Programming is a social activity that heavily relies on collaboration and communication. Instructors design tutorials that align with learners' motivations and cognitive abilities; software engineers build upon one another's work with publicly available open-source libraries; data scientists collaborate with domain experts to tackle cross-disciplinary challenges. Can programming be more than just a set of instructions for machines? What if we viewed programming as a form of literature, written not only to communicate with computers but also with people? This shift in perspective – from programming as machine-oriented to human-centered – has the potential to fundamentally transform how we design, write, and understand code.

My research seeks to help people communicate the dynamic and abstract process of programming more effectively by designing expressive, intelligent, and human-centered interactive systems. Using technical HCI methodologies, my work builds around three themes. First, I explore the design space for **multimodal representations of programming** that integrate code, textual descriptions, visual aids, interactive explorations, and contextual information to improve the explanation and communication of programming. Second, I investigate how these systems can be tailored to users and contexts, **making representations more flexible and adaptive through automation**. Finally, I examine how system design can **balance automation with essential human values** such as trust, autonomy, and explainability, enhancing human capabilities rather than replacing them.

My work is published at top-tier human-computer interaction (HCI) venues (e.g., CHI, CSCW, TOCHI) and has received two best paper awards [3, 11] and three honorable mention awards [10, 12, 14]. My projects have also led to publications at the intersection between HCI and other areas such as software engineering, machine learning, and natural language processing (e.g., ICSE, EMNLP, IJCAI). In addition, I pursue a broader impact of solving real-world problems by outreach to industry and open-source communities. For example, the Jupyter RTC project, which is building collaborative features for the most popular data science development environment, cited my work as one of the primary sources for design inspiration [1].

Multimodal Representations of Programming

Programming processes and concepts can be represented through multiple modalities — natural language, programming languages, visual representations, and even embodied interactions. The first line of my research focuses on exploring diverse approaches to effectively and expressively communicate programming both in professional and educational contexts.

Natural Language: Computational notebooks, widely used for collaborative and exploratory work in both professional and educational settings, combine code, documentation, and visual outputs. They enable real-time, synchronized editing, offering a valuable platform to study collaboration in data science workflows. However, these notebooks also present challenges, such as disorganization and difficulty for collaborators to follow evolving projects. In a mixed-method study [11] (CSCW 2019, Best Paper Award), I was among the first to investigate the benefits and limitations of computational notebooks in collaborative data science. This study identified two main collaboration styles: asynchronous work on individual notebooks and real-time collaborative editing on shared notebooks. I developed a taxonomy of these styles, highlighting both the advantages of synchronous editing (e.g., reduced communication costs) and its challenges (e.g., editing conflicts). To help collaborators catch up with ongoing projects, I developed Callisto [14] (CHI 2020, Honorable Mention Award), which links discussions to notebook elements (code, output, cells, versions diffs). It helps collaborators track design decisions and reasoning processes that naturally occur but are disconnected in the discussion space. Evaluation studies demonstrated that this approach significantly improves the onboarding experience for new collaborators, reduces miscommunication, and prevents redundant work. In addition to synchronous collaboration, asynchronous work in data science often results in disorganized notebooks with no such contextual information to leverage. I also created Themisto [13] based on graph neural network model [5], which nudges data scientists to update documentation with AI-generated suggestions as a starting point.

Shifting from professional to educational contexts, my ongoing work GALAXY [2] and SPARK [20] help instructors analyze student code submissions and monitor multi-step programming exercises, providing insights into student learning patterns and solution variations.

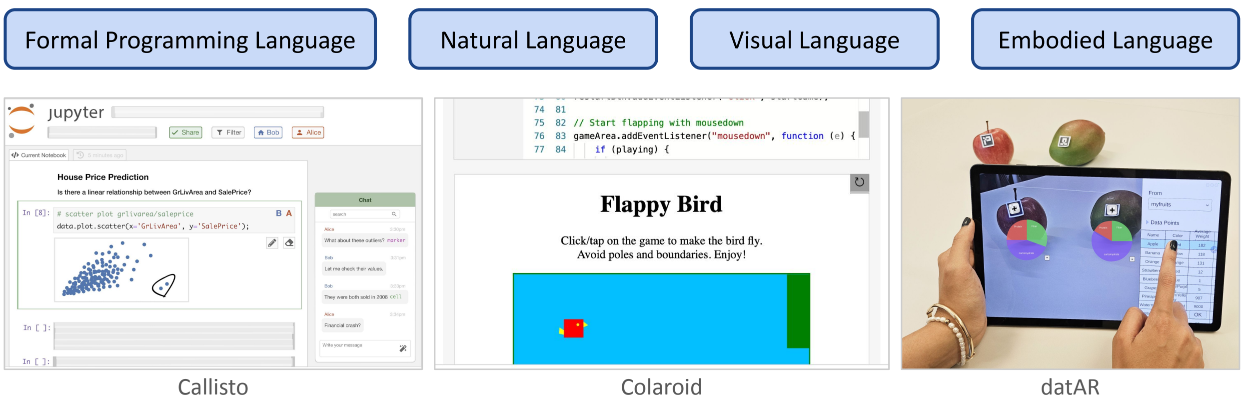

Figure 1: Callisto [14] helps data scientists make sense of synchronous editing in notebooks by linking notebook elements with chat messages. Colaroid [10] supports instructors in creating visual and interactive programming tutorials, while datAR [6] extends beyond screen-based interactions to teach data literacy using everyday objects.

Visual Language: In addition to programming and natural language, my work explores how visual language can represent programming concepts and processes. In professional settings, I developed a tool called Diff in the Loop (DITL) [9], which helps data scientists track not only code changes but also data changes by versioning and visualizing the effects of code on datasets. This approach is essential for exploratory data analysis, as it allows users to monitor how their code impacts the underlying data. Building on this, we further explored how generative AI can distill key analysis, outputs, and textual explanations from complex notebooks to automatically generate presentation slides, aiding in the communication of data insights [21].

In educational settings, visual language and interactivity play a vital role in helping instructors create more effective programming tutorials. However, producing high-quality, multi-stage tutorials is time-consuming. To address this, I developed Colaroid [10] (CHI 2023, Honorable Mention Award), an interactive authoring tool that allows authors to present progressive code snippets and outputs, with highlighted differences and full code context. Learners can experiment with each stage of the tutorial directly in the linked IDE, encouraging more active learning than passive reading. Extending on this idea, our ongoing work explores how this paradigm of interactive, step-by-step, in-application computing notebook can be useful for non-technical users documenting their exploration paths for everyday GUI-based tasks like spreadsheets manipulation or graphical design [19].

Embodied Language: My research on multimodal representations of programming extends from screen-based interaction into the embodied space. By leveraging embodied metaphors and interactions, I aim to motivate and engage younger learners in computational, data, and AI literacy as these are essential skills for them to be a good citizen in the data-driven world. However, high school instructors often face challenges in engaging students from diverse backgrounds with abstract concepts that may seem disconnected from their daily lives. To address this, we introduce the concept of embodied data literacy, which integrates data into everyday objects to make learning more tangible and relatable. This approach is demonstrated through datAR[6], a tablet-based application that uses augmented reality and tangible analysis blocks to help students explore and interact with data. In a case study at a local high school, 15 students with little data science experience were introduced to data literacy by analyzing the nutrition information of familiar snacks. Our findings revealed that, while students initially struggled to connect the concepts to their own lives, datAR fostered hands-on engagement and helped them develop foundational data science skills. We are continuing this thread of research by exploring how everyday objects can be better utilized as data literacy learning tools and expanding the scope from static objects to dynamic human activities.

Adaptive Representations to the User and Domain

No single representation of information works for every context. The most effective way to communicate code depends on the specific needs of the task, the user's expertise, and the domain in which they operate. By aligning representations with the user's abilities and the domain's requirements, we can significantly improve communication and collaboration.

Barriers from Diverse Users: My empirical studies revealed how factors such as the type of programming task, collaborator expertise [4, 15, 16], team size, collaboration synchronicity, and whether the collaboration's goal is productivity or learning [3, 8] can shape the collaboration experience. One example I investigated is interdisciplinary teams. In a study on conversational programmers [12] (CHI 2018 Honorable Mention Award) — adults who learn coding to better communicate with their programmer colleagues without needing to write code themselves, I provided empirical evidence on the unique learning needs of conversational programmers. This research showed how their needs differ from those of end-user programmers and professional programmers, and how modern learning resources, which often focus on artifact creation, can fall short for this group.

Adaptive and Scaffolded Programming Learning Building on my exploration of how users engage with programming tasks, my research focuses on tailoring learning experiences to support diverse learners. In particular, I study a critical computational thinking skill – decomposition, the process of breaking down complex problems into manageable parts. Current self-study methods often provide excessive help or generic solutions that fail to match learners' decomposition strategies. This over-reliance on assistance can impede independent problem-solving and critical thinking. To address this, DBox [7] provides adaptive and targeted assistance, encouraging students to engage deeply with problems and refine their own decomposition strategies. DBox significantly improved learning outcomes, cognitive engagement, and critical thinking, with learners reporting a stronger sense of accomplishment and finding the adaptive assistance beneficial.

Grounding Automations with Human Values

As automation evolves from rule-based systems to LLMs, it opens new opportunities for more expressive and adaptive programming representations. However, it also raises challenges in maintaining transparency, trust, and human values. Below are two examples from my ongoing work that address these challenges through grounding automation with human-centered design principles.

Transparency of AI-Generated Pipeline One area where transparency in AI-generated outcomes is critical is data visualization. AI can automate the generation of data representations from natural language descriptions, enabling users without programming experience to explore data visually. However, when only the final visualization is presented, users may feel disconnected from the underlying process, leading to confusion and doubts about accuracy. To address this, we developed STAGE [18], a system that provides stepwise visual explanations of the AI-generated data analysis pipeline. By revealing intermediate steps and allowing real-time adjustments through spreadsheet-like manipulations, STAGE makes the entire process more transparent and interactive. Users can adjust queries, correct visualizations, and explore data even without a clear starting point, engaging directly with the system. A within-subject study (N=20) showed that STAGE improves user confidence and trust, reduces cognitive load, and enhances exploratory data analysis.

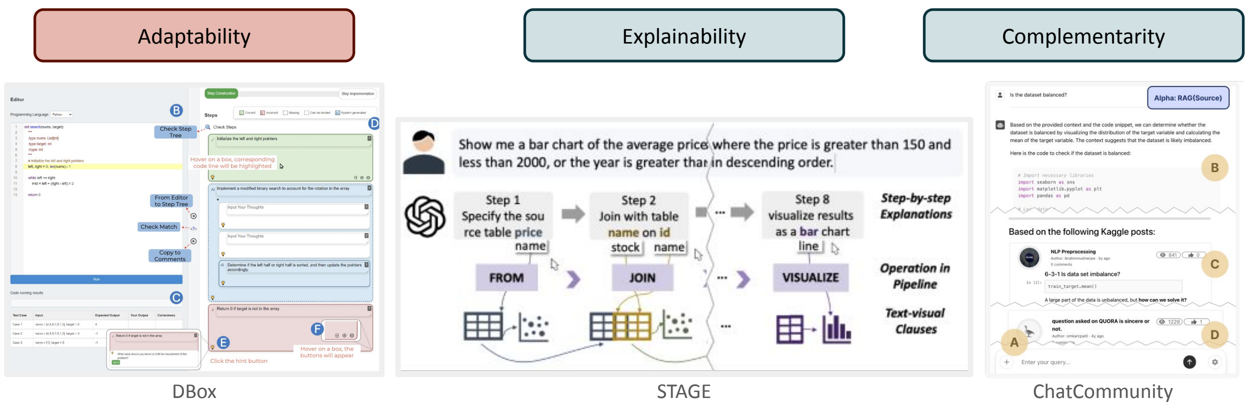

Figure 2: DBox [7] leverages AI to provide adaptive and targeted assistance based on learners' unique decomposition strategies during programming exercises. In contrast, STAGE [18] and ChatCommunity [17] address the limitations of automation by offering stepwise explanations of AI-generated pipelines and grounding AI responses with peer-curated content from online learning communities.

Enriching AI-Generated Responses with Peer Communities: Trained on diverse internet sources, including content from online learning communities, Large Language Model-based chatbots like ChatGPT have significantly improved their ability to generate relevant responses to learners' queries. However, these tools often fail to facilitate direct engagement within online learning communities, and their responses can sometimes be misleading. To address these limitations, we explored an alternative design of AI chatbots [17] that uses a Retrieval-Augmented Generation model that leverages community-contributed content to generate responses enriched with previews of relevant posts and their meta-information(e.g., views, votes, comments). This approach integrates AI-generated responses with community knowledge, fostering more reliable and contextually relevant interactions. We found that enriching RAG responses with peer-curated content improved performance in data science tasks, increased trust in the AI, and enhanced engagement with the community. These findings highlight the potential for generalizing this approach to other online learning communities, positioning AI as a complement to, rather than a replacement for, peer interactions.

Conclusion Programming is no longer just a way for programmers to communicate with machines; it is becoming a vital tool for communication in the rapidly evolving technological world. My research aims to lower the barriers for people to exchange insights with programming by designing expressive, intelligent, and human-centered systems.

References

[1] 2022. JupyterLab RTC. https://jupyterlab-rtc.readthedocs.io/en/latest/about-rtc/user-stories.html

[2] Mamoun Chami, Rasan Younis, Eliot Ullmo, Xiaoyu Zhang, and **April Yi, Wang**. 2024. GALAXY: Understanding Variations in Solutions to Data Science Problems at Scale. (2024). . **In Submission**.

[3] Yan Chen, Jaylin Herskovitz, Gabriel Matute, **April Yi, Wang**, Sang Won Lee, Walter S Lasecki, and Steve Oney. 2020. EdCode: Towards Personalized Support at Scale for Remote Assistance in CS Education. In *2020 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. IEEE, 1–5. **Best Paper Award**.

[4] Will Epperson, **April Yi, Wang**, Robert DeLIne, and Steven Drucker. 2022. Strategies for Reuse and Sharing among Data Scientists in Software Teams. In *44th International Conference on Software Engineering (ICSE)*.

[5] Xuye Liu, Dakuo Wang, **April Yi, Wang**, Yufang Hou, and Lingfei Wu. 2021. HAConvGNN: Hierarchical Attention Based Convolutional Graph Neural Network for Code Documentation Generation in Jupyter Notebooks. In *Findings of the Association for Computational Linguistics: EMNLP 2021*. Association for Computational Linguistics, Punta Cana, Dominican Republic, 4473–4485.

[6] Lilian Lopez, Kiara Chau, Zeyu Xiong, Gustavo Kreia Umbelino, and **April Yi, Wang**. 2024. datAR: An Embodied Approach for Teaching Data Literacy Through Everyday Objects. (2024). . **In Submission**.

[7] Shuai Ma, Junling Wang, Yuanhao Zhang, Xiaojuan Ma, and **April Yi, Wang**. 2024. DBox: Assisting Algorithmic Programming Learning with Interactive and Adaptive Decomposition Box. (2024). . **In Submission**.

[8] **April Yi, Wang**, Yan Chen, John Joon Young Chung, Christopher Brooks, and Steve Oney. 2021. PuzzleMe: Leveraging Peer Assessment for In-Class Programming Exercises. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW2 (2021), 1–24.

[9] **April Yi, Wang**, Will Epperson, Robert A DeLine, and Steven M Drucker. 2022. Diff in the Loop: Supporting Data Comparison in Exploratory Data Analysis. In *CHI Conference on Human Factors in Computing Systems*. 1–10.

[10] **April Yi, Wang**, Andrew Head, Ashley Ge Zhang, Steve Oney, and Christopher Brooks. 2023. Colaroid: A literate programming approach for authoring explorable multi-stage tutorials. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–22.

[11] **April Yi, Wang**, Anant Mittal, Christopher Brooks, and Steve Oney. 2019. How data scientists use computational notebooks for real-time collaboration. *Proceedings of the ACM on Human-Computer Interaction* 3, CSCW (2019), 1–30. **Best Paper Award**.

[12] **April Yi, Wang**, Ryan Mitts, Philip J Guo, and Parmit K Chilana. 2018. Mismatch of expectations: How modern learning resources fail conversational programmers. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–13 . **Best Paper Honorable Mention**.

[13] **April Yi, Wang**, Dakuo Wang, Jaimie Drozdal, Michael Muller, Soya Park, Justin D Weisz, Xuye Liu, Lingfei Wu, and Casey Dugan. 2022. Documentation Matters: Human-Centered AI System to Assist Data Science Code Documentation in Computational Notebooks. *ACM Transactions on Computer-Human Interaction* 29, 2 (2022), 1–33.

[14] **April Yi, Wang**, Zihan Wu, Christopher Brooks, and Steve Oney. 2020. Callisto: Capturing the" Why" by Connecting Conversations with Computational Narratives. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–13. **Best Paper Honorable Mention**.

[15] Soya Park, **April Yi, Wang**, Ban Kawas, Q Vera Liao, David Piorkowski, and Marina Danilevsky. 2021. Facilitating knowledge sharing from domain experts to data scientists for building nlp models. In *26th International Conference on Intelligent User Interfaces*. 585–596.

[16] David Piorkowski, Soya Park, **April Yi, Wang**, Dakuo Wang, Michael Muller, and Felix Portnoy. 2021. How ai

developers overcome communication challenges in a multidisciplinary team: A case study. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW1 (2021), 1–25.

[17] Junling Wang, Gustavo Kreia Umbelino, Kiara Chau, Mrinmaya Sachan, and **April Yi, Wang**. 2024. ChatCommunity: A Retrieval-Augmented System Enriching AI-Generated Responses with Source Content Previews from Online Learning Communities. (2024). . **In Submission**.

[18] Yang Wu, Hai Jin, Yao Wan, and **April Yi, Wang**. 2024. STAGE: Stepwise Visual Explanations of AI-Generated Data Analysis Pipeline. (2024). . **In Submission**.

[19] Zeyu Xiong, Rasan Younis, Eliot Ullmo, Xiaoyu Zhang, and **April Yi, Wang**. 2024. Data Story: From Literate Programming to Literate Computing. (2024). . **In Preparation**.

[20] Yinuo Yang, Ashley Ge Zhang, Steve Oney, and **April Yi, Wang**. 2024. SPARK: Real-Time Monitoring of Multi-Faceted Programming Exercises. (2024). . **In Submission**.

[21] Chengbo Zheng, Dakuo Wang, **April Yi, Wang**, and Xiaojuan Ma. 2022. Telling Stories from Computational Notebooks: AI-Assisted Presentation Slides Creation for Presenting Data Science Work. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–20.