Composable Building Blocks for Controllable and Transparent Interactive AI Systems

Sebe Vanbrabant, Gustavo Rovelo Ruiz, and Davy Vanacken

Hasselt University - Flanders Make, Digital Future Lab, Diepenbeek, Belgium
sebe.vanbrabant@uhasselt.be gustavo.roveloruiz@uhasselt.be
davy.vanacken@uhasselt.be

Abstract. While the increased integration of AI technologies into interactive systems enables them to solve an equally increasing number of tasks, the black box problem of AI models continues to spread throughout the interactive system as a whole. Explainable AI (XAI) techniques can make AI models more accessible by employing post-hoc methods or transitioning to inherently interpretable models. While this makes individual AI models clearer, the overarching system architecture remains opaque. To this end, we propose an approach to represent interactive systems as sequences of structural building blocks, such as AI models and control mechanisms grounded in the literature. These can then be explained through accompanying visual building blocks, such as XAI techniques. The flow and APIs of the structural building blocks form an explicit overview of the system. This serves as a communication basis for both humans and automated agents like LLMs, aligning human and machine interpretability of AI models. We discuss a selection of building blocks and concretize our flow-based approach in an architecture and accompanying prototype interactive system.

Keywords: Intelligibility · Explainable AI · Large Language Models

1 Introduction

Artificial intelligence (AI) is becoming increasingly integrated into various interactive systems, with different challenges arising depending on the AI technologies used and the type of user interactions offered by these systems [6]. The increasing complexity of AI models, from interpretable decision trees to opaque Deep Neural Networks (DNNs) and Large Language Models (LLMs), has led to a decline in their transparency [3,18]. These models are often black boxes, producing results without explanations, justifications, or indications of uncertainties [8].The field of eXplainable AI (XAI) addresses these challenges by complementing AI predictions with explanations [9,30]. Machine learning (ML) workflows can be made more transparent in two ways: either by using white-box models that offer inherent interpretability, like decision trees, or by leveraging post-hoc explanations (e.g., LIME [24] and SHAP [17]) to try to explain the internal workings of black box models, such as neural networks.

We view explainability techniques like LIME, SHAP and the What-If tool [29] as visual *building blocks*. They address AI models' transparency by answering Why, Why-not, and What-if. However, no widespread building blocks exist that support users to control their AI models in the same way that LIME and SHAP address transparency. Current visual approaches can explicate AI behavior by interacting with the model (e.g., allow the user to change model inputs [29,10]) or through overviews of its internals (e.g., the structure of a neural network [5]).

While standardized approaches exist for interpreting model behavior, they are not necessarily applicable to the interactive system in which they are embedded. Kulesza et al. [15] found that the quality of a user's mental model directly correlates to their ability to control the underlying system as desired. This also applies to LLMs, which require careful prompting and the right (amount of) information to address user queries accurately. Approaches like Tool-Augmented Language Models (TALMs) [23] and Anthropic's recent Model Context Protocol (MCP) [2] enable LLMs to invoke code subroutines, facilitating their integration into interactive systems.

We envision an approach that simultaneously empowers users and automated agents to understand and control AI models. This involves extending XAI techniques and subroutine-based tools beyond *model-level* explanations. We instead look to support transparency and control in *system-level* AI workflows through structural and visual building blocks. For example, an AI model (structural building block) can be explained through LIME, SHAP, and WhatIf (visual building blocks), and further controlled through structured building blocks that, for instance, override unintended decisions [13] or give per-instance feedback [14]. To combine these blocks into one approach, we draw inspiration from neuro-symbolic AI (NSAI), which integrates neural and symbolic approaches to combine their strengths while circumventing their inherent weaknesses [27]. By incorporating techniques to explain one AI model into conceptual systems using structural building blocks, we aim to clarify interactive AI systems for humans and enable automated tools and agents, such as LLMs, to audit them using a shared knowledge base, aligning human and machine interpretation of AI models.

2 Related Work

Looking at the nine stages of the ML workflow, two major stages are evident: one data-oriented stage, involving data preparation, and one model-oriented stage, involving model (re)training and deployment [1]. For supervised ML, this results in a model that can predict new outputs from new inputs by leveraging its internal learning process. We can, thus, conceptually, view a trained model as a pipeline that transforms inputs into outputs through a model. These pipelines can be chained to make system behavior more advanced and fit for a task, which is the case for interactive systems embedding AI technologies. *Symbolic* AI, such as decision rules, excels at structured reasoning and provides high inherent explainability and interpretability [27]. However, symbolic approaches are less trainable and more error-prone in unfamiliar situations. In contrast, *con-*

nectionist techniques like neural networks excel at training by discovering and learning patterns from data, yet remain black boxes that require large datasets for effective training.

NSAI combines trainability and interpretability by using neural approaches to learn from experience and applying symbolic reasoning to draw conclusions from that knowledge [27]. Type 2 NSAI, as described by Kautz [12], considers connectionist models as neural module subroutines within a symbolic problem-solving system. TALMs are a recent example of type 2 NSAI systems. Systems like ViperGPT [25] and Chameleon [16] combine LLMs as neural subroutines within a symbolic tool usage framework. TALMs query tools rather than generating the answer directly, which is helpful for mathematical operations or to interface with external APIs.

The strengths of LLMs for XAI are evident in x-[plAIn] [20] and SHAPstories [19], which generate audience-specific summaries of XAI methods tailored to users' knowledge and interests, improving accessibility and decision-making. These approaches, however, do not offer capabilities other than those of the XAI methods. ECHO [26], a conversational approach to XAI, tackles this with a TALM using generated tools for explicating system-specific behavior complemented with predefined tools that address various explanation types and XAI methods. These approaches are all purely textual, however, and could be extended to intelligible interfaces. For instance, visualizations like those in TimberTrek [28] and AI-Spectra [7] can be enhanced by integrating conversational interfaces to help users understand and select the right models for their needs. An explainer offering recommendations and explanatory insights can make the process more accessible and interactive.

Aside from visualizing model analysis, other tools offer ways to build models visually using flow-based graph-like visualizations. To visualize models during development, DeepGraph [11] constructs the data flow graph representation of the architecture from the DNN source code and automatically synchronizes it with its graph representation. To enable users to build deep learning models visually, DeepFlow [5] uses a flow-based visual programming tool, realizing a no-code approach to building neural networks while viewing models as sequences of learnable functions. Existing approaches help visualize complex architectures and democratize AI development, but primarily focus on the AI development process rather than the model's role within the encapsulating AI system.

3 Building Blocks for Intelligibility and Control

While current approaches to explainability typically interrogate AI models by probing parts of the *input* → *model* → *output* pipeline through the Predict method, we propose expanding this conventional pipeline through structural and visual building blocks that also allow AI models to be visualized and controlled. By considering interactive systems embedding AI as type 2 NSAI systems consisting of specific components, their decision process can be represented through structural building blocks, which can be explained through visual building blocks. This gives both humans and AI agents a structured and shared knowledge base of (complex) system architectures through accessible building blocks.

3.1 Visual Building Blocks

Intelligiblity We initially considered the commonly used explanations of 'Why', 'Why-not', and 'What-if' [21]. Why and Why-not can be addressed by visual building blocks encompassing **LIME** and **SHAP**, which explain AI behavior using feature importance, which is commonly used to address Why and Why-not explanations. For What-if questions, we use a visual building block displaying all the predict method's input parameters and a corresponding output value, similar to the approach of He et al. [10].

3.2 Structural Building Blocks

Control One way to honor user feedback through a structural building block would be to allow users to re-label instances and retrain the model accordingly [14]. We draw further inspiration from the work on controllable AI by Kieseberg et al. [13]. Their five methods for managing control loss map to structural building blocks in our interactive NSAI pipeline. For non-autonomous AI systems, **DivineRuleGuard** ensures ethical compliance by overriding harmful or unethical decisions before they are acted upon as a postprocessing step for model output. Conversely, **NonGoalFilter** acts as a pre-processor, rejecting inputs that do not align with intended behaviors or intentions. **ShutdownTrigger** functions as an emergency stop to disable autonomous AI systems at any point. **BiasInjector** strategically influences decision-making by embedding predefined biases to guide the model toward preferred outcomes. Lastly, **LogicBomb** operates as a self-monitoring fail-safe, resetting or shutting down the AI if it ever attempts to produce an outcome that breaches ethical or operational boundaries.

Execution Flow Aside from controlling the individual NSAI components, we also envision components for visualizing and controlling conditional execution flows between components. This is useful in scenarios where multiple AI models are used together, such as in the context of ensemble learning [22] or model multiplicity [28,7], techniques commonly used in high-stakes interactive systems. Specifically, we envision a **Splitter** and **Aggregator**, where the Splitter indicates the dataset being distributed to different AI models. The Aggregator then displays how the final output is produced/aggregated from these models. For model multiplicity, Chernoff bots could be the visual building block for each individual model; its dashboard can be linked to the aggregator.

4 Auditable 5-Layer Architecture for Transparent and Controllable Interactive AI Systems

We apply our proposed 5-layer architecture to an example heart disease prediction ensemble in Fig. 1. The vertical pipeline was loosely based on the XAI system depicted by Mohseni et al. [21], modified to support structural and visual building blocks. We define the following layers:

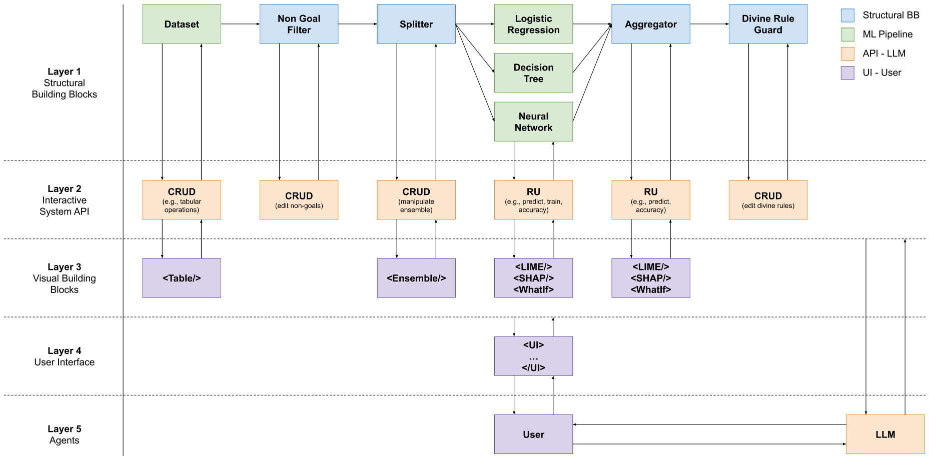


Fig. 1. An example heart disease prediction ensemble to illustrate our proposed architecture. Layer 1 shows the structural building blocks, consisting of our building blocks (in blue) integrated into the ML pipeline (in green). Layer 2 converts the structural building blocks into a callable API, usable by the visual building blocks in layer 3, assembled into the user interface of layer 4. The API is also accessible by the LLM of layer 5 so that both agents use common knowledge.

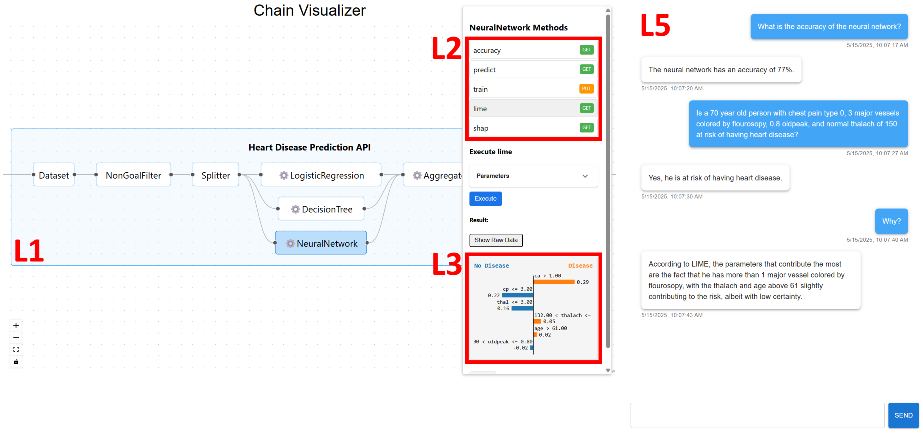
Layer 1: Structural Building Blocks Structural building blocks convey the (conceptual) architecture of the interactive system at a glance. Each block is mapped to a part of the system's source code, completely specified by the developer through function decorators. Since this representation is purely conceptual, the developer can choose what (parts of) the system to expose and how to communicate the pipeline.

Layer 2: Interactive System API Developers can write their code as usual, and link it to conceptual structural building blocks through developer-defined methods. Each block's REST API is automatically generated from the structural building blocks' definition and its decorator of layer 1.

Layer 3: Visual Building Blocks Visual building blocks interact with the structural building blocks of layer 1 through the API of layer 2. For instance, a structural building block of an AI model can have its behavior explained through LIME, for which data is acquired over the REST API.

Layer 4: User Interface Structural and visual building blocks are combined into an interface where they can be explored, interrogated, and controlled. Currently, visual building blocks are assembled into one coherent UI; future research directions include LLM-powered layouts [4].

- **Layer 5: Agents** The final layer of the architecture comprises agents that interact with the building blocks. These can be users interacting with the UI and its visual building blocks, or an automated LLM agent interacting with the shared REST API of layer 2. This API can then be integrated as tools for a TALM such as ECHO [26]. Both agents have access to the same information, and users can interact with the LLM to ask about system behavior.



What is the accuracy of the neural network?
5/15/2025, 10:07:17 AM

The neural network has an accuracy of 77%.
5/15/2025, 10:07:20 AM

Is a 70 year old person with chest pain type 0, 3 major vessels colored by flourosopy, 0.8 oldpeak, and normal thalach of 150 at risk of having heart disease?
5/15/2025, 10:07:27 AM

Yes, he is at risk of having heart disease.
5/15/2025, 10:07:30 AM

Why?
5/15/2025, 10:07:40 AM

According to LIME, the parameters that contribute the most are the fact that he has more than 1 major vessel colored by flourosopy, with the thalach and age above 61 slightly contributing to the risk, albeit with low certainty.
5/15/2025, 10:07:43 AM

Fig. 2. Prototype of our approach using the heart disease prediction ensemble. The UI (layer 4) shows each structural building block (layer 1) influencing predictions, exposing system behavior to both users and automated agents (layer 5) through visual building blocks (layer 3) and an API (layer 2), respectively.

5 Conclusion

Rising AI complexity has led to an increase in challenges regarding explaining and controlling AI behavior. These challenges propagate from the individual model to the system that embeds it, making the entire interactive system opaque to users. We proposed a preliminary architecture for making interactive systems more accessible by explicitly conveying their conceptual model through an API. This enables both users and LLMs to access information related to system behavior, aligning human and machine interpretability of AI models. Future research directions include applying our architecture to more elaborate NSAI applications and use cases involving model multiplicity, such as integrating the AI-Spectra dashboard and its Chernoff bots as visual building blocks [7]. Furthermore, it would be interesting to explore and integrate user-specific, personalized, and dynamic UIs into the textual conversations that adapt to individual user needs.

Acknowledgments. This work was funded by the Special Research Fund (BOF) of Hasselt University, BOF23OWB31.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Amershi, S., Begel, A., Bird, C., DeLine, R., Gall, H., Kamar, E., Nagappan, N., Nushi, B., Zimmermann, T.: Software Engineering for Machine Learning: A Case Study. In: 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP). pp. 291–300 (2019). https://doi.org/10.1109/ICSE-SEIP.2019.00042
2. Anthropic: Introducing the Model Context Protocol (2024), https://www.anthropic.com/news/model-context-protocol
3. Barredo Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., Herrera, F.: Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Information Fusion **58**, 82–115 (2020). https://doi.org/10.1016/j.inffus.2019.12.012
4. Brie, P., Burny, N., Sluÿters, A., Vanderdonckt, J.: Evaluating a Large Language Model on Searching for GUI Layouts. Proc. ACM Hum.-Comput. Interact. **7**(EICS) (Jun 2023). https://doi.org/10.1145/3593230
5. Calò, T., De Russis, L.: DeepFlow: A Flow-Based Visual Programming Tool for Deep Learning Development. In: Proceedings of the 30th International Conference on Intelligent User Interfaces. p. 504–518. IUI '25, Association for Computing Machinery, New York, NY, USA (2025). https://doi.org/10.1145/3708359.3712109
6. Dix, A., Mayer, S., Palanque, P., Panizzi, E., Spano, L.D.: Engineering Interactive Systems Embedding AI Technologies. In: Companion Proceedings of the 2023 ACM SIGCHI Symposium on Engineering Interactive Computing Systems. p. 90–92. EICS '23 Companion, Association for Computing Machinery, New York, NY, USA (2023). https://doi.org/10.1145/3596454.3597195
7. Eerlings, G., Vanbrabant, S., Liesenborgs, J., Rovelo Ruiz, G., Vanacken, D., Luyten, K.: AI-Spectra: A Visual Dashboard for Model Multiplicity to Enhance Informed and Transparent Decision-Making. In: Zaina, L., Campos, J.C., Spano, D., Luyten, K., Palanque, P., van der Veer, G., Ebert, A., Humayoun, S.R., Memmesheimer, V. (eds.) Engineering Interactive Computer Systems. EICS 2024 International Workshops. pp. 55–73. Springer Nature Switzerland, Cham (2025). https://doi.org/10.1007/978-3-031-91760-8_5
8. von Eschenbach, W.J.: Transparency and the Black Box Problem: Why We Do Not Trust AI. Philosophy & Technology **34**(4), 1607–1622 (Dec 2021). https://doi.org/10.1007/s13347-021-00477-0
9. Gunning, D., Stefik, M., Choi, J., Miller, T., Stumpf, S., Yang, G.Z.: XAI—Explainable artificial intelligence. Science Robotics **4**(37), eaay7120 (2019). https://doi.org/10.1126/scirobotics.aay7120
10. He, G., Aishwarya, N., Gadiraju, U.: Is Conversational XAI All You Need? Human-AI Decision Making With a Conversational XAI Assistant. In: Proceedings of the 30th International Conference on Intelligent User Interfaces. p. 907–924. IUI '25, Association for Computing Machinery, New York, NY, USA (2025). https://doi.org/10.1145/3708359.3712133

11. Hu, Q., Ma, L., Zhao, J.: DeepGraph: A PyCharm Tool for Visualizing and Understanding Deep Learning Models. In: 2018 25th Asia-Pacific Software Engineering Conference (APSEC). pp. 628–632 (2018). https://doi.org/10.1109/APSEC.2018.00079
12. Kautz, H.: The Third AI Summer: AAAI Robert S. Engelmore Memorial Lecture. AI Magazine **43**(1), 105–125 (Mar 2022). https://doi.org/10.1002/aaai.12036
13. Kieseberg, P., Weippl, E., Tjoa, A.M., Cabitza, F., Campagner, A., Holzinger, A.: Controllable AI - An Alternative to Trustworthiness in Complex AI Systems? In: Holzinger, A., Kieseberg, P., Cabitza, F., Campagner, A., Tjoa, A.M., Weippl, E. (eds.) Machine Learning and Knowledge Extraction. pp. 1–12. Springer Nature Switzerland, Cham (2023). https://doi.org/10.1007/978-3-031-40837-3_1
14. Kulesza, T., Burnett, M., Wong, W.K., Stumpf, S.: Principles of Explanatory Debugging to Personalize Interactive Machine Learning. In: Proceedings of the 20th International Conference on Intelligent User Interfaces. p. 126–137. IUI '15, Association for Computing Machinery, New York, NY, USA (2015). https://doi.org/10.1145/2678025.2701399
15. Kulesza, T., Stumpf, S., Burnett, M., Kwan, I.: Tell me more? the effects of mental model soundness on personalizing an intelligent agent. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. p. 1–10. CHI '12, Association for Computing Machinery, New York, NY, USA (2012). https://doi.org/10.1145/2207676.2207678
16. Lu, P., Peng, B., Cheng, H., Galley, M., Chang, K.W., Wu, Y.N., Zhu, S.C., Gao, J.: Chameleon: Plug-and-Play Compositional Reasoning with Large Language Models. In: Proceedings of the 37th International Conference on Neural Information Processing Systems. NIPS '23, Curran Associates Inc., Red Hook, NY, USA (2023)
17. Lundberg, S.M., Lee, S.I.: A Unified Approach to Interpreting Model Predictions. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. p. 4768–4777. NIPS'17, Curran Associates Inc., Red Hook, NY, USA (2017)
18. Luo, H., Specia, L.: From Understanding to Utilization: A Survey on Explainability for Large Language Models (2024), https://arxiv.org/abs/2401.12874
19. Martens, D., Hinns, J., Dams, C., Vergouwen, M., Evgeniou, T.: Tell me a story! Narrative-driven XAI with Large Language Models. Decision Support Systems **191**, 114402 (2025). https://doi.org/10.1016/j.dss.2025.114402
20. Mavrepis, P., Makridis, G., Fatouros, G., Koukos, V., Separdani, M.M., Kyriazis, D.: XAI for All: Can Large Language Models Simplify Explainable AI? (2024), https://arxiv.org/abs/2401.13110
21. Mohseni, S., Zarei, N., Ragan, E.D.: A Multidisciplinary Survey and Framework for Design and Evaluation of Explainable AI Systems. ACM Trans. Interact. Intell. Syst. **11**(3–4) (Sep 2021). https://doi.org/10.1145/3387166
22. Opitz, D., Maclin, R.: Popular ensemble methods: an empirical study. J. Artif. Int. Res. **11**(1), 169–198 (Jul 1999)
23. Parisi, A., Zhao, Y., Fiedel, N.: TALM: Tool Augmented Language Models (2022), https://arxiv.org/abs/2205.12255
24. Ribeiro, M.T., Singh, S., Guestrin, C.: "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. p. 1135–1144. KDD '16, Association for Computing Machinery, New York, NY, USA (2016). https://doi.org/10.1145/2939672.2939778

25. Surís, D., Menon, S., Vondrick, C.: ViperGPT: Visual Inference via Python Execution for Reasoning. In: 2023 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 11854–11864 (2023). https://doi.org/10.1109/ICCV51070.2023.01092
26. Vanbrabant, S., Eerlings, G., Rovelo Ruiz, G., Vanacken, D.: ECHO: Enhancing Conversational Explainable AI through Tool-Augmented Language Models. Proc. ACM Hum.-Comput. Interact. **9**(EICS) (Jun 2025). https://doi.org/10.1145/3734191
27. Wang, W., Yang, Y., Wu, F.: Towards Data-And Knowledge-Driven AI: A Survey on Neuro-Symbolic Computing. IEEE Transactions on Pattern Analysis and Machine Intelligence **47**(2), 878–899 (2025). https://doi.org/10.1109/TPAMI.2024.3483273
28. Wang, Z.J., Zhong, C., Xin, R., Takagi, T., Chen, Z., Chau, D.H., Rudin, C., Seltzer, M.: TimberTrek: Exploring and Curating Sparse Decision Trees with Interactive Visualization. In: 2022 IEEE Visualization and Visual Analytics (VIS). pp. 60–64 (2022). https://doi.org/10.1109/VIS54862.2022.00021
29. Wexler, J., Pushkarna, M., Bolukbasi, T., Wattenberg, M., Viégas, F., Wilson, J.: The What-If Tool: Interactive Probing of Machine Learning Models. IEEE Transactions on Visualization and Computer Graphics **26**(1), 56–65 (2020). https://doi.org/10.1109/TVCG.2019.2934619
30. Xu, X., Yu, A., Jonker, T.R., Todi, K., Lu, F., Qian, X., Evangelista Belo, J.a.M., Wang, T., Li, M., Mun, A., Wu, T.Y., Shen, J., Zhang, T., Kokhlikyan, N., Wang, F., Sorenson, P., Kim, S., Benko, H.: XAIR: A Framework of Explainable AI in Augmented Reality. In: Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems. CHI '23, Association for Computing Machinery, New York, NY, USA (2023). https://doi.org/10.1145/3544548.3581500