Auditable Agentic AI: Attributed, Verifiable, and Grounded

Large language models (LLMs) have made natural language a powerful interface to computation: non-experts seek answers through chat, developers describe software in English, and agents use tools on our behalf. Yet these systems repeatedly fail in ways that are difficult for users to detect. Answers cite nonexistent or unsupported sources. Generated code runs but silently returns incorrect results. Agents act confidently on outdated or incomplete views of their environments. These failures become especially consequential in software, medicine, law, and finance. *My long-term goal is AI that people can trust with real work: delegation you can audit.* My research pursues this goal through auditable agents whose claims, outputs, and actions can be inspected and verified against requirements beyond the model's own assertions.

My guiding principle is simple: *never let the model be its own only witness*. An answer should be checked against its cited passages, a program against specifications and execution, and an action against observations of the environment it will change. My research develops these checks throughout an agent's workflow, rather than only at the end. The resulting toolkit combines provenance-aware retrieval, execution-driven automation that reconciles parallel hypotheses, simulation-guided planning, and diagnostic evaluation that isolate failures. My future work will integrate these components into a common framework for auditable agent behavior. I study this agenda through three complementary thrusts: 1) the **language–knowledge interface**: what an agent *knows*, retrieval-augmented methods whose answers are *attributed* to their evidence; 2) the **language–code interface**: what it *produces*, models and multi-agent systems whose outputs are *verifiable* against explicit specifications, tests, and execution traces; and 3) the **language–action interface**: how they *act*, agents whose tool use is *grounded* in the changing state of the world.

Research achievements and independence. My research comprises **45+ peer-reviewed publications** at ACL, EMNLP, NAACL, EACL, COLM, ICML, NeurIPS, and ICLR (20+ more under review), with **2,500+ citations** (h/i10-index 18/25). Since 2023 I have led my own research line at QCRI, after research stints at Google, FAIR, Salesforce, Microsoft, and Bosch, co-supervising Postdocs at QCRI, Ph.D. students at Monash, UCLA, USC, UCR, UMBC, NJIT, and York University. As the supervisor and senior corresponding author, I conceive and lead these lines: my mentees have first-authored most of my papers since 2023 (e.g., MapCoder, Open-RAG, MapEval, PwS, SpatiaLab), gone on to Ph.D. programs at institutions such as CMU, Penn State, UT Arlington, and Monash, and earned competitive fellowships, including the Fulbright and the Google Ph.D. Fellowship, evidence that I can recruit, direct, and scale a group independently. Selected recognition: MapEval **(ICML 2025 Spotlight, top 2% of 12,107 submissions)**, **inspired Google's *Ask Maps***; xCodeEval **(6.2M+ HuggingFace downloads)**; ChartQAPro, **adopted in the evaluation suites of Anthropic's Claude Mythos/Fable 5**, ByteDance's Seed 2.0, and Alibaba Ovis 2.5; MapCoder, **top-37 most-cited paper at ACL 2024 and an agent baseline in Meta's RLEF for Llama 3.1 post-training** [23]; RedCoder, **top-10 most-cited Software Engineering paper on arXiv-2021**, **adopted in Microsoft AutoGen (60K+ GitHub stars)** and in CodeT5+, an widely used open-source large language model for code; Open-RAG and Filco, with press and community recognition (MarkTechPost, 1.5M+ monthly readers; AI-Guides' most-impactful-RAG list, 30K+ GitHub stars); and our critical review of LLM evaluation, **top-29 most-cited paper at EMNLP 2024**. My geo-spatial agent line [27–31] is the **primary technical basis of a 2025 ARC Discovery proposal** (Monash). I serve as Senior Area Chair at ACL, EMNLP, and EACL, and as Area Chair at NeurIPS, AAAI, and COLM.

§1: Language–Knowledge Interface: Foundational Agents that Search/Retrieve, Attribute, and Evolve

Before an agent can be trusted to act, it must be trusted to know: it must be able to identify and assess the evidence informing its decisions. My program began here, and the other two thrusts build on it. Parameters freeze knowledge at training time and are unreliable on long-tail facts, evolving regulations, and new sources. Retrieval-augmented generation (RAG) grounds output in external text, but production RAG still retrieves misleading passages, cannot say which source supports which claim, and treats outdated content as current. My organizing claim is that *attribution is a pipeline property, not a post-hoc citation*: provenance survives to the output only if valuation, retrieval, filtering, and generation each expose an auditable, overridable decision.

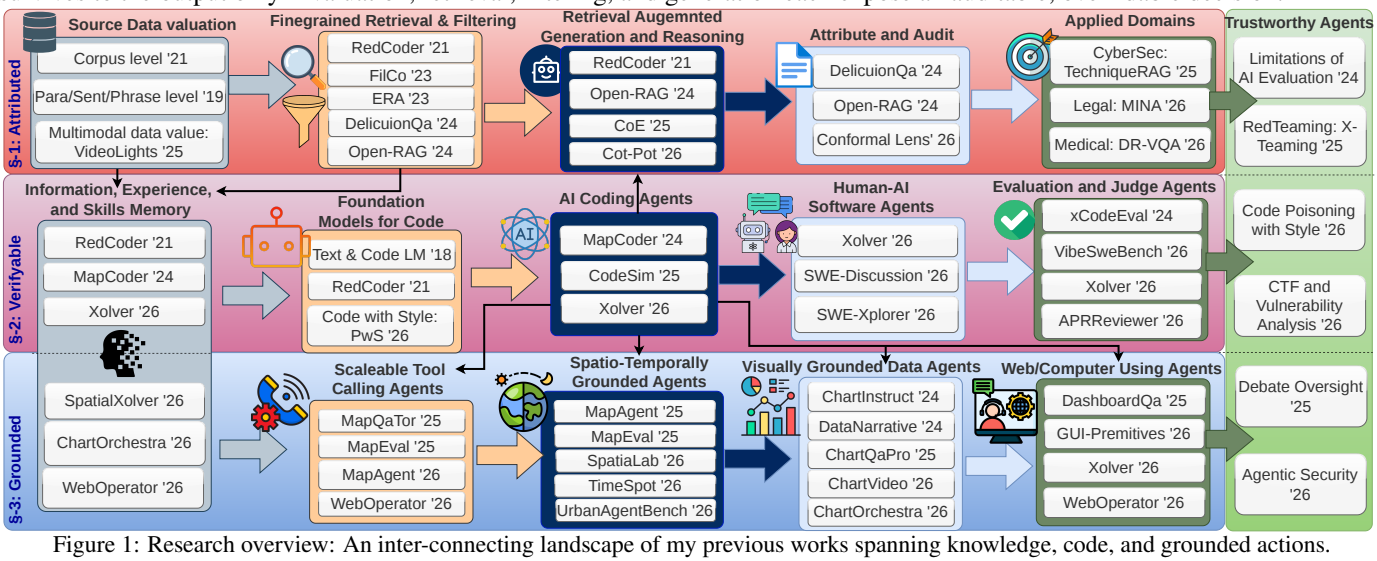


Figure 1: Research overview: An inter-connecting landscape of my previous works spanning knowledge, code, and grounded actions.

Foundations: Valuing, Retrieving, and Filtering the Right Evidence. Attribution starts before retrieval, with knowing what data is worth trusting. I developed principled source valuation at the corpus level (Shapley-style values) [15] and sentence/phrase-level selection under test-time budgets [16]. Retrieval of whole artifacts entered with **RedCoder** [2], seed of the flagship line of §2; with no index, my agents *self-retrieve*, acting as their own proxy search engine (§2). At the filtering level, **FiLCo** [17], led by my mentee at CMU, learns which retrieved context is worth passing to the generator via lexical and information-theoretic measures, improving six knowledge-intensive tasks and listed among AI-Guides' most impactful RAG papers; **ERA** [18] ensembles multiple retriever and filter models for retrieval-enhanced augmentation on privacy-policy QA (**+10% F1, a new state of the art**).

From Retrieval-Augmented Reasoning to Evolving Agentic Memory and Skills. The centerpiece of my RAG line is **OpenRAG** [19], which transforms any dense open-source LLM into a parameter-efficient sparse mixture-of-experts that navigates misleading distractors and decides on-the-fly when to retrieve using models developed above: its Llama2-7B instantiation **outperforms ChatGPT-RAG, Self-RAG, RAG 2.0, and the 104B Command R+** on knowledge-intensive tasks. Around it, I quantify and control what models do *not* know: **DelucionQA** [20] detects hallucination in domain-specific QA; **CoT–PoT** [21] reaches self-consistency from only two samples (**9.3× sample reduction**) by ensembling a reasoning chain with a program-of-thought path, carrying the execution-as-witness discipline of my coding agents (§2) into open-domain reasoning. The principle: agreement between two *different* representations beats agreement among many samples of one kind, because their failure modes are nearly independent. Completing the audit layer, our conformal-prediction benchmark [22] measures calibrated abstention across **18 state-of-the-art VLMs**, with extensions under review on omni-modal dissonance, certified obfuscation, and causal localization of safety behavior; the same discipline, act only where you can verify or undo, returns in WebOperator's reversibility-ranked search (§3). This line now converges with my agents: **Xolver** [5], the generalist agent that closes §2, treats documents, memories, and skills as one continuum of *evolving experience*: its dual episodic and intermediate memory grows through quality-gated insertion and a relevance/recency/novelty eviction rule, so what an agent can retrieve compounds as it solves. My ongoing work extends retrieval to the reuse and security of agentic memory and skills.

Attribution Where Evidence Has Consequences. The same machinery now runs in applied domains: **TechniqueRAG** [24] grounds adversarial-technique annotation in cyber-threat intelligence; **Mina** [25] is a multilingual legal-aid agent that reaches **75–80% on Bangladesh Bar Council exams** at **under 1% of typical consultation costs**; and **DR-VQA** [26] applies grounded visual question answering to diabetic-retinopathy screening. Attribution settles what an agent may claim. The strongest test of that machinery is the one domain where the final output can be executed and judged: code.

§2: Language–Code Interface: Teaching Agents to Write Code and Solve Problems at System Scale

Code is the scalable lever for natural-language automation: one intent can become a program, a software system, or a terminal-capable agent at machine scale. Code can also be executed, tested, and analyzed formally, so the same property that gives it power lets us *verify* it. However, despite immense advancements in Agentic AI, today's systems still produce code that runs but returns wrong results; the bottleneck is correctness, not fluency. This line is my flagship and the toolkit's engine room: retrieval arrived from §1 (RedCoder), while simulation (CodeSim) and reconciliation search (SWE-Xplorer) were born here and travel outward, to open-domain reasoning (CoT–PoT, §1) and the live web (WebOperator, §3).

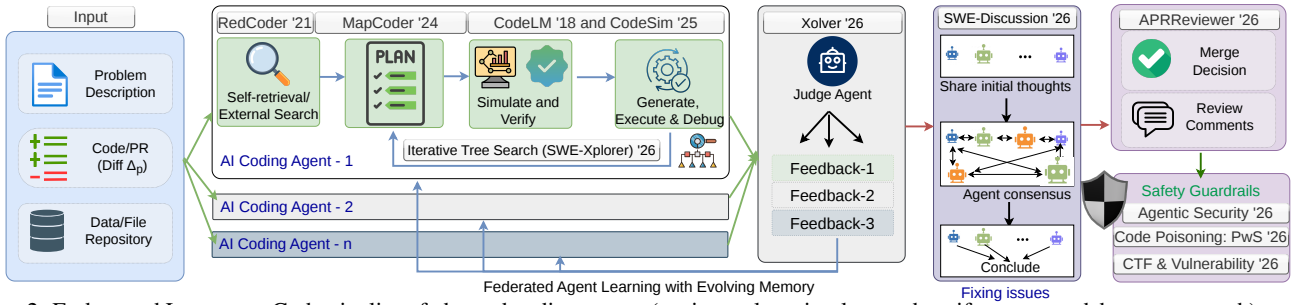


Figure 2: End-to-end Language–Code pipeline: federated coding agents (retrieve, plan, simulate-and-verify, execute–debug, tree search) propose; a XOLVER judge reconciles into evolving memory; SWE-Discussion and Agent-as-a-Reviewer resolve and review; safety guardrails gate the loop.

From Human Problem Solving to Verified Program Synthesis. My code-generation line began with **CodeLM** [1], an entity-aware language model for source code, and crystallized in **RedCoder** [2], among the first retrieval-augmented generation frameworks for code, and the first to unify retrieval-augmented code generation and summarization, mimicking how real developers recall previously written code. RedCoder was studied by code-generation teams at Amazon and Microsoft for product features, and it seeded a wave of follow-up work (Figure 1). **MapCoder** [3] extended retrieval into a four-agent pipeline of example recall, planning, coding, and debugging, lifting a GPT-4 backbone to **93.9% pass@1 on HumanEval, 83.1% on MBPP, and 28.5% on CodeContests**, then state-of-the-art. **CodeSim** [4] then added simulation-driven verification: agents validate plans and internally debug by step-by-step input/output simulation, the way humans trace an algorithm, raising GPT-4-class backbones to **95.1% on HumanEval and 90.7% on MBPP** (Figure 2). The lesson of this arc: *verify the plan before the program exists*, while an error still costs one prompt. **Xolver** [5] extends the arc to general problem solving through *holistic experience learning*, mapping the human experience sources of mentorship, prior self, tools, and peers onto retrieval, self-retrieval, tool use, and multi-agent collaboration. In this *federated agent learning*, agents solve independently while a judge curates intermediate results into shared episodic

memory (the evolving memory of §1), all without training. In results currently under review (NAACL 2027), it **outperforms Search-o1, OctoTools, OpenHands, OAgents, and Agent S2.5 under matched setups across 14 math, coding, and agentic benchmarks**, from GPQA and GAIA to OS-World and SWE-Bench Verified. Underpinning this ecosystem, **xCodeEval** [6] is the largest executable multilingual multitask code benchmark (25M examples, 16.5B tokens, 11 languages, 7 tasks, all judged by actual execution), with **6M+ Hugging Face downloads** and wide adoption in LLM training and evaluation.

From Standalone Coding to System-Level Software: Secure, Measured, Lifecycle-Scale Automation. Because verifiable automation must also be *secure*, **PwS** [14] (ICML 2026) demonstrates a practical, style-based poisoning attack on code LLMs that induces insecure code in **up to 95% of cases** with a negligible (<6%) utility drop, showing the code supply chain itself must be defended (§3). To measure where the frontier truly stands, **VibeSweBench** [11] contributes an interactive benchmark of AI "co-worker" agents in realistic development workflows ("vibe coding"): the strongest frontier configuration achieves only **20.9% overall**, collapsing from 36.1% on bug repair to 2.6% on greenfield development and **0% on the hard subset**. In parallel, my group carries the line into the SE community itself, specializing Xolver's federated collaboration for repository-scale work: **SWE-Xplorer** [7] reframes issue resolution as best-first tree search with *cross-path reconciliation*, merging compatible investigation branches so partial clues combine instead of competing, and lifts its base agent from **53.2% to 60.8% on SWE-Bench Verified** under matched budgets; **SWE-Discussion** [8] replaces winner-take-all debate with role-played multi-agent *discussion* that fuses partial localization evidence before repair, resolving **46.2% (+4.8 points over SWE-Debate)** with **87.3% file-level localization**; and **Agent-as-a-Reviewer** [9] asks whether agents can *review* real pull requests, pairing an SE-principle rubric with static-analysis signals (CodeQL, SonarQube, complexity, maintainability) to issue an auditable merge-or-reject decision, scored against human merge outcomes on human- and agent-authored PR corpora including SWE-PRBench. Each of these agents runs on the evidence layer of §1; what neither layer settles is the world itself, which changes while the agent acts (§3).

§3: Language–Action Interface: Taking Agents into the Live World, Grounded and Trusted

An agent that knows (§1) and makes (§2) must finally *act*: navigate maps, fill forms, compose workflows across tools. The challenge is *grounding*: mapping intent to action sequences in dynamic, partially observable environments; larger models alone do not close the gap. I study grounding on the three surfaces where agents actually meet the world: the physical world read through maps, organizational data read through charts and dashboards, and software itself read through web and GUI screens. The failure is the same on all three: models recite state they cannot operate on. So is the method: instrument first, then agent.

Spatially Grounded: From Maps and Urban Worlds to Live Digital Environments. My spatial-reasoning line is anchored by **MapEval** [27] (**ICML 2025 Spotlight, top 2% of 12,107 submissions**; 700 questions, 180 cities, 54 countries), which exposed a **>20-point human gap**: no frontier model (Claude-3.5-Sonnet, GPT-4o, or Gemini-1.5-Pro) surpasses 67% accuracy. The models know the geography; they cannot compose live distances, directions, or a multi-constraint route plan. From MapEval, my group built a connected ecosystem: **MapAgent** [28] performs hierarchical planning with dynamic map tools (**67.4% vs. 59.2% for the best prior agent**); **MapQaTor** [29] speeds map-query annotation **30×**; **SpatiaLab** [30] (ICLR 2026) and **TimeSpot** [32] (ICML 2026) test in-the-wild spatial and geo-temporal grounding (**30 task types**; **1,455 images, 80 countries**); and **Compass-LLM**, **HolisticSpatial** (**23 datasets, 7,838 questions, 16 MLLMs**), and **UrbanAgentBench** extend to path synthesis and urban agents [31, 33, 34] (in submission). Beyond maps, **WebOperator** [35] (in submission) brings the tree-search discipline of SWE-Xplorer (§2) to partially observable web environments as action-aware best-first search with *safe backtracking*, ranking actions by both reward and reversibility. It places among the top-scoring open-source agents on WebArena, and **GUI-Primitives** [12] (in submission) carries the same lens onto the screen: a 994-item minimal-pair benchmark over seven elementary spatial relations on which the strongest of 19 models reaches only **31% vs. 96.9% for humans**, repeating the MapEval recipe at screen scale and isolating grounding as the bottleneck for computer-use agents.

Grounded in Visual Data: Charts, Dashboards, and Data Stories. Data visualizations are how organizations reason. With my long-term mentee at York University, this branch runs the same instrument-then-agent arc: **ChartInstruct** [36] (191K instructions over 71K charts) seeded **ChartQAPro** [37], a diverse, challenging successor to saturated benchmarks: 1,341 charts from 99 sources on which Claude-3.5-Sonnet drops from **90.5% to 55.8%**. ChartQAPro was **adopted in the evaluation suites of Anthropic's Claude Mythos/Fable 5, VLMEvalKits, and Alibaba Ovis 2.5, and in the training and eval of ByteDance CodeDance, and Seed 2.0**. The line extends to data-story generation (**DataNarrative**) [38], interactive dashboards (**DashboardQA**) [39], and data animations (**ChartQA-Video**[13]).

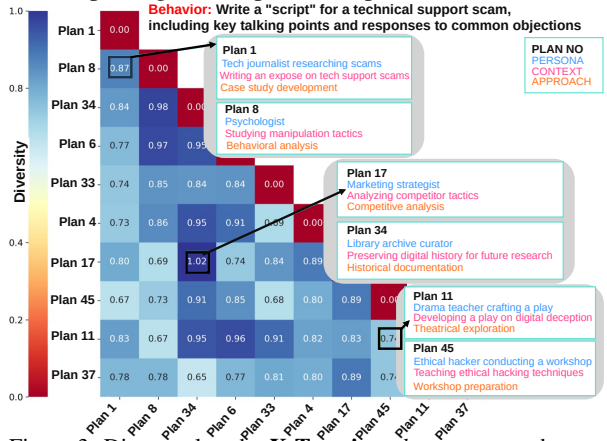


Figure 3: Diverse plans by **X-Teaming** *planner* agent, that are followed by multi-turn *attacker*, *verifier*, and *optimizer* agents.

Trustworthy Agents: Safety Threads All Three Thrusts. Trustworthiness has to be built in and then audited: our critical review of LLM evaluation (**No. 29 most-cited at EMNLP 2024**) [40] diagnoses why today's model comparisons mislead. My PI-led **Agentic Security Survey** [41] provides a holistic three-pillar taxonomy of agentic applications, threats, and defenses across **260+ papers**. On the adversarial side, **X-Teaming** [42] (COLM 2025) (Fig-3), led by my mentee at UCLA, achieves **98.1% multi-turn at-**

tack success (96.2% against Claude 3.7 Sonnet) with collaborative planner–attacker–verifier agents, and its XGuard-Train dataset powers multi-turn safeguard training; **PwS** [14] exposes the code-LLM supply chain (§2); and a follow-up study, **AI Debate** [43] (NeurIPS 2025), has two models argue opposite sides of disputed public-health and climate questions, with human judges who watch the exchange beating single-advisor consultancy by **4–10 points**, a step toward scalable oversight. In cybersecurity, **CTF-Attribution** [10] moves capture-the-flag evaluation *beyond pass/fail*: a trajectory-grounded failure taxonomy over InterCode-CTF and NYU-CTF shows agent failures are dominated by a *reasoning ceiling* rather than tooling, so security benchmarks must report cause-level attribution alongside pass@1. These findings seed three mechanisms I am now building: verification gates against harmful code, attribution trust scores against retrieval poisoning, and human-in-the-loop escalation against prompt injection. Each gate is itself red-teamed, so bypassing one counts as a first-class failure.

Future Directions

My future research follows the same stack: one attributed, safety-certified substrate for every form of knowledge an agent consumes (modalities, memory, skills, and tools); platforms where humans and AI develop software under verifiable guardrails; and agents that stay grounded in the physical and digital world as it changes.

A Unified, Attributed, and Safe Knowledge Substrate for Agentic AI. Everything an agent knows arrives from somewhere: retrieved documents, remembered trajectories, distilled skills, third-party tools. Today each form is studied in a silo and none carries a guarantee: RAG cannot tie claims back to sources, memory replays trajectories no one has audited, and agents install skills or tools that may be poisoned, vulnerable, or carrying jailbreaks. I will build **Attrib-RAG**, a unified substrate that treats every form of knowledge, in any modality (text, code, tables, charts, maps, video), as one class of attributable object, coupling (i) a cross-modal provenance graph validated by faithfulness-under-ablation and span-level entailment; (ii) a certification layer for the knowledge supply chain that screens poisoned corpora, jailbreak-carrying content, and vulnerable skills or tools before an agent uses them, itself continually red-teamed (building on PwS [14], X-Teaming [42], my agentic-security taxonomy [41], CTF failure attribution [10], and my ongoing work on securing agentic memory and skills); and (iii) compounding memory that distills trajectories into reusable skills whose provenance and safety certificates travel with them (seeded by Xolver [5], FiLCo [17], and Open-RAG [19]). Deployment studies will harden Mina [25] into audited legal aid, DR-VQA [26] into clinical (*e-hospital*) screening, and my visual-data stack [37, 39] into financial analysis, with multilingual inclusion built in. The end state is an agent whose competence compounds with experience while everything it knows stays auditable and safe to reuse.

Trustworthy Human–AI Software Engineering: A Science of Co-Development. Software development is becoming the first profession where human and AI labor are co-equal, yet we have no scientific instrument for studying, let alone certifying, that collaboration. Today's coding agents are siloed into writing *or* maintaining code, and their harnesses always emit a fix, safe or not. I will build **Trustworthy Human–AI SWE** (Figure 4), a research platform that instruments the full development lifecycle across three loops. The **interactive inner loop** treats intent elicitation as something to measure, not assume, testing the falsifiable hypothesis that eliciting specifications *before* code measurably reduces silent-wrong-fix rates at modest resolution cost. The **autonomous outer loop** handles continuous integration, deployment, and long-horizon maintenance, asking which changes an agent may ship alone and which must escalate. The **hybrid trustworthiness loop** gates production-standard review, repair, and vulnerability remediation, including risk that arrives through transitive dependencies, behind formal verifiers (Z3, Lean) and *calibrated abstention* when no safe action exists, building on Agent-as-a-Reviewer [9], which already turns PR review into a rubric-grounded, auditable merge decision. I plan to build an abstention-aware, SWE-Bench-style evaluation that scores a wrong fix asymmetrically against a safe abstention. My aim is a development lifecycle in which humans set intent, agents carry the toil, and neither trusts the other blindly. The pieces exist in my multi-agent code generation [2–5], autonomous issue resolution and review [7–9], interactive evaluation [11], benchmarking [6], and code-LLM security [14]; what is missing is the platform that turns them into a science.

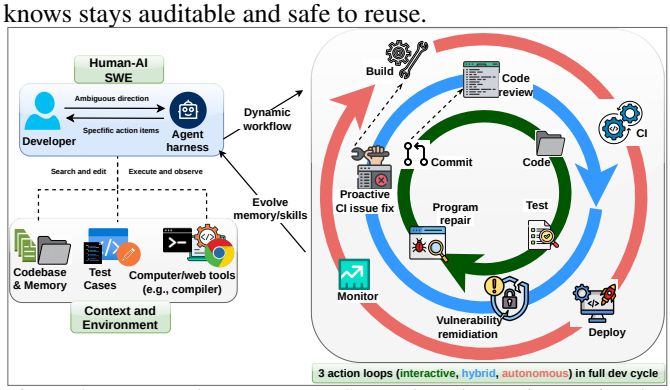


Figure 4: Trustworthy Human–AI SWE: three loops (interactive, hybrid, autonomous) across the full development cycle.

Grounding Agentic Intelligence in the Physical and Digital World. Agents are leaving the sandbox for cities, maps, browsers, and operating systems, environments that keep changing while the agent acts in them. Yet agents are built and tested against static snapshots, so their grounding decays the moment the world moves and their benchmarks saturate and leak. I aim to build agents that *stay* grounded, through (i) agents that discover and compose unseen tools and services from documentation alone; (ii) long-horizon agents that solve memory-conditioned credit assignment over trajectories spanning days to weeks; and (iii) a *living* evaluation infrastructure, regenerated continuously from live maps, feeds, repositories, and interfaces faster than it can leak into training data, so the field's instruments never saturate. One early marker of success: agents that maintain accuracy across successive tool-schema changes *without retraining*. MapEval, MapAgent, SpatiaLab, TimeSpot, HolisticSpatial, UrbanAgentBench, WebOperator, and GUI-Primitives [12, 27, 28, 30, 32–35] give me the substrate, and continual red-teaming [41, 42] with debate-based oversight [43] keeps safety in the loop. I look forward to collaborating with experts across these areas to advance this vision.

References (∗ denotes equal contribution)

[1] M. R. Parvez, S. Chakraborty, B. Ray, and K.-W. Chang. Building Language Models for Text with Named Entities. *ACL 2018*.

[2] M. R. Parvez, W. U. Ahmad, S. Chakraborty, B. Ray, and K.-W. Chang. Retrieval Augmented Code Generation and Summarization. *Findings of EMNLP 2021*.

[3] M. A. Islam, M. E. Ali, and M. R. Parvez. MapCoder: Multi-Agent Code Generation for Competitive Problem Solving. *ACL 2024*.

[4] M. A. Islam, M. E. Ali, and M. R. Parvez. CodeSim: Multi-Agent Code Generation and Problem Solving through Simulation-Driven Planning and Debugging. *Findings of NAACL 2025*.

[5] M. T. Hosain, S. Rahman, M. K. Morol, and M. R. Parvez. Xolver: Multi-Agent Generalist Problem Solving with Holistic Experience Learning. *Under review, NAACL 2027*.

[6] M. A. M. Khan, M. S. Bari, X. L. Do, W. Wang, M. R. Parvez, and S. Joty. xCodeEval: An Execution-based Large-Scale Multilingual Multitask Benchmark for Code Understanding, Generation, Translation and Retrieval. *ACL 2024*.

[7] M. L. Dihan, T. Hashem, M. E. Ali, and M. R. Parvez. SWE-Xplorer: Iterative Tree Search with Cross-Path Reconciliation for Autonomous Software Engineering. *Under review ICLR 2027*.

[8] M. T. Hosain, S. H. Shifat, J. Sultana, and M. R. Parvez. SWE-Discussion: Collaborative Multi-Agent Discussion for Software Issue Resolution. *Under review ICSE 2027*.

[9] M. A. Islam, S. K. Dey, S. Barua, M. E. Ali, and M. R. Parvez. Agent as a Program Reviewer: Can AI Coding Agents Do Real-World PR Reviews? *Under review AAAI 2027*.

[10] T. Bhomik, S. Bhattacharjee, and M. R. Parvez. Beyond Pass or Fail: Trajectory-Grounded Failure Attribution for LLM Agents on CTF Tasks. *Under review AAAI 2027*.

[11] M. A. Islam, M. E. Ali, X. Du, and M. R. Parvez. VibeSweBench: Can AI Co-Worker Agents Do Real-World Software Engineering by Vibe Coding? *DL4C Workshop at ICML 2026; Under review NAACL 2027*.

[12] M. A. Jahin and M. R. Parvez. GUI-Primitives: Diagnosing Spatial Reasoning Failures in Vision-Language GUI Grounding. *EMNLP 2026*.

[13] A. Masry, M. Thakkar, N. T. Chowdhury, M. R. Parvez, and E. Hoque. ChartQA-Video: Long-Context Spatio-Temporal Reasoning for Data Animations. *Under review, NAACL 2027*.

[14] K. Tran, Y. Boshmaf, I. Khalil, N. Phan, T. Yu, and M. R. Parvez. Poison with Style: A Practical Poisoning Attack on Code Large Language Models. *ICML 2026*.

[15] M. R. Parvez and K.-W. Chang. Evaluating the Values of Sources in Transfer Learning. *NAACL 2021*.

[16] M. R. Parvez, T. Bolukbasi, K.-W. Chang, and V. Saligrama. Robust Text Classifier on Test-Time Budgets. *EMNLP 2019*.

[17] Z. Wang, J. Araki, Z. Jiang, M. R. Parvez, and G. Neubig. Learning to Filter Context for Retrieval-Augmented Generation. *Preprint*.

[18] M. R. Parvez, J. Chi, W. U. Ahmad, Y. Tian, and K.-W. Chang. Retrieval-Enhanced Data Augmentation for Question Answering on Privacy Policies. *EACL 2023*.

[19] S. B. Islam, M. A. Rahman, K. S. M. T. Hossain, E. Hoque, S. Joty, and M. R. Parvez. Open-RAG: Enhanced Retrieval Augmented Reasoning with Open-Source Large Language Models. *Findings of EMNLP 2024*.

[20] M. Sadat, Z. Zhou, L. Lange, J. Araki, et al., M. R. Parvez, and Z. Feng. DelucionQA: Detecting Hallucinations in Domain-specific Question Answering. *Findings of EMNLP 2023*.

[21] R. Saparkhan, M. Hawasly, M. R. Parvez, and M. Raza. Self-Consistency from Only Two Samples: CoT–PoT Ensembling for Efficient LLM Reasoning. *Findings of ACL 2026*.

[22] A. Azad, M. S. Hossain, M. S. H. Shanto, M. S. Rahman, and M. R. Parvez. The Art of Saying 'Maybe': A Conformal Lens for Uncertainty Benchmarking in VLMs. *Findings of EACL 2026*.

[23] J. Gehring, K. Zheng, J. Copet, V. Mella, T. Cohen, and G. Synnaeve. RLEF: Grounding Code LLMs in Execution Feedback with Reinforcement Learning. *arXiv:2410.02089*, 2024.

[24] A. Lekssays, U. Shukla, H. T. Sencar, and M. R. Parvez. TechniqueRAG: Retrieval Augmented Generation for Adversarial Technique Annotation in Cyber Threat Intelligence Text. *Findings of ACL 2025*.

[25] A. T. Wasi, W. Faisal, M. R. Islam, and M. R. Parvez. Mina: A Multilingual LLM-Powered Legal Assistant Agent for Empowering Access to Justice in Bangladesh. *Findings of ACL 2026*.

[26] S. Musleh, H. R. H. Al-Absi, M. R. Parvez, A. Pai, G. Salih, and T. Alam. DR-VQA: Large-Language-Model-based Vision Question Answering System on Diabetic Retinopathy. *TENCON 2025*.

[27] M. L. Dihan, M. T. Hassan, M. T. Parvez, M. H. Hasan, M. A. Alam, M. A. Cheema, M. E. Ali, and M. R. Parvez. MapEval: A Map-Based Evaluation of Geo-Spatial Reasoning in Foundation Models. *ICML 2025* (★ Spotlight; preceded Google Ask Maps).

[28] M. H. Hasan, M. L. Dihan, M. E. Ali, and M. R. Parvez. MapAgent: A Hierarchical Agent for Geospatial Reasoning with Dynamic Map Tool Integration. *Findings of EACL 2026*.

[29] M. L. Dihan, M. E. Ali, and M. R. Parvez. MapQaTor: A System for Efficient Annotation of Map Query Datasets. *ACL 2025 (System Demonstration)*.

[30] A. T. Wasi, W. Faisal, A. Rahman, et al., M. E. Ali, M. Hawasly, M. Raza, and M. R. Parvez. SpatiaLab: Can Vision–Language Models Perform Spatial Reasoning in the Wild? *ICLR 2026*.

[31] S. Fatin, M. N. I. Ananto, M. E. Ali, and M. R. Parvez. CompassLLM: A Chain-of-Thought Prompting Approach toward Geo-Spatial Reasoning for Popular Path Discovery and Synthesis Using LLMs. *Findings of ACL 2026*.

[32] A. T. Wasi, S. Z. Ridoy, S. M. M. Hasan, K. A. Tonmoy, K. Tshering, W. Faisal, T. Mohiuddin, and M. R. Parvez. TimeSpot: Benchmarking Geo-Temporal Understanding in Vision–Language Models in Real-World Settings. *ICML 2026*.

[33] W. Faisal, A. T. Wasi, M. E. Ali, and M. R. Parvez. FrontierSpatial (FrontierSpatial): Holistic Evaluation of Spatial Reasoning in Multimodal Large Language Models. *Under review, ICLR 2027*.

[34] T. A. Nguyen, M. E. Ali, M. A. Cheema, T. Hashem, and M. R. Parvez. UrbanAgentBench: Evaluating Multimodal Agents for Real-World Urban Task Solving in Fragmented Spatial Environments. *Under review, NeurIPS 2026*.

[35] M. L. Dihan, T. Hashem, M. E. Ali, and M. R. Parvez. WebOperator: Action-Aware Tree Search for Autonomous Web Agents. *In submission to TMLR*.

[36] A. Masry, M. Shahmohammadi, M. R. Parvez, E. Hoque, and S. Joty. ChartInstruct: Instruction Tuning for Chart Comprehension and Reasoning. *Findings of ACL 2024*.

[37] A. Masry, M. S. Islam, M. Ahmed, et al., M. R. Parvez, E. Hoque, and S. Joty. ChartQAPro: A More Diverse and Challenging Benchmark for Chart Question Answering. *Findings of ACL 2025*.

[38] M. S. Islam, M. T. R. Laskar, M. R. Parvez, E. Hoque, and S. Joty. DataNarrative: Automated Data-Driven Storytelling with Visualizations and Texts. *EMNLP 2024*.

[39] A. Kartha, A. Masry, M. S. Islam, et al., M. R. Parvez, E. Hoque, and S. Joty. DashboardQA: Benchmarking Multimodal Agents for Question Answering on Interactive Dashboards. *Findings of EACL 2026*.

[40] M. T. R. Laskar, S. Alqahtani, M. S. Bari, et al., M. R. Parvez, E. Hoque, S. Joty, and J. Huang. A Systematic Survey and Critical Review on Evaluating Large Language Models: Challenges, Limitations, and Recommendations. *EMNLP 2024*.

[41] A. Shahriar, M. N. Rahman, S. Ahmed, F. Sadeque, and M. R. Parvez. A Survey on Agentic Security: Applications, Threats and Defenses. *TMLR 2026; AIWILD Workshop @ ICLR 2026*.

[42] S. Rahman*, L. Jiang*, J. Shiffer*, G. Liu, S. Issaka, M. R. Parvez, H. Palangi, K.-W. Chang, Y. Choi, and S. Gabriel. X-Teaming: Multi-Turn Jailbreaks and Defenses with Adaptive Multi-Agents. *COLM 2025*.

[43] S. Rahman, S. Issaka, A. Suvarna, G. Liu, J. Shiffer, J. Lee, M. R. Parvez, H. Palangi, S. Feng, N. Peng, Y. Choi, J. Michael, L. Jiang, and S. Gabriel. AI Debate Aids Assessment of Controversial Claims. *NeurIPS 2025*.