 15

University Partnerships: 7

Unique Programs: 11

Students Engaged With: 101

Medium

Number of Blogs: 16

Number of Blog Views: 782

Outcomes

Technologies Utilized/ Researched: 84

Prototypes / Demos: 36

Repos / Decks: 28

Videos: 12

LinkedIn

Number of LinkedIn Posts: 17

LinkedIn Organic Impressions: 14,030

LinkedIn Engagement Rate Avg: 6.21%

TECHNOLOGIES SNAPSHOT FOR SPRING 2026

List of the 84 technologies utilized/researched across Spring 2026 projects, highlighting the breadth of student engagement and technical exploration.

🧠 Intelligence + AI

- OpenAI API
- Qwen/Qwen2.5-1.5B-Instruct for local LLM execution
- Hugging Face Transformers
- Torch
- Accelerate
- BitsAndBytes
- Sentence Transformers
- all-MiniLM-L6-v2 for semantic embeddings
- Scikit-learn
- NetworkX
- UMAP
- HDBSCAN
- CodeRankEmbed
- LocAgent
- LSTM
- Temporal Graph Transformer
- MCP
- Graphiti
- Google Gemini
- Anthropic / Claude
- scikit-learn
- OpenAI / GPT-5 API
- PyTorch
- FAISS
- BM25
- OpenAI text-embedding-3-small
- PointNet
- LlamaIndex
- RepoGraph

☁️ Cloud / Data / Communication

- Docker
- SQLite
- Supabase
- Postgres / JSONB
- Neo4j
- Neo4j 5 Desktop
- Neo4j Graph Data Science
- Cypher
- Neo4j Bloom
- SerpAPI
- Excel
- Pandas
- NumPy

💻 Software

- Python
- Jupyter Notebooks
- Plotly
- JSON Repair
- tqdm
- Gradio
- PKL data conversion pipeline
- OpenCode
- FastAPI
- tree-sitter
- GitHub
- Opencode AI
- Langfuse
- Autodeck Fusion
- SimScale
- LangGraph StateGraph
- ReactFlow
- Recharts
- VS Code extensions
- Electron
- React
- TypeScript
- Git / GitHub
- PyMuPDF / PDF
- Jekyll
- HTML
- CSS
- JavaScript
- Matplotlib
- Seaborn
- Beautiful Soup / Requests
- AST parsing
- JSON
- Dagre auto-layout

📱 Hardware

- RealSense 99ACXA
- Raspberry Pi
- TI IWR6843 mmWave radar
- Unitree 4D L2 LiDAR
- 3D Printing / FDM

Spring 2026 University Engagements Summary Table Part 1

Innovation Prompt	University / Program	Key Numbers	Link(s)
How might we use AI and patent data to identify which generated product ideas are truly novel, valuable, and worth prioritizing?	Carnegie Mellon University — Master of Science in Business Analytics	4 students 19 technologies 14 weeks	Blog · LinkedIn · Deck Repo · Poster
How might we design an AI ideation system that measures semantic similarity, filters hidden redundancy, and produces a diverse portfolio of innovation concepts that teams can confidently evaluate and act on?	Carnegie Mellon University — Master of Science in Business Analytics	4 students 6 technologies 14 weeks	Blog · LinkedIn · Report Deck · Data Nexus Callout
How might we design safer, more intentional ways for drivers to communicate context and intent without increasing distraction, emotional friction, or conflict on the road?	Carnegie Mellon University — Master of Human-Computer Interaction	5 students 14 weeks	Blog · LinkedIn
How might we detect and model patterns of coordination, neutral activity, and disruption in shared indoor spaces using privacy-preserving motion data processed at the edge?	Ohio State — Electrical & Computer Engineering	3 students 6 technologies 28 weeks	Blog · LinkedIn Deck · Mid-Report Video
How might we design adaptive retrieval systems that help AI coding agents understand evolving codebases while reducing stale, irrelevant, or noisy context?	Ohio State — Computer Science and Engineering	4 students 10 technologies 14 weeks	Blog · LinkedIn Repo · Deck
How might we use AI agents to improve public transportation on Ohio State's campus?	Ohio State — Make OHI/O	40 students 2 days	Blog · LinkedIn Video · 1st Place
How might we improve the reliability of AI systems that learn and act from context?	Smith College — Data Science Clinic	6 students 3 technologies 14 weeks	Blog · Report LinkedIn · Deck
How might we design a compact, efficient, and low-noise vertical axis wind turbine that performs reliably in turbulent rooftop conditions and integrates seamlessly with urban architecture?	Smith College — Design Clinic	4 students 4 technologies 28 weeks	Blog 1 · Blog 2 · Report Linkedin · Deck

Spring 2026 University Engagements Summary Table Part 2

Innovation Prompt	University / Program	Key Numbers	Link(s)
How might AI reshape the way research teams identify, evaluate, and act on emerging innovation opportunities?	Likelion Student Club	6 students 7 technologies 14 weeks	Blog LinkedIn Demo
How might we help people build with AI without feeling overwhelmed by the technical details?	Likelion Student Club	6 students 8 technologies 14 weeks	Blog LinkedIn Deck
How might we evaluate LLM agents so we can distinguish genuine reasoning from the effects of context visibility?	Halıcıoğlu Data Science Institute (HDSI)	4 students 3 technologies 28 weeks	Blog LinkedIn Report Poster
How might we design scalable evaluation systems that make LLM-generated summaries more reliable, interpretable, and aligned with human judgment?	Halıcıoğlu Data Science Institute (HDSI)	4 students 6 technologies 28 weeks	Blog LinkedIn Report Poster
How might we measure the gap between what LLM judges reward and what humans find helpful in AI-generated explanations?	Halıcıoğlu Data Science Institute (HDSI)	4 students 11 technologies 28 weeks	Blog LinkedIn Report Poster
How might we help AI understand a codebase as a connected system rather than a collection of separate files?	Master of Science in Data Science	3 students 8 technologies 14 weeks	Blog Report Demo LinkedIn Deck
How might we help AI systems understand software the way engineers do: through relationships, dependencies, and change impact rather than isolated code snippets?	Master of Science in Computer Science	4 students 9 technologies 14 weeks	Blog Repo Report LinkedIn Deck Poster

Innovation Prompt

How might we use AI and patent data to identify which generated product ideas are truly novel, valuable, and worth prioritizing?

Description

This project developed an AI-driven framework for evaluating whether generated product ideas are truly novel, strategically valuable, and worth prioritizing. Rather than focusing on idea generation alone, it transformed public patent data into a structured evaluation pipeline that compares concepts by meaning, not just wording. The system used a standardized idea schema, semantic embeddings, clustering, and deterministic scoring to identify white-space opportunities and rank concepts by novelty and business value. The result is a practical decision framework that helps teams cut through repeated or surface-level ideas, surface differentiated opportunities, and prioritize innovation candidates with clearer evidence for further technical review.

Key Learnings

Novelty is not the same as new wording. LLMs can generate many ideas that sound different but reuse the same underlying concept, so evaluation has to compare ideas by structure and meaning rather than surface language.

White space needs evidence, not intuition. Patent embeddings, clustering, and similarity scoring make it possible to see where existing research is dense, where gaps exist, and whether a generated idea is actually distant from prior work.

AI can support prioritization, but judgment still matters. Semantic novelty and deterministic value scores help narrow the field, but final decisions still require technical validation, market context, and expert review before an idea becomes a real investment.

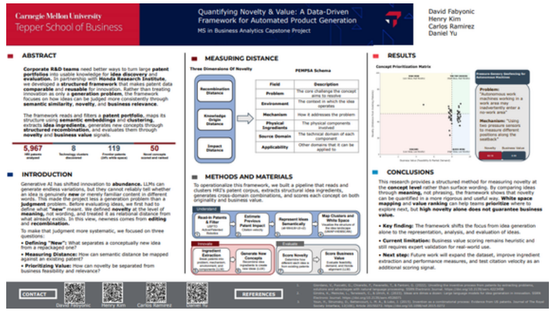
Possible Next Steps

Validate novelty against stronger evidence. Expand the comparison set beyond the initial patent corpus and test whether the same "white space" opportunities still appear when benchmarked against broader patents, publications, products, and competitor activity.

Calibrate value scoring with expert input. Replace or supplement heuristic business value scores with structured reviews from technical, market, and strategy experts, then use their judgments to tune the scoring logic and reduce false confidence.

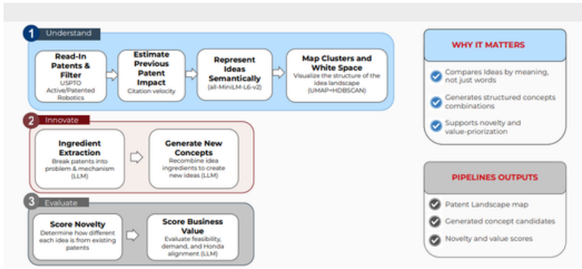
Turn the framework into a reusable research tool. Package the pipeline, scoring rubric, data schema, and visualization outputs into a repeatable workflow that other teams can apply to new domains, with clear guidance on inputs, assumptions, limits, and decision points.

Engagement Highlights

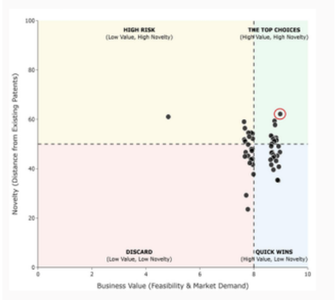


Project poster summarizing the framework, methodology, and results.

A framework overview showing how unstructured patent data is transformed into structured ideas, recombined concepts, and prioritized innovation opportunities.



An end-to-end pipeline showing how patent data is mapped, recombined, scored, and prioritized into innovation opportunities.



A prioritization matrix showing how generated concepts are ranked by novelty and business value to identify high-potential innovation opportunities.

Key Numbers: Duration: 14 weeks Number of students: 4 Technologies utilized / researched: 19

Links: Blog LinkedIn Repo Deck Poster

Innovation Prompt

How might we design an AI ideation system that measures semantic similarity, filters hidden redundancy, and produces a diverse portfolio of innovation concepts that teams can confidently evaluate and act on?

Carnegie Mellon University

MASTER OF SCIENCE IN BUSINESS ANALYTICS (CAPSTONE)

Description

This project developed Kaleidoscope, an AI-powered innovation intelligence engine for building diverse, non-redundant idea portfolios. It reframes AI brainstorming as a measurable selection process: ideas are broken into six semantic dimensions, scored for similarity, and filtered before entering the portfolio. The system combines idea generation, concept brief development, and early patent-risk screening into a repeatable workflow. Kaleidoscope helps teams reduce hidden redundancy, surface stronger concepts, and evaluate opportunities with clearer evidence. It was selected as one of three finalist capstone projects from 30 competing projects to present at Carnegie Mellon University's Tepper School of Business Data Nexus 2026.

Key Learnings

More ideas are not the same as more innovation. AI can generate large volumes of concepts quickly, but without semantic measurement, many of those ideas repeat the same underlying pattern in different words.

Diversity has to be enforced after generation, not just requested in the prompt. Kaleidoscope showed that structured selection, using similarity scoring across six semantic dimensions, is what turns raw AI output into a truly distinct portfolio.

Novelty becomes more useful when it is paired with evidence. By combining similarity scoring, concept brief development, and early patent-risk screening, the system helps teams move from brainstorming to clearer, more defensible innovation decisions.

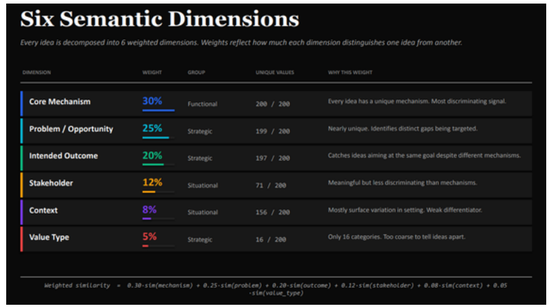
Possible Next Steps

Build adaptive stopping criteria. Instead of setting a fixed portfolio size, monitor when marginal novelty begins to fall and stop the run before the idea space saturates.

Add domain-tuned similarity models. Improve the system's ability to recognize meaningful differences in specialized fields by testing embeddings trained or calibrated on domain-specific research, patents, and strategy documents.

Create a human review layer for portfolio shaping. Let experts adjust strategic priorities, mark must-keep ideas, and refine dimension weights so the final portfolio reflects both semantic diversity and real-world decision needs.

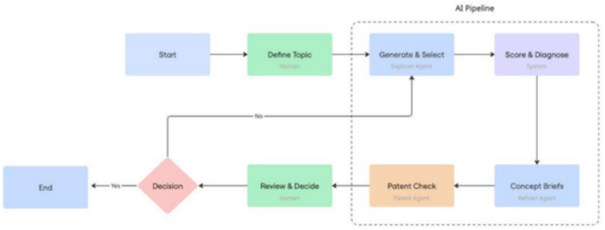
Engagement Highlights



Each idea is scored across six weighted dimensions. The weights help Kaleidoscope spot real overlap, not just matching words.

The rules push the system to explore beyond obvious idea patterns. A semantic safety check rejects concepts that are too similar to ideas already accepted.

The Kaleidoscope team was selected as one of three finalist teams to present at Carnegie Mellon University's Tepper School of Business Data Nexus 2026, chosen from 30 competing capstone projects.



Kaleidoscope moves from topic definition to AI generation, scoring, concept briefs, and patent checks. Human review stays in the loop before ideas are accepted or sent back for another iteration.

Key Numbers: Duration: 14 weeks Number of students: 4 Technologies utilized / researched: 6

Links: Blog LinkedIn Deck Data Nexus Callout Report

Innovation Prompt

How might we design safer, more intentional ways for drivers to communicate context and intent without increasing distraction, emotional friction, or conflict on the road?

Carnegie Mellon University

MASTER IN HUMAN COMPUTER INTERACTION (CAPSTONE)

Description

The project explores how drivers communicate on the road, moving beyond the limits of the horn, flashing lights, and ambiguous gestures. The five-student team studied road rage, intent, empathy, trust, and attention through driver research, assumption testing, and early design probes. Their work reframed the challenge from preventing road rage to designing mediated, low-friction systems that help drivers understand context and coordinate safely without adding distraction or escalation. The project will continue into the summer as the team moves from research insights into concepts, prototypes, and evaluation.

Key Learnings

Road rage is not a single, stable design target. Drivers define it differently, and the same behavior can be read as normal, aggressive, or enraging depending on the person and situation.

More communication is not automatically better. Direct driver-to-driver messaging could create distraction, distrust, or escalation. The stronger opportunity is mediated, constrained communication that clarifies intent without opening the door to conflict.

Context helps only when it is useful and actionable. Drivers may feel more empathy when they understand why something happened, but they only want information that helps them make a safer decision, like braking, yielding, or creating space.

Possible Next Steps

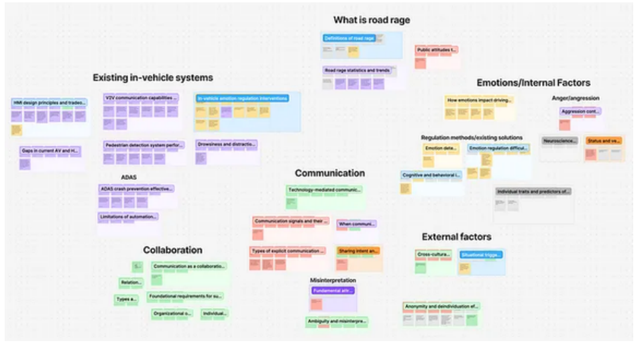
Move from research into concepts, prototypes, and evaluation. The team can turn early insights into more focused design directions.

Test assumptions more directly. This could include subject matter expert conversations, analogous research, and higher-fidelity simulated driving experiences.

Refine mediated communication ideas. Early findings point away from open driver-to-driver messaging and toward quieter, ambient systems where the vehicle surfaces useful context only when it supports safer action.

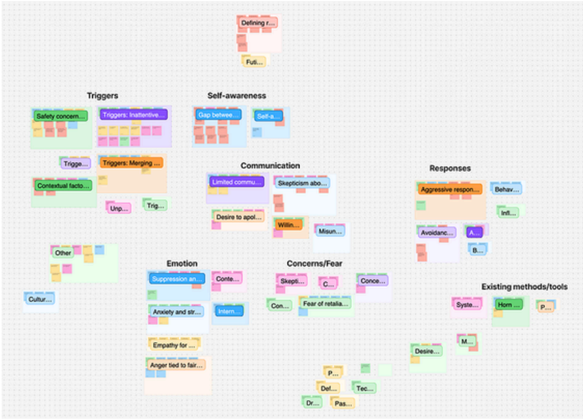
Engagement Highlights

The team builds a shared understanding of the automotive landscape through early research, mapping, and discussion.



The team organized early research into themes spanning road rage, vehicle systems, communication, collaboration, emotions, and external driving factors.

A kickoff workshop helped the team clarify priorities, define shared measures of success, and align around seven guiding words: safety, well-being, communication, intentionality, empathy, scalability, and norms.



The team synthesized research findings into themes around triggers, emotion, communication, self-awareness, and driver responses.

Key Numbers: Duration: 14 weeks Number of students: 5

Links: Blog LinkedIn

Innovation Prompt

How might we detect and model patterns of coordination, neutral activity, and disruption in shared indoor spaces using privacy-preserving motion data processed at the edge?

Description

This project continued in the spring by moving from system design to a working prototype. The team tested the original mmWave radar approach, then pivoted to the Unitree 4D L2 LiDAR after finding that radar data were not dense enough for reliable multi-person detection. The final system used LiDAR sensing, a conversion pipeline, encrypted local storage, two machine learning layers, and a post-session interface. PointNet and LSTM classified individual actions, while a Temporal Graph Transformer classified group states such as coordination, neutral behavior, and disruption. By the end of the semester, the prototype captured spatial motion data, processed it through both model layers, stored it securely, and presented results through an interface. The main limitation was neutral behavior classification, which future work would address through better data, features, coverage, and real-time processing.

Key Learnings

Privacy shaped the goal, but data quality shaped the design. The team started with mmWave radar because it supported privacy-preserving sensing, but testing showed the data were not dense enough for reliable group detection. That led to the pivot to LiDAR.

Preprocessing became core to the system. Raw sensor data were not ready for modeling. The team had to build a conversion pipeline that turned LiDAR sessions into structured PKL files the models could use.

Group behavior is harder than individual action. Actions like walking and waving have clearer motion patterns. Group states, especially neutral behavior, were harder to separate and need better data, features, and coverage.

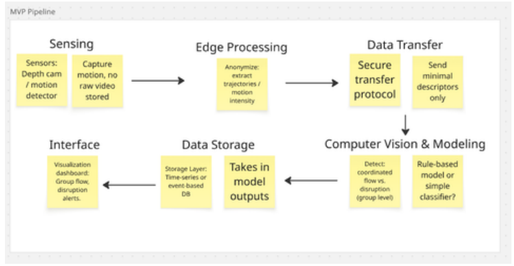
Possible Next Steps

Expand and balance the dataset.
Collect more examples of coordination, neutral behavior, and disruption, especially neutral cases, so the model can better separate similar group states.

Improve group-level classification.
Refine frame selection, feature engineering, and Temporal Graph Transformer inputs to improve confidence and reduce confusion between neutral behavior and coordination.

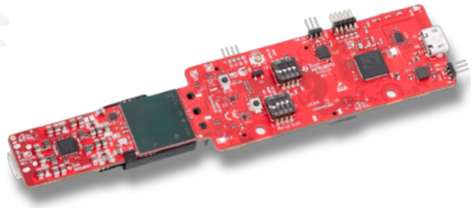
Move closer to real-time deployment.
Extend the post-session prototype toward live processing and display, while improving LiDAR coverage to reduce occlusion and support more reliable multi-person tracking.

Engagement Highlights



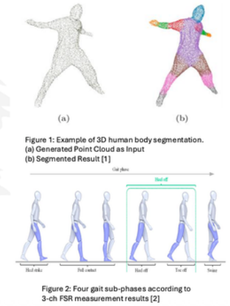
MVP pipeline diagram showed the end-to-end flow from sensing and edge anonymization through secure transfer, modeling, storage, and the dashboard. It reduced implementation risk by clarifying required components and handoffs.

Team photo from the final project presentation, featuring the student team and mentors after sharing the Edge-Based Spatial Rhythm prototype and project outcomes.



Type: IWR6843

Hardware selection slide showed the TI IWR6843 mmWave radar chosen as the primary sensor for privacy-preserving motion capture. It de-risked the build by locking in a sensor capable of producing the point-cloud data needed for downstream modeling.



Layer 2 software concept slide outlined a PointNet+LSTM approach to classify collaborative movements (walking, sitting, leaning) from radar point-cloud sequences. It provided a concrete modeling plan that linked raw sensor data to the higher-level behaviors needed for coordination detection.

Innovation Prompts

How might we design adaptive retrieval systems that help AI coding agents understand evolving codebases while reducing stale, irrelevant, or noisy context?

OSU COMPUTER SCIENCE AND ENGINEERING (CAPSTONE)

Description

REBOOT is a capstone research project that explored how AI coding agents can retrieve better context from large, evolving codebases. The team built an MCP-compatible adaptive retrieval prototype using tree-sitter, Graphiti, Neo4j, Python, FastAPI, and SQLite to parse code, create knowledge graph-based context, rank search results, collect feedback, and explain retrieval decisions. Through an evaluation harness based on real software issues, the team found that a simple exploration baseline using tools like grep and file reading outperformed the knowledge graph approach in this setting. The project's main contribution was a working prototype and a clear research finding: agentic code retrieval systems need early baseline evaluation, carefully chosen code relationships, and practical search behavior to avoid noisy or stale context.

Key Learnings

Simple baselines matter. The project showed that familiar code search tools like grep and file reading can outperform more complex graph-based retrieval, making early baseline testing essential before investing in advanced architectures.

Code relationships need focus. Knowledge graphs can capture many connections across a codebase, but not every connection helps an agent solve a task. Useful retrieval depends on surfacing the right structures, such as functions, files, dependencies, and implementation paths.

Adaptation must stay lightweight. Feedback signals and confidence scoring can help retrieval improve over time, but large pre-ingestion steps can slow developer workflows. Future systems need to improve context quality without adding too much setup or maintenance.

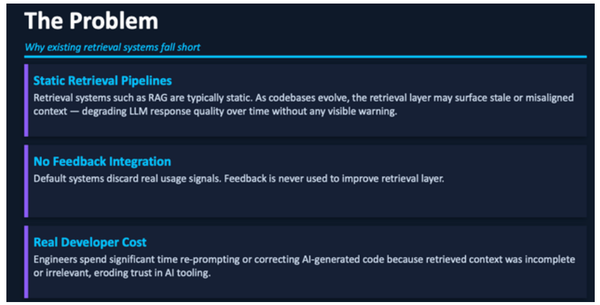
Possible Next Steps

Build a focused code-retrieval benchmark.
Create a small, human-labeled evaluation set with real codebase questions, expected files, and relevant symbols to compare graph-based retrieval against grep, file search, and agent exploration.

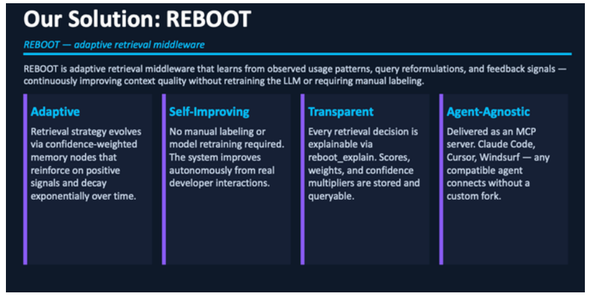
Redesign the graph around high-value code structures.
Focus ingestion on relationships that matter most for coding tasks, such as function calls, imports, file ownership, dependencies, and implementation paths, while filtering out low-value or noisy graph connections.

Prototype a lightweight adaptive retrieval loop.
Test feedback signals, confidence scoring, and result explanations in a narrow end-to-end workflow to see whether adaptation improves retrieval quality without slowing down developer workflows.

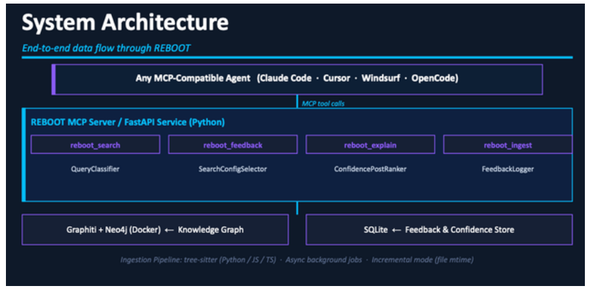
Engagement Highlights



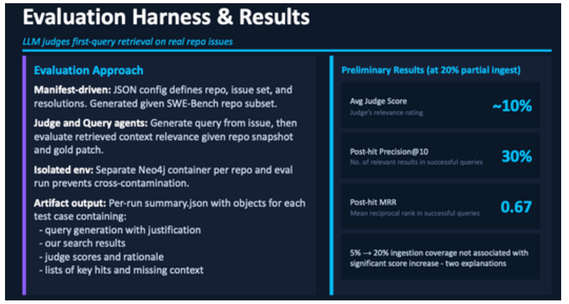
Frames the core challenge: static retrieval systems can return stale or incomplete code context, creating extra work for developers using AI tools.



Explains REBOOT as adaptive retrieval middleware that learns from feedback, explains ranking decisions, and works across compatible AI coding agents.



Shows how REBOOT connects MCP-compatible coding agents to its FastAPI middleware, graph database, feedback store, and code ingestion pipeline.



Summarizes how the team tested retrieval quality using real repository issues, LLM judges, isolated eval environments, and early retrieval metrics.

Key Numbers: Duration: 14 weeks Number of students: 4 Technologies utilized / researched: 10

Links: Blog Linkedin Repo Deck

Innovation Prompts

How might we use AI agents to improve public transportation on Ohio State's campus?

OSU MAKE OHI/O (HACKATHON)

Description

At MakeOHI/O 2026, we served as industry mentors and challenged student teams at The Ohio State University to explore how AI agents could improve campus public transportation. The students built hardware-software prototypes that addressed real rider and operator needs, including accessible bus stop decision support, live seat-capacity tracking, demand monitoring, and smarter route optimization. For 99P Labs, the event was a meaningful opportunity to support emerging builders, share our perspective on applied AI and mobility, and see how student teams translate complex transportation challenges into tangible, human-centered prototypes.

Key Learnings

AI agents became tangible.
Teams embedded AI into real transit touchpoints, including kiosks, dashboards, sensors, cameras, and stop-level recommendation tools.

Accessibility made ideas stronger.
The best projects focused on clear rider needs, such as deciding whether to wait, walk, or choose another service when transit information is hard to access.

Real-time mobility needs full-system thinking.
Teams stood out when they connected live data, hardware inputs, user communication, and operational decisions into one adaptive loop.

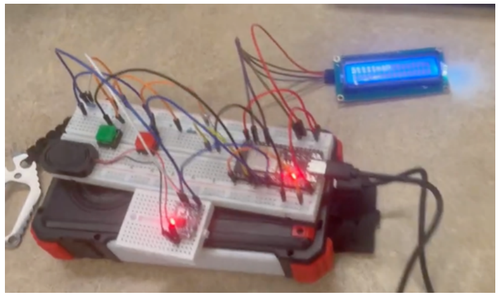
Possible Next Steps

Launch a rapid follow-on research sprint.
Offer a short, structured program after the event where promising teams receive mentorship, refine their prototypes, and test whether their concepts can produce useful mobility insights.

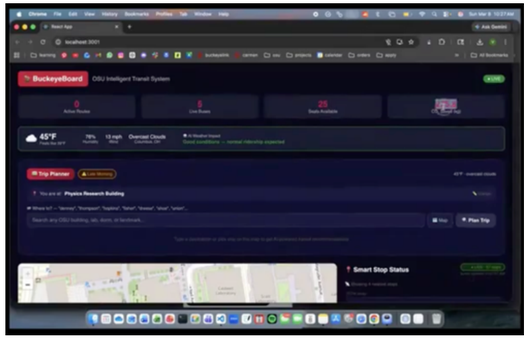
Identify pilot-ready campus transit concepts.
Review the strongest ideas, especially around accessibility, live capacity tracking, and demand-aware routing, to see which could be tested with real data, riders, or campus transit partners.

Create a repeatable challenge framework.
Package the prompt, design principles, sample data sources, and evaluation criteria so future student teams can build on this work and produce more comparable results.

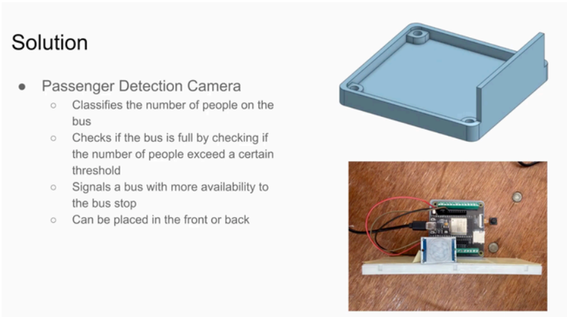
Engagement Highlights



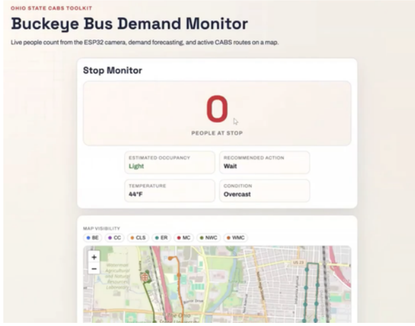
Aaron Zhou's first-place hardware prototype helped riders decide whether to wait for the next bus, walk to another stop, or use an on-demand service by combining bus data, weather, walking distance, and ambient light into one clear recommendation.



BuckeyeBoard's software dashboard helped riders plan campus trips by combining live bus locations, seat availability, weather, crowd levels, and estimated travel options into one AI-assisted mobility recommendation.



Team ERVA's smart campus bus routing system used an ESP32-CAM and computer vision to estimate bus occupancy in real time, helping operators identify overcrowded routes and reroute buses where demand is highest.



Buckeye Bus Demand Monitor used an ESP32 camera, motion sensor, YOLO, and Llama to estimate demand at a bus stop in real time and recommend whether riders should board the arriving bus or wait for the next one.

Key Numbers: Duration: 2 days | Number of students: 40 | Number of teams: 10

Links: Blog | LinkedIn | Video | 1st Place

Innovation Prompt

How might we improve the reliability of AI systems that learn and act from context?

Description

This research project examined how large language models learn and fail when adapting to new tasks through context alone. The study evaluated model behavior across turn-taking, rule inference, ambiguity detection, and relational reasoning tasks, with a focus on the structure of model failures rather than benchmark performance alone. Across experiments, the findings suggest that context strongly shapes model behavior, but models often default to plausible, low-complexity answers when information is incomplete or ambiguous. The project points toward future research on context-level "teaching patches" that improve reliability without retraining the model.

Key Learnings

LLMs often choose a plausible rule instead of recognizing uncertainty. In ambiguous tasks, models frequently committed to simple answers even when the examples did not support one clear rule.

Model confidence does not reliably track correctness. Even when performance dropped on complex prompts, models often responded with certainty.

Context-level "teaching patches" offer a practical path to better reliability. Because failures were systematic, prompt structures like rule comparison, turn counters, relationship hierarchies, and uncertainty checks may improve behavior without retraining.

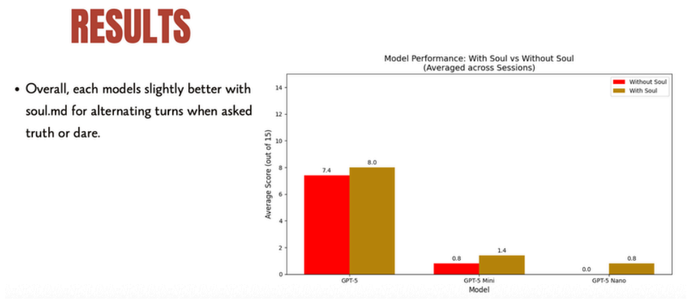
Possible Next Steps

Develop teaching patches that help models recognize uncertainty. Testing prompt structures that require rule comparison or explicit abstention could reduce guessing when examples are ambiguous.

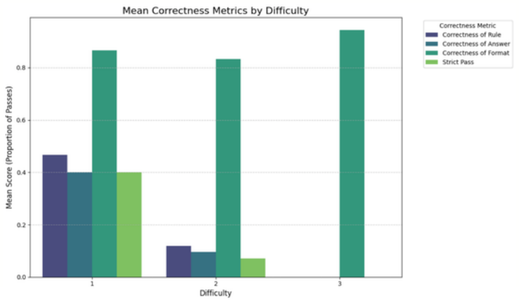
Expand evaluation across more models, tasks, and trial sizes. Running the same experiments at larger scale would show which failure patterns generalize and which are tied to specific task designs.

Build stronger context structures for multi-step reasoning. Organizing rules, relationships, and state information more clearly could improve model performance on turn-taking, relational reasoning, and conflicting constraints.

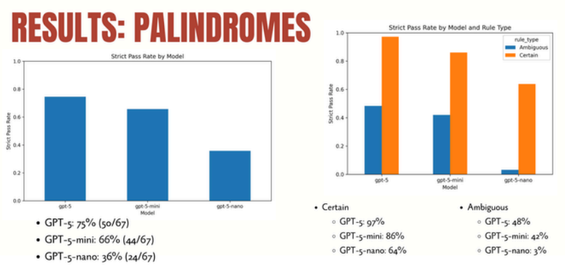
Engagement Highlights



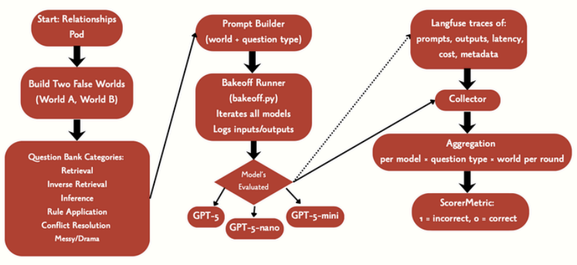
Adding a structured persona file slightly improved turn-based performance, but models still struggled to maintain sequence and track whose turn it was across multiple exchanges.



Model performance dropped sharply as character-level transformation rules became more complex, highlighting the limits of learning precise rules from examples alone.



Models performed well when the reversal rule was clear, but struggled to abstain when examples supported more than one possible rule.



Relationships experiment workflow, from fictional worlds and question banks to model testing, Langfuse logging, and result aggregation.

Key Numbers: Duration: 14 weeks Number of students: 6 Technologies utilized / researched: 3

Links: Blog LinkedIn Report Deck

Innovation Prompt

How might we design a compact, efficient, and low-noise vertical axis wind turbine that performs reliably in turbulent rooftop conditions and integrates seamlessly with urban architecture?

Description

VAWT Ventures explored the design of a vertical axis wind turbine for rooftop renewable energy generation in low-speed, turbulent urban wind conditions. The team researched 224 turbine concepts, narrowed them to four prototype directions, and used CAD modeling, 3D printing, wind tunnel testing, rooftop wind data, and SimScale CFD to evaluate performance. The final concept was a helical hybrid VAWT combining an outer Darrieus rotor with an internal Savonius rotor. While the design is not yet viable for low-wind rooftop deployment, the project created a strong foundation for future turbine optimization, rooftop testing, and digital twin development.

Key Learnings

Urban wind is difficult to harness. Rooftop wind conditions are low-speed, turbulent, and multidirectional, making reliable power generation more challenging than expected.

Hybrid designs show promise but need more optimization. The selected helical hybrid VAWT improved startup potential by combining Savonius and Darrieus rotor concepts, but its power output remained too low for practical rooftop deployment.

Physical testing and simulation must be developed together. Wind tunnel testing, rooftop wind data, and SimScale CFD each provided useful insights, but the project showed that CFD alone cannot predict real rooftop performance without stronger validation.

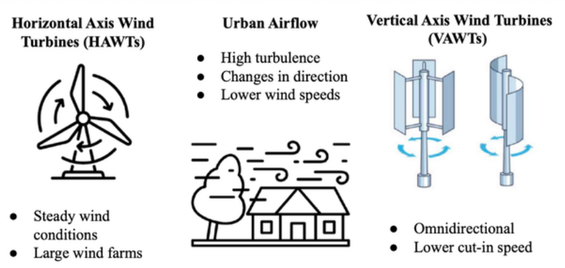
Possible Next Steps

Optimize for low-wind rooftop conditions. Continue refining the helical hybrid design through improved wind tunnel testing, torque measurement, and small-scale rooftop trials.

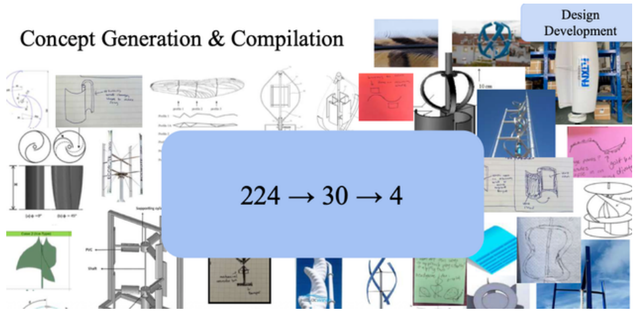
Explore higher-wind applications. Test whether a larger or redesigned VAWT could perform better in locations with stronger, more consistent wind, such as taller urban buildings.

Advance toward a digital twin. Build a higher-fidelity simulation model that can be validated against physical testing and used to predict turbine behavior more accurately.

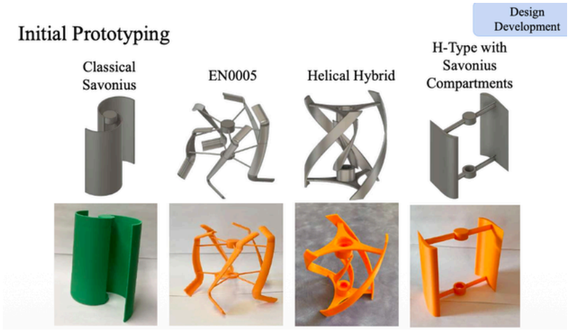
Engagement Highlights



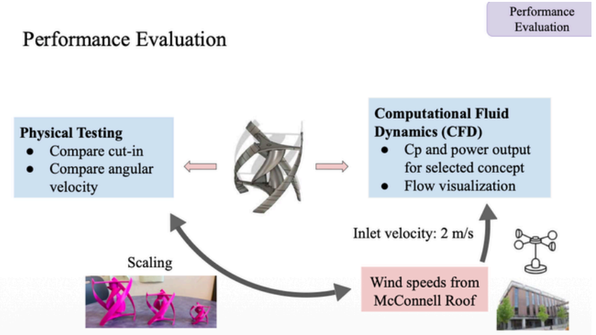
Urban rooftops create low-speed, turbulent, multidirectional wind conditions, making vertical axis wind turbines a promising format for distributed renewable energy.



The team explored 224 turbine concepts, narrowed the field to 30 viable directions, and selected four designs for prototyping and testing.



Four VAWT concepts were modeled in CAD and 3D printed to compare their performance in controlled wind tunnel tests.



The selected turbine concept was evaluated through wind tunnel testing, rooftop wind data, SimScale CFD, and scaling studies to understand performance and future design needs.

Key Numbers: Duration: 28 weeks Number of students: 4 Technologies utilized / researched: 4

Links: Blog 1 Blog 2 Linkedin Report Deck

Innovation Prompt

How might AI reshape the way research teams identify, evaluate, and act on emerging innovation opportunities?

UNIVERSITY OF CALIFORNIA, BERKELEY (CAPSTONE)

Description

AURA is an AI-augmented research platform that helps teams turn complex research landscapes into clearer directions for innovation. Rather than stopping at a traditional summary of what is already known, AURA generates evidence-based hypotheses that can be evaluated, compared, and refined. The system uses domain-aware workflows to account for the technical, market, regulatory, and operational factors that shape emerging technology decisions. Through structured scoring and iterative ranking, AURA helps surface stronger opportunities, clarify tradeoffs, and identify where further exploration is most valuable. It supports a research process focused not only on understanding a topic, but on deciding what may be worth testing, investing in, or developing next.

Key Learnings

Research synthesis is only the starting point. The project showed that innovation teams need more than summaries of what is already known. Useful research should help teams form clear, testable directions for what to explore next.

Domain context shapes better hypotheses. AURA showed that the same topic can lead to different outputs depending on the lens applied. Technical, market, regulatory, and operational factors all influence which ideas become most relevant.

Evaluation must be transparent and actionable. AI-generated hypotheses are more useful when teams can see how ideas are scored, compared, and refined. Visible tradeoffs help turn research into decisions teams can trust and act on.

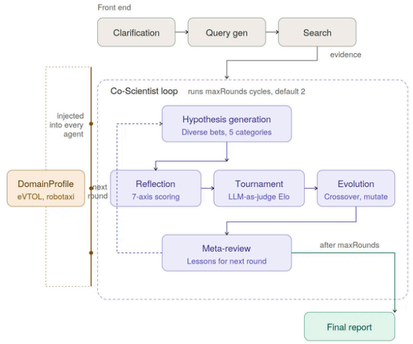
Possible Next Steps

Expand domain testing.
Test AURA across new research areas beyond mobility, such as medical devices, energy storage, or sustainability, to see how well the domain-aware framework adapts.

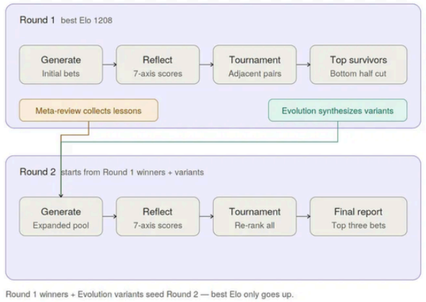
Improve long-term evaluation.
Build stronger ways to track how hypotheses evolve across research rounds, including clearer metrics for progress, confidence, and evidence quality.

Compare outputs across domains.
Run the same research question through multiple domain profiles to understand how different lenses change the hypotheses, rankings, and strategic recommendations.

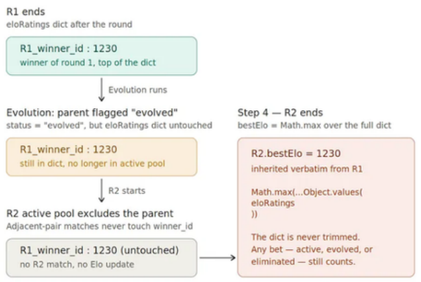
Engagement Highlights



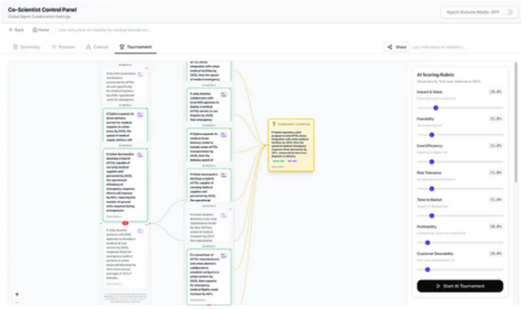
AURA's domain-aware research workflow moves from clarification and evidence gathering into an iterative AI co-scientist loop for generating, scoring, ranking, and refining innovation hypotheses.



The tournament process carries forward top hypotheses, synthesizes stronger variants, and re-ranks the expanded pool to produce a final set of innovation bets.



An untrimmed ratings dictionary allowed an inactive Round 1 winner to carry its Elo score into Round 2, revealing a gap in how tournament progress was measured.



The tournament UI shows how hypotheses move through comparison, scoring, and champion selection, making the evaluation process easier to inspect and understand.

Key Numbers: Duration: 14 weeks Number of students: 6 Technologies utilized / researched: 7

Links: Blog LinkedIn Demo

Innovation Prompt

How might we help people build with AI without feeling overwhelmed by the technical details?

UNIVERSITY OF CALIFORNIA, BERKELEY (CAPSTONE)

Description

This research project explored how AI-assisted coding changes the needs of new and emerging developers after code has been generated. The team first investigated codebase understanding through Code Compass, a VS Code extension for repository questions, code threads, and usage-style insights. As the research progressed, the focus shifted from explaining code to helping people safely manage it. The resulting prototype, VIVA, examined how visual workflows, safety guardrails, and targeted AI support could make version control more approachable for users who may not be comfortable with Git or the terminal. Through features such as save assistance, natural language undo, file insight, and conflict guidance, the project studied how developer tools can help people understand, protect, and evolve the work they create with AI.

Key Learnings

AI helps people create code, but they still need help managing it.
The project showed that new builders need support after generation, especially with saving, reviewing, recovering, and sharing their work.

Beginner tools should prioritize safety over full automation.
VIVA kept important decisions visible so users could understand risks before making changes.

Visual workflows can make technical systems easier to understand and trust.
By turning Git actions into clearer steps, the project made version control feel more approachable without removing its power.

Possible Next Steps

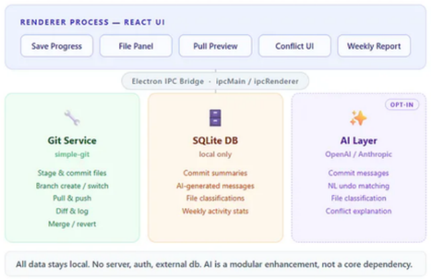
Study real AI-assisted coding workflows.
Observe how new builders save, edit, undo, and share code after using AI tools.

Test which safety patterns build trust.
Compare previews, confirmations, plain-language labels, and recovery flows to see what helps users feel more in control.

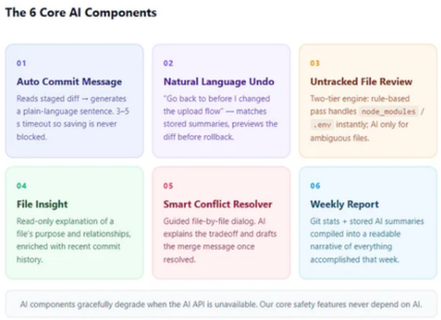
Explore version control beyond the desktop.
Research how visual Git workflows could support people building with AI on web, tablet, or mobile devices.

Engagement Highlights

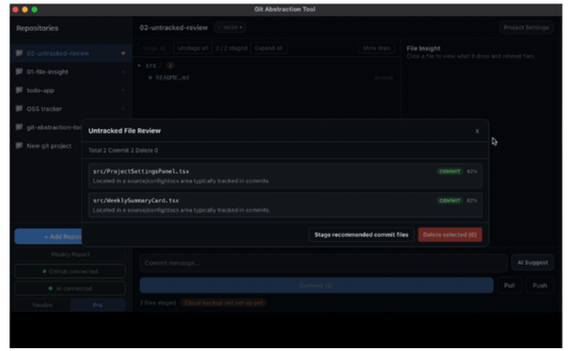
This timeline shows the project's shift from Code Compass, a codebase knowledge tool, to VIVA, a safety-first prototype for helping AI-assisted builders manage their code.



This architecture diagram shows how VIVA combines a React interface, local Git operations, SQLite storage, and optional AI features while keeping project data local by default.



This overview highlights the six AI features tested in VIVA, each designed to support a specific version control task without making AI responsible for core safety decisions.



This interface shows VIVA's natural language undo workflow, where users can describe the change they want to reverse instead of searching through commit hashes or using Git commands.

Key Numbers: Duration: 14 weeks Number of students: 6 Technologies utilized / researched: 8

Links: Blog LinkedIn Deck

Innovation Prompt

How might we evaluate LLM agents so we can distinguish genuine reasoning from the effects of context visibility?

Description

ContextEval examines how the information given to an LLM agent shapes its behavior during machine learning experimentation. The project asks whether apparent performance gains come from reasoning or from the context the agent can see. By holding the model, task, prompt structure, and optimization loop fixed while varying access to task descriptions, metrics, parameter bounds, and feedback history, the study isolates context visibility as an experimental variable. Across multiple benchmarks, the findings show that LLM agents behave more like corrective heuristics than true optimizers: they improve weak starting configurations quickly, but struggle to refine strong ones and do not consistently outperform random search.

Key Learnings

Context visibility must be controlled.
Agent performance can change based on what information the LLM sees, not just how well it reasons.

LLM agents act more like corrective heuristics than true optimizers.
They improve weak configurations quickly, but struggle to refine strong ones.

More information is not always better.
Longer feedback histories can hurt performance, while parameter bounds reduce invalid proposals without always improving outcomes.

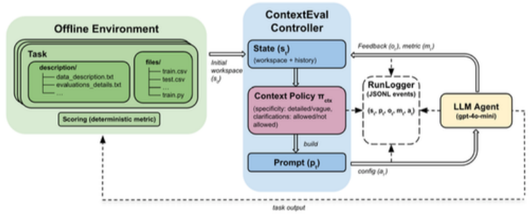
Possible Next Steps

Test stronger models and longer horizons.
Evaluate whether larger or reasoning-focused LLMs can move beyond early correction and show stronger iterative optimization.

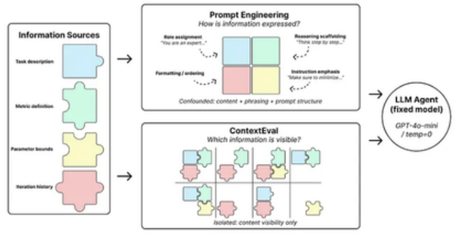
Explore adaptive context policies.
Study whether agents perform better when context changes over time, such as showing more history only after performance improves.

Compare against stronger optimization baselines.
Benchmark LLM agents against Bayesian optimization, TPE, and other sequential search methods under the same evaluation budget.

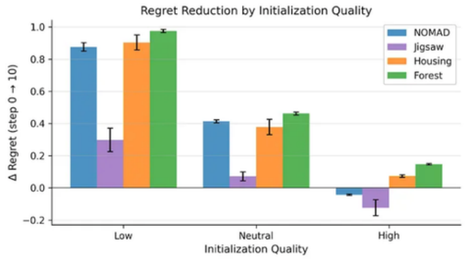
Engagement Highlights



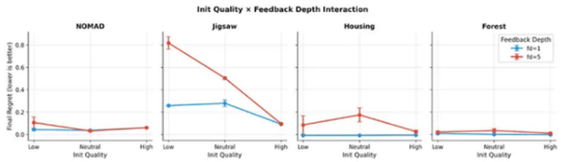
ContextEval architecture showing an offline ML environment that scores agent outputs, a controller that applies a configurable context policy to build prompts for a fixed LLM agent, and a JSONL run logger that records state, actions, and feedback for analysis.



ContextEval isolates information visibility as the sole experimental variable, in contrast to prompt engineering approaches where content and phrasing are confounded.



Starting conditions dominate agent performance: LLMs recover well from poor configurations but show limited ability to refine strong ones.



Longer feedback histories can hurt performance by anchoring the agent to prior poor results, especially when the initial configuration is weak.

Innovation Prompt

How might we design scalable evaluation systems that make LLM-generated summaries more reliable, interpretable, and aligned with human judgment?

Description

This project investigates how LLM-generated lecture summaries can be evaluated more reliably at scale. The team built an iterative LLM-as-judge framework that combines rubric-based scoring with deterministic signals such as section coverage, glossary recall, length error, and suspected hallucination rate. Rather than treating evaluation as a one-time score, the system uses judge feedback to refine summaries across multiple rounds, applies domain-aware rubrics based on lecture content, and selects the strongest recent output when improvement plateaus. From a research perspective, the work explores how hybrid evaluation methods can make AI-generated educational content more interpretable, reproducible, and better aligned with human expectations.

Key Learnings

Hybrid evaluation added clarity. LLM judging became more useful when paired with measurable signals like coverage, recall, and hallucination risk.

Iteration improved consistency. The judge-refine loop helped summaries get stronger while avoiding overreliance on the final rewrite.

Domain context mattered. The project showed that evaluation criteria should shift based on the subject area and type of content.

Possible Next Steps

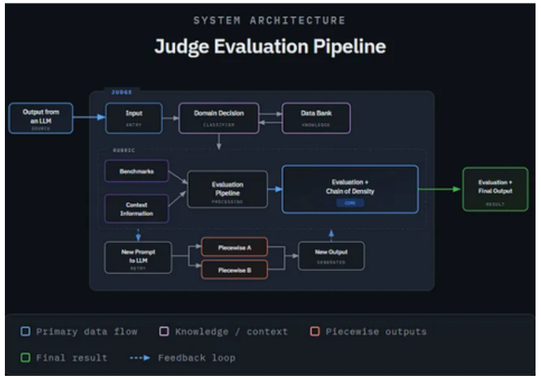
Test across more datasets. Expand beyond seven lectures to see how the framework performs across more subjects, formats, and levels of complexity.

Compare against human judgment. Validate the scoring system with student, instructor, or expert ratings to see how well it reflects real-world quality.

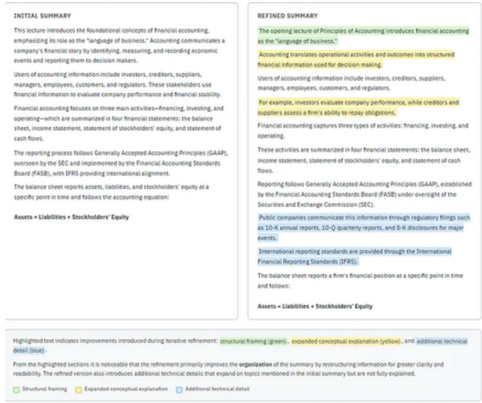
Improve grounding checks. Move beyond lexical hallucination signals toward claim-level verification, so the system can better distinguish faithful paraphrase from unsupported content.

Engagement Highlights

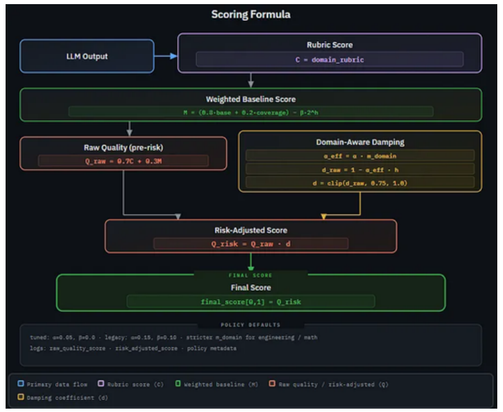
Project team at the poster session, sharing their work on iterative refinement for LLM evaluation.



System architecture for the iterative LLM-as-judge pipeline, showing how outputs move through domain routing, rubric evaluation, refinement, and final selection.



Refinement example demonstrating how judge feedback helps improve summary coverage, clarity, and completeness across iterations.



Hybrid scoring methodology that blends LLM judge scores, structural coverage signals, and domain-aware hallucination damping.

Key Numbers: Duration: 28 weeks Number of students: 4 Technologies utilized / researched: 7

Links: Blog Linkedin Report Poster

Innovation Prompt

How might we measure the gap between what LLM judges reward and what humans find helpful in AI-generated explanations?

Description

This research project examines how prompt design shapes the quality of LLM-generated explanations and how closely automated LLM judges align with human preferences. Using technical concepts across AI, computer science, and statistics, the team tested multiple prompt styles, evaluated outputs through both human judgment and LLM-as-Judge pairwise comparisons, and used Elo-based ranking to compare performance. The findings show that conversational, structured prompts tend to produce clearer explanations than formal academic prompts, while human and LLM judges often reward different qualities. This highlights the need for evaluation methods that account for human understanding, not just automated scoring.

Key Learnings

Prompt design is a major research variable. Small changes in instruction style produced measurable differences in explanation quality, showing that prompt framing should be treated as part of the experimental design, not just a setup detail.

Human and LLM judges reward different qualities. LLM-as-Judge evaluation aligned with human preferences at a broad level, but humans tended to value structure, completeness, and readability more than concise fluency alone.

Iterative evaluation can improve explanation quality. Using evaluation feedback to refine prompts led to stronger results, suggesting that prompt optimization can be studied as a repeatable research process rather than a one-off tuning exercise.

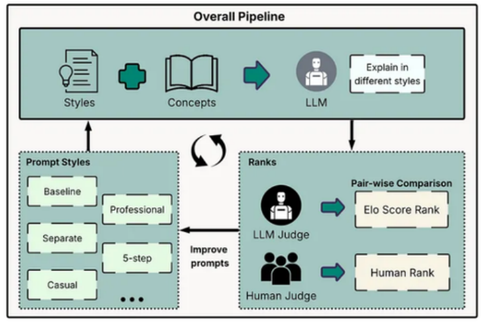
Possible Next Steps

Expand human evaluation. Test the framework with a larger and more diverse group of human evaluators to better understand how preferences vary by background, expertise, and learning goals.

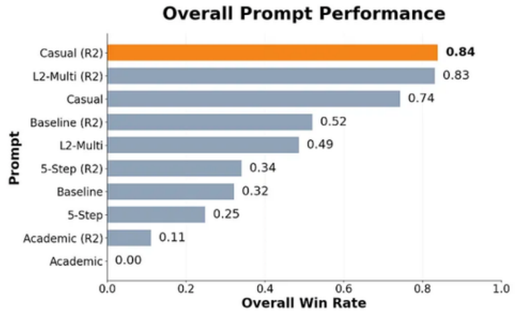
Compare multiple model families. Run the same prompt and evaluation pipeline across different generation and judge models to see whether the findings generalize beyond a single LLM setup.

Measure learning outcomes directly. Add comprehension quizzes, recall tasks, or user studies to test whether higher-ranked explanations actually help people understand and retain complex concepts.

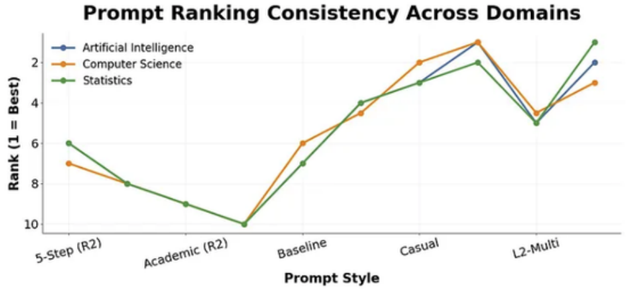
Engagement Highlights



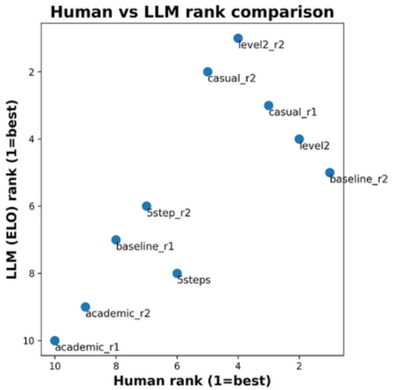
A diagram of the structured evaluation pipeline for testing how different prompt styles shape LLM-generated explanations.



Conversational and structured prompts achieved the highest win rates, while formal academic prompts performed the weakest across domains.



Prompt rankings remained mostly consistent across AI, computer science, and statistics, suggesting that effective prompt styles generalize across domains.



Human and LLM rankings broadly aligned, but key differences show that each judge rewarded explanation quality in different ways.

Key Numbers: Duration: 28 weeks Number of students: 4 Technologies utilized / researched: 11

Links: Blog Linkedin Report Poster

Innovation Prompt

How might we help AI understand a codebase as a connected system rather than a collection of separate files?

UNIVERSITY OF COLORADO BOULDER (CAPSTONE)

Description

This capstone project investigated how knowledge graphs can help software teams understand the impact of code changes in complex engineering systems. The research focused on connecting scattered context across code structure, change history, test behavior, and system dependencies into a unified model that supports more informed testing decisions. By representing software components and their relationships as a graph, the project explored how teams can move from intuition-based validation toward explainable, impact-aware prioritization. Its research value lies in showing how applied AI and knowledge systems can make software change more visible, traceable, and actionable, while creating a foundation for safer and more efficient engineering workflows.

Key Learnings

Connected context improves decision-making.
Software change is easier to evaluate when code structure, change history, tests, and dependencies are modeled together instead of treated as separate sources of information.

Explainability matters as much as automation.
For complex engineering systems, a useful AI-supported workflow must show why a recommendation was made, not just produce a ranked output.

Graph-based models can make hidden impact visible.
Knowledge graphs are well suited to tracing relationships across software systems, helping teams see downstream effects that are hard to capture through manual review or intuition alone.

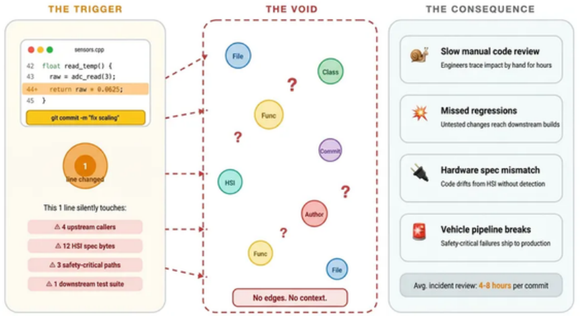
Possible Next Steps

Validate on real-world repositories.
Apply the approach to larger, production-like codebases to study how well the graph model, scoring logic, and dependency tracing scale beyond synthetic data.

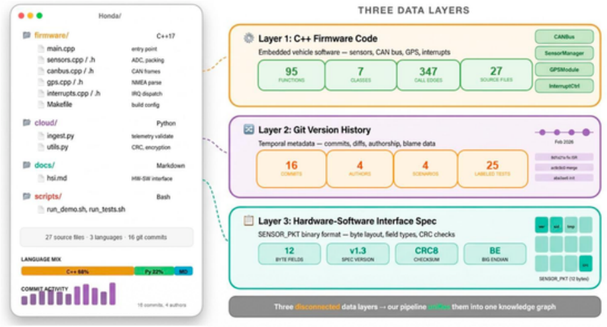
Integrate with development workflows.
Explore how impact-aware test prioritization could plug into code review, CI/CD, or release planning so teams receive recommendations at the moment a change is made.

Improve adaptive intelligence.
Use historical test outcomes and engineering feedback to refine the scoring model over time, while preserving explainability and traceability for high-stakes decisions.

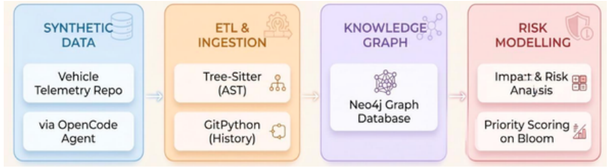
Engagement Highlights



The project's problem statement is that small code changes can create hidden impacts across tests, hardware specs, and safety-critical paths.



This image shows the three disconnected data layers the project unified: firmware code, version history, and hardware-software interface specs.



This image shows the project pipeline, from synthetic repository data to graph ingestion, knowledge graph construction, and risk-based test prioritization.

This image summarizes the project's results, showing that the system reduced the test set from 25 total tests to 4 to 7 recommended tests across evaluated scenarios.

Key Numbers: Duration: 14 weeks Number of students: 3 Technologies utilized / researched: 8

Links: Blog LinkedIn Report Deck Demo

Innovation Prompt

How might we help AI systems understand software the way engineers do: through relationships, dependencies, and change impact rather than isolated code snippets?

Description

Hindsight is a research project exploring how AI systems can better understand large software repositories by using code structure, not just text similarity. The project studies whether graph-augmented retrieval can improve repository-level question answering and code localization by capturing relationships such as function calls, imports, dependencies, and cross-file interactions. Using benchmarks like DeepCodeBench and LocBench, the team compared traditional retrieval methods with graph-guided approaches to understand when structural context helps most. The findings suggest that repository graphs are especially useful for broad, multi-file questions and cases where keyword search alone misses relevant code, while also showing that future work should improve reranking and function-level precision.

Key Learnings

Structure matters most when questions span files. Graph-based retrieval showed the most promise on broad, multi-file questions where the answer depends on relationships across modules, not one obvious code snippet.

Search is still a strong baseline. BM25 and embedding retrieval performed well, especially when questions had clear keywords or location hints, showing that graph methods need strong ranking to beat simpler retrieval.

Better answers depend on better context. The biggest gap was not how clearly models wrote answers, but whether they retrieved the right facts. Future work should focus on context selection, reranking, and finer function-level localization.

Possible Next Steps

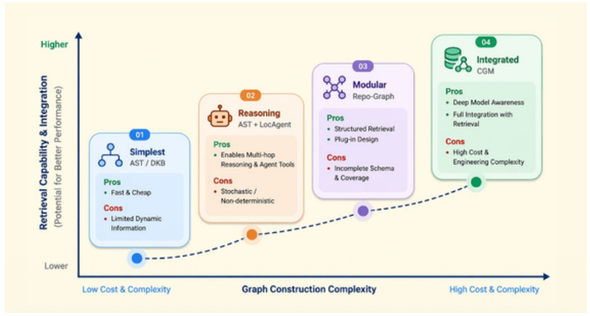
Improve graph-aware reranking.
Study how to rank files, modules, and functions after graph traversal so the system not only finds the right area of the repo, but surfaces the most useful code first.

Expand structural signals beyond code graphs.
Add test-to-code links, commit history, issue threads, PR context, and dependency metadata to see whether richer repository context improves answer quality.

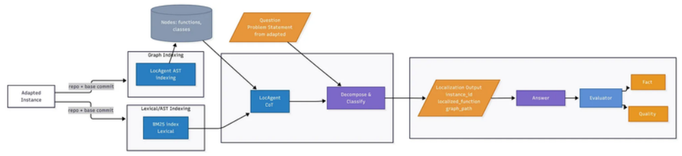
Evaluate by developer intent.
Run separate experiments for planning, debugging, and exploratory questions to understand which retrieval strategies work best for each real-world engineering workflow.

Engagement Highlights

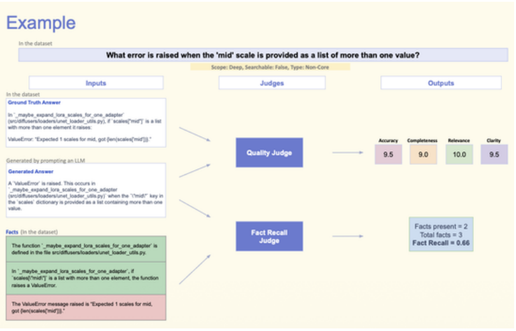
Hindsight identifies gaps in AI code retrieval and groups developer questions by intent: planning, exploration, and debugging.



The project compares graph construction strategies, balancing retrieval capability against implementation cost and complexity.



Hindsight combines lexical search and graph indexing to localize relevant code, generate answers, and evaluate factual quality.



An example evaluation compares a generated answer against ground-truth facts, scoring both factual recall and overall answer quality.

Key Numbers: Duration: 14 weeks Number of students: 4 Technologies utilized / researched: 9

Links: Blog LinkedIn Report Repo Deck Poster

CONCLUSION

As we wrap up this edition of the 99P Labs' University Innovation Impact Report, we extend our sincere gratitude for your engagement and the time you have dedicated to reviewing our endeavors from the spring semester of 2026. Your interest is the cornerstone of our continued pursuit of excellence in academic and research collaborations.

This document is not only a reflection of our past efforts but also a stepping stone towards future collaborations. We invite you to share your insights or express your interest in joining us on this journey of innovation and discovery.

Should you wish to discuss the content of this report further or explore avenues for collaboration, please do not hesitate to reach out to Rajeev Chhajer at rajeev_chhajer@honda-ri.com or Ryan Lingo at ryan_lingo@honda-ri.com.

We are excited about the potential collaborations that may arise from this report and are looking forward to the opportunity to bring these prospects to fruition together.

Thank you once again for your interest and the possibility of future collaborations. Your participation is invaluable to us, and we eagerly anticipate your thoughts and contributions.