IvoryOS: An Interoperable Platform and Community for Self-Driving Labs

Ivory Wenyu Zhang [1 2 3] **Maria Politi** [1 2] **Veronica Lai** [3] **Rama El-khawaldeh** [1] **Jason E. Hein** [1 2 3 4]

[1]*The University of British Columbia, Department of Chemistry, Vancouver, Canada* [2]*Acceleration Consortium, University of Toronto* [3]*Telescope Innovations Corp., Vancouver, Canada* [4]*University of Bergen, Norway*. Correspondence to: Ivory Wenyu Zhang ivoryzhang@chem.ubc.ca, Jason E. Hein jhein@chem.ubc.ca.

1. Introduction

Self-driving laboratories (SDLs) promise to accelerate scientific discovery by tightly coupling automation, data acquisition, and AI–driven decision making. Despite significant advances in optimization algorithms and robotic platforms, the practical development and adoption of SDLs remain challenging. A major barrier is the lack of hardware–software interoperability, which forces most SDL projects to rely on bespoke engineering solutions that tightly couple hardware control, experimental logic, and optimization workflows. This fragmentation introduces repetitive works, limits reuse and reduces reproducibility [1, 2]. Furthermore, SDL development requires close collaboration between domain scientists and engineers; however, such "bilingual" expertise is rare, as current educational and training pipelines remain largely siloed by discipline. As a result, experimentalists are constrained by limited programming fluency, while engineers lack context when building SDL infrastructures. Together, these challenges hinder platform reproducibility, scalability, and the broader adoption of SDL technologies.

2. Interoperable SDL Interface

Interoperability and visual interfaces are critical to lowering the barrier to building and operating SDLs. Interoperable software decouples hardware control, experimental logic, and optimization strategies, enabling reuse across instruments, workflows, and laboratories while improving reproducibility and long-term maintainability. At the same time, visual interfaces provide an accessible layer that allows domain scientists to design, inspect, and modify experimental workflows without requiring deep programming expertise.

2.1 IvoryOS architecture

IvoryOS introduces an interoperable orchestrator that enables closed-loop experimentation on existing automated protocols without imposing a new abstraction layer [3]. Rather than requiring protocols to be rewritten in a domain-specific language, IvoryOS programmatically introspects executable Python code to extract protocol metadata, which is then used to dynamically generate operation menus for the controller and workflow builder (Figure 1). Scientists can directly use the protocols or low-code design a sequential experiment workflows that supports dynamic parameters. With the workflow engine, the experiment can be executed in closed-loop with options of several open-source optimizer libraries. Simultaneously, the workflow engine supports granular process control, and systematic logging and error handling across iterations.

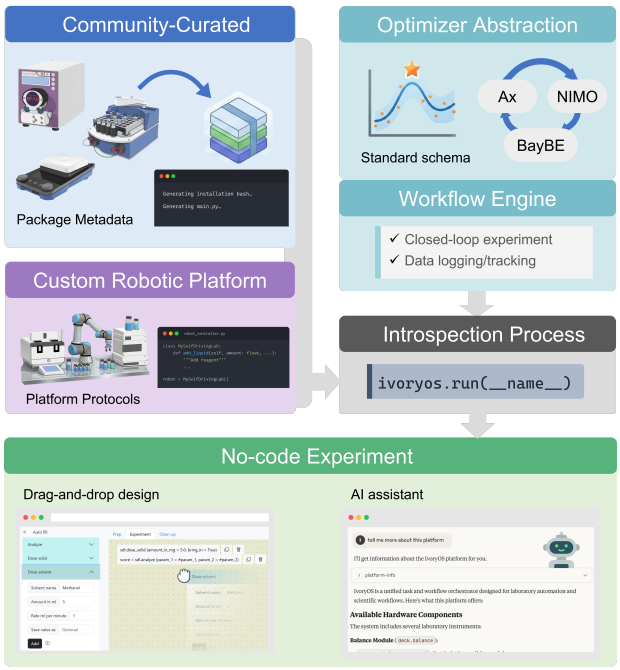

Fig. 1: IvoryOS Architecture Overview

2.2 Optimization Standardization Schema

The optimization layer employs a structured schema to interface with different Bayesian optimization (BO) engines, including Ax [4], BayBE [5], and NIMO [6]. By standardizing experimental data input and output, this layer enables users to interchange optimization providers and strategies without modifying the underlying experimental code.

2.3 Natural Language Layer

To further reduce the on-boarding time, this orchestrator is also extensible with a Model Context Protocol (MCP) server. The server uses Application Programming Interface (API) request to communicate to the main orchestration layer, enabling AI-native decision-making that can plan, adapt, and optimize experiments in real time [7]. Because all metadata is derived directly from executable code, it remains synchronized with the underlying experiment and requires no manual specification or schema maintenance.

2.4 Demonstration in SDLs

This approach is validated through several automation scenarios, including high-throughput parameter sweep, closed-loop optimization and multi-instrument workflows. In these demonstrations, the same experimental interface is successfully deployed on more than 20 robotic platforms in different institutions without modification of the underlying control code. This highlights how introspection-based metadata decouples experimental design from algorithmic choice. Additionally, we show that experimental workflows expressed through IvoryOS interfaces can be shared and re-executed across deployments with minimal configuration effort. Such reproducibility within the domain is critical for knowledge transfer and for accelerating the development of scientific automation.

2.5 Ecosystem and Community Effects

Beyond individual laboratories, IvoryOS supports a community-driven ecosystem for scientific automation. Developers can share metadata for instrument drivers, reusable data analysis modules, and experiment templates without requiring standardization. The platform automatically generates an installation package that launches IvoryOS, reducing the complexity of setting up Python environments and dependencies. This ecosystem provides a curated catalog of automation-ready instruments and software, enabling anyone to construct a customized platform and execute SDL workflows with minimal coding effort. While accelerating and democratizing the research and development of SDLs, this also accelerates the training of programming along with the susses of platform development.

Acknowledgments

Financial support for this work was provided by the Canada Foundation for Innovation (CFI-35833, CFI-44843), Natural Sciences and Engineering Research Council of Canada (RGPIN-2021–03168, Discovery Accelerator) and the University of British Columbia to J.E.H. This research was undertaken thanks in part to funding provided to the University of Toronto's Acceleration Consortium from the Canada First Research Excellence Fund (CFREF-2022–00042).

References

[1] Richard B. Canty, Jeffrey A. Bennett, Keith A. Brown, Tonio Buonassisi, Sergei V. Kalinin, John R. Kitchin, Benji Maruyama, Robert G. Moore, Joshua Schrier, Martin Seifrid, Shijing Sun, Tejs Vegge, and Milad Abolhasani. Science acceleration and accessibility with self-driving labs. *Nature Communications*, 16:3856, 2025.

[2] Martin Seifrid, Robert Pollice, Andrés Aguilar-Granda, Zamyla Morgan Chan, Kazuhiro Hotta, Cher Tian Ser, Jenya Vestfrid, Tony C. Wu, and Alán Aspuru-Guzik. Autonomous chemical experiments: Challenges and perspectives on establishing a self-driving lab. *Accounts of Chemical Research*, 55(17):2454–2466, 2022.

[3] Wenyu Zhang, Lucy Hao, Veronica Lai, Ryan Corkery, Jacob Jessiman, Jiayu Zhang, Junliang Liu, Yusuke Sato, Maria Politi, Matthew E. Reish, Rebekah Greenwood, Noah Depner, Jiyoon Min, Rama El-khawaldeh, Paloma Prieto, Ekaterina Trushina, and Jason E. Hein. IvoryOS: an interoperable web interface for orchestrating Python-based self-driving laboratories. *Nature Communications*, 16(1):5182, 2025.

[4] Miles Olson, Elizabeth Santorella, Louis C. Tiao, Sait Cakmak, David Eriksson, Mia Garrard, Sam Daulton, Maximilian Balandat, Eytan Bakshy, Elena Kashtelyan, Zhiyuan Jerry Lin, Sebastian Ament, Bernard Beckerman, Eric Onofrey, Paschal Igusti, Cristian Lara, Benjamin Letham, Cesar Cardoso, Shiyun Sunny Shen, Andy Chenyuan Lin, and Matthew Grange. Ax: A platform for adaptive experimentation. In *AutoML 2025 ABCD Track*, 2025.

[5] Martin Fitzner, Adrian Šošić, Alexander V. Hopp, Marcel Müller, Rim Rihana, Karin Hrovatin, Fabian Liebig, Mathias Winkel, Wolfgang Halter, and Jan Gerit Brandenburg. Baybe: a bayesian back end for experimental planning in the low-to-no-data regime. *Digital Discovery*, 4:1991–2000, 2025.

[6] Ryo Tamura, Koji Tsuda, and Shoichi Matsuda. Nims-os: an automation software to implement a closed loop between artificial intelligence and robotic experiments in materials science. *Science and Technology of Advanced Materials: Methods*, 3(1):2232297, 2023.

[7] Wenyu Zhang, Maria Politi, Rama El-Khawaldeh, and Jason Hein. Interoperable natural language interfaces for self-driving labs via model context protocol. In *Proceedings of the NeurIPS 2025 Workshop on AI for Materials and Autonomous Labs (AI4Mat)*, 2025. OpenReview, CC BY 4.0.