PhD projects (Systems)

- Software sustainability analysis and improvement
- Automated Search-Based Optimisation of Quantum Implementations
- Debugging compiler-optimised code: stateful approaches
- Understanding software growth and induced demand effects
- Testing Methodology for Theorem Provers Ecosystems
- Quantum Machine Learning: Characterising the Quantum Advantage

Software sustainability analysis and improvement

Supervisor: Kevin Lano

Project Description

The project would consider techniques for analysing the software environmental sustainability (in the sense of energy use and energy efficiency) of machine learning systems, or of other kinds of software system with high energy use (such as quantitative financial software) using either rule-based analysis and refactoring, or by the use of deep learning techniques such as LLMs to identify energy use flaws and potential refactorings to improve energy efficiency.

There is the potential for industrial collaboration in this area.

References

K. Lano et al, "Model-based Software Sustainability Analysis and Improvement", SoSym, 2026.
M. Mehditabar and S. Rajput and T. Sharma, "A Validated Taxonomy on Software Energy Smells", arXiv:2604.04809v1.
H. Siala et al, "Using LLMs for software energy-use analysis and improvement", SusMod '26.

Automated Search-Based Optimisation of Quantum Implementations

Supervisor: Dr Karine Even-Mendoza

Project Description

Quantum implementations are developed within complex software ecosystems, from high-level implementations in frameworks such as Qiskit or Cirq, through parameterised circuits, to transpiled circuits targeting specific backends. Their performance depends on many choices across this process, including hyperparameters, circuit structure, initialisation, compilation and transpilation decisions, the properties of the input quantum system, and the way noise is represented or modelled.

This PhD project will investigate how classical and quantum machine learning, genetic algorithms, and search-based software engineering techniques can be used to analyse and optimise quantum implementations, focusing at first on VQE, while aiming to explore later on, other famalies of implemenations. The main research question is: how can we systematically optimise quantum programs across representations using learned models and search?

It builds on AccelerQ, a framework that learns surrogate performance models from small, classically simulable quantum systems and uses these models to guide the optimisation of larger quantum eigensolver implementations. Instead of repeatedly executing expensive quantum runs during search, ML models approximate candidate quality, enabling techniques such as genetic algorithms to explore large optimisation spaces more efficiently.

The project will extend this idea beyond hyperparameter optimisation by considering information available at different stages, including the input, parameterised circuits, transpiled circuits, backend characteristics, and noise models. It will investigate how these representations can be used together to guide optimisation decisions, and how search and ML/QML can identify improvements that remain effective when validated through execution.

The overall aim is to develop a principled, empirically validated framework for automated optimisation of quantum software across multiple representation levels, contributing techniques for quantum program analysis, optimisation, and search-based software engineering while improving the accuracy, efficiency, and practical scalability of quantum implementations.

References

[1] Avner Bensoussan, Elena Chachkarova, Karine Even-Mendoza, Sophie Fortz, and Connor Lenihan. 2025. AccelerQ: Accelerating Quantum Eigensolvers with Machine Learning on Quantum Simulators. Proc. ACM Program. Lang. 9, OOPSLA2, Article 354 (October 2025), 31 pages. https://doi.org/10.1145/3763132
[2] Juan Manuel Murillo, Jose Garcia-Alonso, Enrique Moguel, Johanna Barzen, Frank Leymann, Shaukat Ali, Tao Yue, Paolo Arcaini, Ricardo Perez-Castillo, Ignacio Garcia-Rodriguez de Guzman, Mario Piattini, Antonio Ruiz-Cortes, Antonio Brogi, Jianjun Zhao, Andriy Miranskyy, and Manuel Wimmer. 2025. Quantum Software Engineering: Roadmap and Challenges Ahead. ACM Trans. Softw. Eng. Methodol. 34, 5, Article 154 (June 2025), 48 pages. https://doi.org/10.1145/3712002
[3] Felix Gemeinhardt, Stefan Klikovits, and Manuel Wimmer. 2025. GeQuPI: Quantum Program Improvement with Multi-Objective Genetic Programming. J. Syst. Softw. 219, C (Jan 2025). https://doi.org/10.1016/j.jss.2024.112223

Debugging compiler-optimised code: stateful approaches

Supervisor: Stephen Kell

Project Description

Optimised code, as generated by ahead-of-time compilers such as GCC and LLVM, has long been poorly supported by existing debugging tools. A recurring bottleneck is that there is no comprehensive way to retain *state* in the debugger -- for example, when a local variable is eliminated by the compiler. Several obscure compiler-side mechanisms attempt to rectify this "by the back door" but are limited and poorly specified. Meanwhile other debugging tools, such as rr or UndoDB, which maintain large amounts of additional state, do so at a high cost.

This thesis project will investigate how to evolve our current execution infrastructure in commodity debuggers, operating systems and language implementations to provide reliable source-level debugging using stateful approaches. For example, it may devise an overhaul to DWARF debugging information that provides a general and elegant approach to debugger-side state. It may develop efficient hybrids that bridge the gap between powerful but state-heavy recording techniques and the lightweight convenience of attaching the debugger only when needed. It is likely to explore, to some extent, the usability issues inherent in more stateful tools.

It builds on our EPSRC- and DARPA-funded prior work, so will not be starting from scratch.

Understanding software growth and induced demand effects

Supervisor: Stephen Kell

Project Description

Computing resources are no longer scarce: computers are plentiful and powerful. Most software's functionality rarely challenges the limits of the hardware. Despite this, users continue to experience software that is slow. Moreover, users continue to replace hardware with newer hardware in order to 'keep up' with the declining unit performance of everyday software (long observed, e.g. the famous Wirth's law). One theory to explain this is 'induced demand', or Jevons' paradox, where greater capacity changes habits of programmers and users in ways that effectively 'soak up' the extra capacity and, often, worsen the apparent infrastructure shortfall.

This thesis project will study the phenomenon of induced demand in commodity software. It will most likely consist of developing novel profiling tools to study the evolution of the performance of commodity software, and of case studies that pinpoint technical decisions or changes which explain the loss of performance. One possible angle is to study open-source desktop software over the period from the mid-1990s to the present; one tool-building tactic would be to exploit how a single Linux kernel can host user software environments spanning a large interval of time.

Testing Methodology for Theorem Provers Ecosystems

Supervisor: Dr Mohammad Ahmad Abdulaziz Ali Mansour, Dr Karine Even Mendoza

Project Description

Proof assistants play a central role in applications, ranging from practical software verification, such as verifying a Linux kernel [1], a C compiler [2], or cryptographic software by Apple [3] and AWS [4], to more purely scientific applications in mathematics, such as verifying the correctness of the proof of Fermat's Last Theorem [5]. In software verification, a proof assistant checks a formal argument that a program satisfies its specification and behaves as intended.

Nonetheless, proof assistants are themselves complex software systems. They integrate numerous components, including an IDE, external proof engines (e.g. SMT solvers), internal proof engines (e.g. term-rewriting systems, unifiers or simplifiers), and a kernel that checks proofs. Bugs in these components, or in the interactions between them, can lead to incorrect or misleading results, crashes, hangs, and other failures that affect the reliability and usability of the overall verification system. Although the trusted core of theorem provers such as Isabelle is designed to be highly reliable, formally verifying an entire proof-assistant ecosystem is difficult because many of its components and interactions do not have complete, human-readable formal specifications. AI-driven verification may help automate parts of the verification process, yet it cannot establish correctness without a specification of what the system is intended to do. Automated testing therefore offers a complementary way to assess reliability without requiring a complete formal model of the whole toolchain.

This PhD project will investigate new automated testing methodologies for proof assistants through a focused study of Isabelle. The project will explore the design and effectiveness of: (1) automated test harnesses for individual proof-assistant components; (2) generators and mutators for highly structured test inputs; (3) component-specific test oracles for identifying unexpected behaviour; and (4) cross-version fuzzing and differential testing for discovering regressions, inconsistencies, and behavioural changes.

Through this work, the PhD candidate will be exposed to ideas and techniques from logic, formal methods, programming languages, compilers, and fuzzing, and will develop the ability to connect these areas when addressing research problems in proof-assistant reliability.

References

[1] Klein G, Elphinstone K, Heiser G, Andronick J, Cock D, Derrin P, Elkaduwe D, Engelhardt K, Kolanski R, Norrish M, Sewell T. seL4: Formal verification of an OS kernel. In Proceedings SOSP 2009 (pp. 207-220).
[2] Leroy, X., 2009. Formal verification of a realistic compiler. Commun. ACM 52, 107—115. https://doi.org/10.1145/1538788.1538814
[3] https://support.apple.com/en-gb/guide/security/sec59b0b31ff/web
[4] https://github.com/awslabs/AutoCorrode
[5] https://leanprover-community.github.io/blog/posts/FLT-announcement/
[6] Nipkow, T., Paulson, L.C., Wenzel, M., 2002. Isabelle/HOL - A Proof Assistant for Higher-Order Logic, Lecture Notes in Computer Science. Springer. https://doi.org/10.1007/3-540-45949-9

Quantum Machine Learning: Characterising the Quantum Advantage

Supervisor: Mohammad Reza Mousavi (Informatics), Rocio Martinez Nunez (Infectious Diseases)

Project Description

Quantum Machine Learning (QML) has emerged as a novel extension of "traditional" machine learning where parts of the machine learning pipeline are replaced by quantum circuits. Quantum computing processes information using qubits (instead of classical bits, 1 or 0), which can be in any combination of 0 and 1 states simultaneously[RM1.1]. Due to the rich input and output space of quantum computing, this promises novel ways of encoding and processing the data, potentially leading to faster convergence and/or better performance, and different answers to classical methods[RM2.1]. Many of the traditional machine learning techniques have now been extended to use the rich space of quantum information with early results suggesting quantum advantages, e.g., in terms of reduced training time, for the interpretation of complex datasets such as those contained in omics technologies, currently employed to interrogate disease-defining changes at the molecular level.

The aim of this project is to characterise the shape of data that gives various quantum machine learning techniques a practical advantage over classical data. We will start off with simple classification techniques (such as quantum support vector machines) and will move on to more sophisticated architectures (such as quantum neural networks).

We want to characterise the relevance of these approaches in solving biologically meaningful questions. While great advances have been done to discover the genetic make-up of cells, decoding and interpreting what it means in the subsequent steps in gene expression remains a hard task. Machine learning is currently used to classify gene expression patterns, or determine the most important parameters that explain large variation patterns in gene expression. However, the biological relevance of these models remains elusive, and they are likely missing connections that cannot be revealed using classical methods. We aim to understand fundamental questions underpinning gene expression patterns of importance during infection and inflammation. When cells encounter external stimuli, such a virus, they reprogramme themselves to be able to survive infection and stop the virus from spreading and replicating. This entails changes in the levels of thousands of molecules, in different places inside the cells. Current methodologies enable measuring thousands of mediators and their localization, but good and novel tools to easily establish predictive patterns or reduce the dimensionality of datasets to the minimal features that underlie these changes is a hard computational task.

We will investigate the extent of the quantum advantage promise on different machine learning tasks, with different hyper-parameters and measure its effect in a systematic manner. Our project will be conducted on data provided by colleagues from infectious diseases to demonstrate practical advantages in concrete applications in drug discovery and causal discovery in respiratory and inflammatory diseases.
Project aims and objectives

As an application area, we aim is to develop a qML model that enables integration of omics datasets to predict antiviral and inflammatory responses. The student will do this following these objectives:
- Develop a classifier to group molecules into clusters underpinning certain biological outputs (e.g. antiviral status, anti-inflammatory status);
- Develop a qML model that reduces the number of variables to those key underpinning the gene expression output;
- Characterise the boundaries of quantum advantage of Quantum ML techniques over their classical counterparts, particularly focussing on the qML models and the biological datasets developed above.

References

Elias F. Combarro and Samuel Gonzalez-Castillo, A Practical Guide to Quantum Machine Learning and Quantum Optimization: Hands-on Approach to Modern Quantum Algorithms, Packt, 2023
Zoran Krunic, Frederik F. Flother, George Seegan, Nathan D. Eearnest-Noble , and Omar Shehab. Quantum Kernels for Real-World Predictions Based on Electronic Health
Records. IEEE Trans. on Quantum Engineering. 2022.
Md Nadim, Mohammad Hassan, Ashis Kumar Mandal, Chanchal K. Roy. Quantum vs. Classical Machine Learning Algorithms for Software Defect Prediction: Challenges and Opportunities Quantum. Proc. of Q-SE 2025, ACM/IEEE, 2025.
Xinyi Wang, Shaukat Ali, Paulo Arcaini, Narasimha Raghavan Veeraragavan, Jan F Nygard.
Quantum Neural Network Classifier for Cancer Registry System Testing: A Feasibility Study.
arXiv preprint arXiv:2411.04740, 2024.